

Moduł szkoleniowy JavaScript

poziom podstawowy 15 godzinny

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Spis treści

1	Metryka dokumentu.....	5
2	Wstęp	6
3	Cel	6
4	Opis sposobu realizacji celów	6
5	Treści kształcenia.....	7
6	Opis założonych osiągnięć ucznia	7
7	Korelacja z podstawą programową	7
8	Sposoby osiągania celów	8
8.1	Projekt zaliczeniowy	8
9	Propozycje kryteriów oceny i metod sprawdzania osiągnięć ucznia	8
10	Test końcowy sprawdzający wiedzę	9
11	Lekcje.....	11
11.1	Lekcja 1 - Wprowadzenie.....	11
11.1.1	Cel lekcji.....	11
11.1.2	Sposoby osiągania celów kształcenia	11
11.1.3	Treść - slajdy z opisem	12
11.1.4	Ćwiczenia	29
11.1.5	Opis założonych osiągnięć ucznia	29
11.2	Lekcja 2 – Zmienne, typy danych i operatory	30
11.2.1	Cel lekcji.....	30
11.2.2	Sposoby osiągania celów kształcenia	30
11.2.3	Treść - slajdy z opisem	30
11.2.4	Ćwiczenia	47
11.2.5	Opis założonych osiągnięć ucznia	47
11.3	Lekcja 3 – Instrukcje warunkowe i pętle	47
11.3.1	Cel lekcji.....	47
11.3.2	Sposoby osiągania celów kształcenia	47
11.3.3	Treść – slajdy z opisem	47
11.3.4	Ćwiczenia	72
11.3.5	Opis założonych osiągnięć ucznia	72

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



11.4	Lekcja 4 – Tablice	72
11.4.1	Cel lekcji	72
11.4.2	Sposoby osiągnięcia celów kształcenia	72
11.4.3	Treść – slajdy z opisem	73
11.4.4	Ćwiczenia	89
11.4.5	Opis założonych osiągnięć ucznia	90
11.5	Lekcja 5 – Obiekty i funkcje	90
11.5.1	Cel lekcji	90
11.5.2	Sposoby osiągnięcia celów kształcenia	90
11.5.3	Treść – slajdy z opisem	90
11.5.4	Ćwiczenia	115
11.5.5	Opis założonych osiągnięć ucznia	115
11.6	Lekcja 6 – Model dokumentu – Document Object Model	115
11.6.1	Cel lekcji	115
11.6.2	Sposoby osiągnięcia celów kształcenia	115
11.6.3	Treść – slajdy z opisem	116
11.6.4	Ćwiczenia	127
11.6.5	Opis założonych osiągnięć ucznia	127
11.7	Lekcja 7 – Zdarzenia	128
11.7.1	Cel lekcji	128
11.7.2	Sposoby osiągnięcia celów kształcenia	128
11.7.3	Treść – slajdy z opisem	128
11.7.4	Ćwiczenia	139
11.7.5	Opis założonych osiągnięć ucznia	139
11.8	Lekcja 8 – Obsługa błędów	139
11.8.1	Cel lekcji	139
11.8.2	Sposoby osiągnięcia celów kształcenia	139
11.8.3	Treść – slajdy z opisem	140
11.8.4	Ćwiczenie	148
11.8.5	Opis założonych osiągnięć ucznia	149
11.9	Lekcja 9 – Wykorzystanie języka JavaScript	149

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



11.9.1	Cel lekcji.....	149
11.9.2	Sposoby osiągnięcia celów kształcenia	149
11.9.3	Treść – slajdy z opisem	149
11.9.4	Ćwiczenia	172
11.9.5	Opis założonych osiągnięć ucznia	173
11.10	Lekcja 10 – Biblioteka JQuery	173
11.10.1	Cel lekcji.....	173
11.10.2	Sposoby osiągnięcia celów kształcenia	173
11.10.3	Treść – slajdy z opisem	173
11.10.4	Ćwiczenia	193
11.10.5	Opis założonych osiągnięć ucznia	194
11.11	Lekcja 11 – Interdyscyplinarny projekt książki teleadresowej	194
11.11.1	Cel lekcji.....	194
11.11.2	Treść – slajdy z opisem	194
11.11.3	Opis założonych osiągnięć ucznia	200

Człowiek - najlepsza inwestycja

1 Metryka dokumentu

Szczeciński Park Naukowo Technologiczny			
Dokument	Moduł Szkoleniowy JavaScript		
Krótki opis dokumentu	Moduł szkoleniowy kursu JavaScript zawierający opis celów, treści poszczególnych lekcji, ćwiczeń oraz test końcowy sprawdzający wiedzę		
Data druku	22.10.2014	Liczba stron	200
Nazwa pliku	JavaScript Moduł szkoleniowy.docx	Status	zatwierdzony

Historia zmian

Nr wersji	Data	Opis	Działanie (*)	Autorzy
0.1	01.02.2013	Utworzenie nowego dokumentu	N	Tomasz Czajkowski, Daily Group Sp. z o.o.
1.0	12.02.2013	Wersja gotowa do publikacji	Z	Tomasz Czajkowski, Daily Group Sp. z o.o.
1.1	19.02.2013	Zmiany po pierwszych warsztatach – dodanie systemu oceniania, korelacja treści kształcenia z podstawą programową	Z	<i>Tomasz Czajkowski, Daily Group Sp. z o.o.</i>
1.2	26.02.2013	Uzupełnienie lekcji o czynności nauczyciela i ucznia	Z	<i>Tomasz Czajkowski, Daily Group Sp. z o.o.</i>
1.3	06.03.2013	Dodanie wstępu, krótkiej charakterystyki języka	Z	<i>Tomasz Czajkowski, Daily Group Sp. z o.o.</i>
1.4	07.10.2014	Dodanie lekcji 11 – projekt interdyscyplinarny	Z	<i>Tomasz Czajkowski, Daily Group Sp. z o.o.</i>

(*) Działanie: N-Nowy, Z-Zmiana, W-Weryfikacja

Człowiek - najlepsza inwestycja

2 Wstęp

JavaScript jest nazywany "językiem internetu" – to jedyny język rozumiany przez przeglądarki internetowe. Wszystkie nowoczesne serwisy internetowe działają w oparciu, lub z wykorzystaniem języka JavaScript.

JavaScript pozwala na dodawanie do stron internetowych elementów takich jak animowane menu, pokazy slajdów, galerie zdjęć itp. Dzięki wykorzystaniu tego języka strony internetowe mogą reagować na działania użytkowników. JavaScript jest także podstawą technologii AJAX – umożliwiającą interakcję z innymi systemami bez konieczności przeładowywania strony w przeglądarce. Dzięki temu witryny korzystające z tej technologii zaczynają działać jak aplikacje uruchamiane bezpośrednio w systemie operacyjnym.

W ostatnich latach język JavaScript wyrósł na podstawowy język wykorzystywany w nowoczesnych rozwiązaniach mobilnych. Jego wieloplatformowość pozwala na stworzenie jednej podstawowej wersji aplikacji i uruchamianie jej na różnych urządzeniach (np. telefonach z systemem Android, iOS itp.)

Dla osoby pragnącej tworzyć nowoczesne aplikacje internetowe, oprócz znajomości języków HTML i CSS niezbędna jest także umiejętność korzystania z JavaScript.

W poszczególnych częściach kursu szczegółowo zostaną opisane podstawowe elementy składni języka JavaScript. Zapoznanie się z ich treścią pozwoli na swobodną pracę z tym językiem przy tworzeniu nowoczesnych witryn www.

3 Cel

Celem kursu jest poznanie na poziomie podstawowym języka programowania obiektowego jakim jest JavaScript, jego składni, podstawowych poleceń i struktur danych. Dodatkowo w trakcie kursu uczestnik pozna kilka praktycznych zastosowań języka przy tworzeniu nowoczesnych, interaktywnych witryn WWW. Zostanie przedstawiona również biblioteka jQuery jako jedno z najpopularniejszych obecnie rozszerzeń języka.

4 Opis sposobu realizacji celów

10 półtoragodzinnych lekcji składających się z przypomnienia wiedzy z poprzedniej lekcji, przedstawienia materiału wraz z przykładami, ćwiczeń praktycznych. Po zakończeniu całego cyklu - przeprowadzenie egzaminu sprawdzającego wiedzę.

Dodatkowo wprowadzono lekcję 11 zawierającą część projektu interdyscyplinarnego łączącego 4 dziedziny wiedzy z zakresu informatyki (SQL, PHP, JavaScript oraz JAVA) w zakresie wzbogacenia interfejsu użytkownika oraz automatyczną walidację poprawności danych.

Człowiek - najlepsza inwestycja

5 Treści kształcenia

Treść kursu została podzielona na 10 bloków tematycznych – po jednym do każdej lekcji:

1. Wprowadzenie do kursu, podłączenie skryptów do strony www, podstawowe polecenia
2. Zmienne, typy danych i operatory
3. Instrukcje warunkowe i pętle
4. Tablice
5. Obiekty i funkcje
6. Model dokumentu – Document Object Model
7. Zdarzenia
8. Obsługa błędów
9. Wykorzystanie JavaScript
10. Framework jQuery
11. Projekt interdyscyplinarny – baza teleadresowa kolegów i koleżanek ze szkoły.

6 Opis założonych osiągnięć ucznia

Uczestnik po odbyciu kursu pozna podstawy języka JavaScript, pozna składnię i typy danych tego języka. Nauczy się wykorzystywać JavaScript w tworzeniu witryn WWW.

7 Korelacja z podstawą programową

Programowanie w języku JavaScript		
Cele kształcenia	Treści kształcenia	Podstawa programowa
- uczeń poznaje podstawowe typy danych języka JavaScript, oraz podstawowe instrukcje	Lekcja 2, 4	Uczeń korzysta z wbudowanych typów danych. Uczeń stosuje instrukcje, funkcje, procedury (...) wybranych języków programowania.
- uczeń zna podstawowe instrukcje języka, - uczeń tworzy własne funkcje i obiekty	Lekcja 3, 5	Uczeń tworzy własne typy danych. Uczeń stosuje instrukcje, funkcje, procedury (...). Uczeń tworzy własne funkcje, procedury, obiekty (...)
- uczeń poznaje model DOM jako reprezentację struktury strony w przeglądarce, poznaje jego podstawowe obiekty i uczy się je wykorzystywać - uczeń poznaje zdarzenia dostępne w trakcie przeglądania strony	Lekcja 6, 7	Uczeń stosuje skrypty wykonywane po stronie klienta przy tworzeniu aplikacji internetowych

Człowiek - najlepsza inwestycja

- uczeń potrafi obsługiwać zdarzenia w przeglądarce		
- uczeń poznaje sposoby obsługi błędów w tworzonych skryptach	Lekcja 8	Uczeń testuje tworzoną aplikację i modyfikuje jej kod źródłowy, Uczeń przestrzega zasad programowania
- uczeń poznaje praktyczne zastosowania języka JavaScript - uczeń potrafi wykorzystać JavaScript w programowaniu witryn www	Lekcja 9	Uczeń stosuje skrypty wykonywane po stronie klienta przy tworzeniu aplikacji internetowych
- uczeń poznaje podstawy biblioteki funkcji jQuery, - uczeń potrafi wykorzystać funkcje dostarczane przez bibliotekę w tworzeniu stron www	Lekcja 10	Uczeń wykorzystuje frameworki do tworzenia własnych aplikacji, Uczeń przestrzega zasad programowania

8 Sposoby osiągnięcia celów

Po odbyciu kursu uczeń powinien wykonać projekt/zadanie, które zmotywuje go do pracy indywidualnej ze środowiskiem programistycznym i bazą danych. Zadanie utrwali zdobytą na kursie wiedzę i zmusi do wykorzystania zdobytej wiedzy teoretycznej w praktyce. Zadanie powinno być sformułowane w sposób otwarty, tak aby każdy z uczniów mógł wybrać coś dla niego interesującego. Powinna też zostać dostarczona lista możliwych tematów projektu do wyboru - dla osób, które nie będą miały pomysłu na projekt.

8.1 Projekt zaliczeniowy

1. Utwórz skrypty JS które przygotują animowane menu na witrynie www.

9 Propozycje kryteriów oceny i metod sprawdzania osiągnięć ucznia

Oceną zaliczającą kurs będzie wynik testu sprawdzającego wiedzę z zakresu kursu, ocena pracy ucznia w trakcie semestru oraz przygotowanie prostego projektu (dla chętnych)

Ocenę **dopuszczającą (2)** otrzymuje uczeń, który potrafi podłączyć i uruchomić skrypt na stronie internetowej, zna podstawowe pojęcia związane z językiem JavaScript, potrafi własnymi słowami wyjaśnić podstawowe instrukcje języka.

Ocenę **dostateczną (3)** otrzymuje uczeń, który spełnia wymagania oceny dopuszczającej oraz potrafi stworzyć własne skrypty pozwalające na operowanie elementami strony internetowej. Zna

Człowiek - najlepsza inwestycja

podstawowe struktury danych w języku JavaScript (zmienne i tablice). Wykonuje ćwiczenia z niewielką pomocą nauczyciela. Potrafi wskazać podstawowe zastosowania języka JavaScript w tworzeniu stron internetowych

Ocenę **dobrą (4)** otrzymuje uczeń, który spełnia wymagania na ocenę dostateczną oraz potrafi stworzyć skrypt i podłączyć skrypty reagujące na zdarzenia na stronie internetowej, zna i potrafi wykorzystać obiektowy model programowania w języku JavaScript. Na lekcjach wykonuje samodzielnie zadane ćwiczenia. Umiejętnie korzysta z mechanizmów pozwalających na modyfikację witryny internetowej, jej elementów i stylów.

Ocenę **bardzo dobrą (5)** otrzymuje uczeń, który spełnia wymagania na ocenę dobrą oraz bardzo dobrze zna teorię omawianą na zajęciach, w trakcie wykonywania ćwiczeń wykazuje inicjatywę, potrafi wskazać kilka rozwiązań zadania, tworzy skrypty języka z wykorzystaniem dobrych praktyk programistycznych. Wykorzystuje funkcje dostępne w bibliotekach jQuery.

Ocenę **celującą (6)** otrzymuje uczeń, którego wiedza wykracza poza omawiany zakres. Np. potrafi stworzyć aplikację/grę w języku JavaScript na platformy mobilne (www, lub smartfony). Tworzy zaawansowane interaktywne witryny www, udziela się na forach i grupach dyskusyjnych poświęconych językowi JavaScript.

Kryteria oceny testu:

- 0-7 poprawnych odpowiedzi - 1
- 8-9 poprawnych odpowiedzi - 2
- 10-11 poprawnych odpowiedzi - 3
- 12-13 poprawnych odpowiedzi - 4
- 14-15 poprawnych odpowiedzi - 5

10 Test końcowy sprawdzający wiedzę

Krótki (ok 20 minut) test składający się z 15 pytań

Przykładowy test (pogrubioną czcionką zaznaczono poprawne odpowiedzi)

1. Wewnątrz jakiego elementu należy umieścić treść skryptu
 - a. <js>
 - b. <script>**
 - c. <javascript>
 - d. <scripting>
2. Jaka jest prawidłowa składnia polecenia wypisującego tekst „Witaj świecie!”
 - a. ("Witaj świecie!");
 - b. "Witaj świecie!";
 - c. response.write("Witaj świecie!")
 - d. document.write("Witaj świecie!");**

Człowiek - najlepsza inwestycja

3. Jaka jest poprawna składnia polecenia ładującego zewnętrzny plik skrypt.js?
 - a. `<script type="text/javascript" name="skrypt.js">`
 - b. `include("skrypt.js");`
 - c. **`<script type="text/javascript" src="skrypt.js">`**
 - d. `<script type="text/javascript" href="skrypt.js">`
4. W jaki sposób tworzymy funkcję w JavaScript
 - a. `function = mojaFunkcja()`
 - b. `function:mojaFunkcja()`
 - c. **`function mojaFunkcja()`**
5. Jaki jest poprawny sposób utworzenia tablicy JavaScript?
 - a. **`var tab = new Array("Jan", "Adam", "Jacek");`**
 - b. `var tab = new Array(1: "Jan", 2:"Adam", 3:"Jacek");`
 - c. `var tab = new Array:1=("Jan")2=("Adam")3=("Jacek");`
 - d. `var tab = new Array = "Jan", "Adam", "Jacek"`
6. W jaki sposób można wywołać funkcję o nazwie mojaFunkcja?
 - a. `exec mojaFunkcja()`
 - b. **`mojaFunkcja()`**
 - c. `call mojaFunkcja()`
 - d. `call function mojaFunkcja()`
7. W jaki sposób zapisać instrukcję warunkową jeśli i jest równe 5 to wykonaj instrukcje
 - a. `if i=5 then instrukcje;`
 - b. `check(i = 5) instrukcje`
 - c. **`if (i == 5) { instrukcje }`**
 - d. `if (i == 5) then { instrukcje }`
8. Jak definiujemy pętlę While
 - a. **`while (i <= 10)`**
 - b. `while i= 1 to 10`
 - c. `while(i <= 10; i++)`
9. Warunek pętli do ... while jest sprawdzany
 - a. przed każdą iteracją pętli
 - b. **po każdej iteracji pętli**
 - c. raz – na końcu pętli
 - d. raz – na początku pętli
10. Jakie zdarzenie sprawdza czy pole formularza zmieniło wartość
 - a. `onsubmit`
 - b. `onblur`
 - c. **`onchange`**
 - d. `onclick`
11. Ile parametrów można przekazać do funkcji
 - a. żadnego
 - b. tyle ile chcesz

Człowiek - najlepsza inwestycja

- c. **jeden na każdy argument funkcji**
 - d. jeden
12. Kiedy zdarzenie JavaScript nie zostanie uruchomione
- a. kiedy inne zdarzenie jest jeszcze obsługiwane
 - b. **kiedy JavaScript jest wyłączony w przeglądarce**
 - c. kiedy strona używa arkusza stylu CSS
 - d. kiedy strona uruchamiana jest lokalnie a nie na serwerze
13. Tablica w JavaScript jest:
- a. zmienną
 - b. **obiektem**
 - c. metodą
 - d. funkcją
14. W jaki sposób można zmienić tekst zawierający przecinki na tablicę
- a. `tablica = tekst.indexOf(",")`
 - b. **`tablica = tekst.split(",");`**
 - c. `tablica = tekst.trim(",");`
 - d. `tablica = tekst.substring(",");`
15. W którym obiekcie DOM dostępna jest treść strony
- a. window
 - b. browser
 - c. **document**
 - d. location

11 Lekcje

11.1 Lekcja 1 - Wprowadzenie

11.1.1 Cel lekcji


Celem lekcji jest wprowadzenie do tematyki związanej z programowaniem interaktywnych stron internetowych. Uczestnik pozna podstawowe informacje o tworzeniu stron WWW z wykorzystaniem JavaScript, historię tego języka oraz sposoby umieszczania skryptów w plikach html.

11.1.2 Sposoby osiągnięcia celów kształcenia


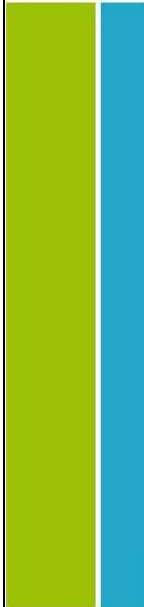

Czynności nauczyciela	Czynności ucznia
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Prowadzi z uczniami dyskusję na temat wybranych przez nich rozwiązań ćwiczeń	Prezentuje rozwiązania ćwiczeń nauczycielowi i pozostałym uczniom, tłumaczy wybrane przez siebie rozwiązanie
Podsumowuje lekcję, ocenia pracę uczniów na zajęciach.	Dokonyuje samooceny swojej pracy, omawia z nauczycielem rozwiązania ćwiczeń

Człowiek - najlepsza inwestycja

11.1.3 Treść - slajdy z opisem

<p>Slajd 1</p>		<p>W kursie przedstawimy podstawy języka JavaScript, jego podstawowe polecenia, typy danych. Zaprezentujemy sposoby jego wykorzystania w witrynach WWW. Przedstawimy bibliotekę jQuery jako obecnie najpopularniejszą i wykorzystywaną na większości stron WWW. Zaczniemy od informacji co to jest JavaScript, w jaki sposób można go podłączyć do strony www i do czego można go wykorzystać. W kolejnych lekcjach wprowadzimy podstawowe pojęcia i instrukcje języka JavaScript, pokażemy model obiektowy języka, model dokumentu html i możliwości reagowania na działania użytkownika.</p>
--------------------	---	--

Człowiek - najlepsza inwestycja

 **B2E**
BUSINESS TO EDUCATION  SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY **JavaScript**



- język skryptowy
- nie wymaga kompilowania przed uruchomieniem
- wykonywany przez program zwany interpreterem/parserem
- parser jest wbudowany w większość przeglądarek

JavaScript to język tekstowy który nie wymaga żadnej konwersji przed uruchomieniem. Inne języki takie jak Java i C++ muszą zostać skompilowane zanim będzie można je uruchomić, natomiast JavaScript jest uruchamiany od razu przez program zwany parserem lub interpreterem. Praktycznie każda nowoczesna przeglądarka zawiera w sobie parser języka JavaScript.

JavaScript nie jest używany jako samodzielny język, został bowiem zaprojektowany do łatwego osadzania w innych produktach i aplikacjach, jak na przykład przeglądarkach internetowych. Wewnątrz swojego bazowego środowiska, JavaScript może być połączony z obiektami otoczenia, w którym się znajduje, aby zapewnić nad nim programową kontrolę. Wykorzystywany w witrynach www do zapewnienia interakcji z użytkownikiem.

Człowiek - najlepsza inwestycja


**KAPITAŁ LUDZKI**
NARODOWA STRATEGIA SPÓJNOŚCIUNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

JavaScript

Krótka historia

- Stworzony przez firmę Netscape w 1995 roku.
- Głównym twórcą jest Brandon Eich
- Początkowo nazywany "Mocha", później "LiveScript"
- Po podpisaniu umowy z SUN Microsystem wprowadzono nazwę JavaScript
- Organizacja ECMA rozpoczęła pracę nad standardem języka w 1996 roku, standard został nazwany ECMAScript
- Obecna wersja to 5.1 z czerwca 2011



W roku 1995 Netscape Navigator był dominującą przeglądarką na rynku i firma zdecydowała dodać interaktywność do stron HTML dzięki lekkiemu językowi programowania. Prace nad językiem zostały zlecone Brandanowi Eichowi, który napisał pierwszą wersję języka w 10 dni. Wstępna nazwa "Mocha" została zmieniona na "LiveScript" a w wyniku porozumienia z firmą Sun Microsystems na "JavaScript". Język ten miał służyć jako klej łączący aplety wykonane w technologii Java, dodatkowo chciano wykorzystać szum medialny związany z Javą. Pod taką nazwą został opublikowany w przeglądarce Netscape Navigator 2.0B3. W 1996 roku rozpoczęły się prace nad opracowaniem standardu przez organizację standaryzującą ECMA International. Z racji tego, że Sun był właścicielem marki Java nowy standard nie mógł zostać nazwany JavaScript. W wyniku tego nazwa standardu to ECMAScript a jego implementacje to JavaScript, JScript (Microsoft) itd. Obecna wersja ECMAScript to 5.1 – jest ona implementowana przez większość nowych przeglądarek (Chrome w wersji 19+, Mozilla Firefox 4+, Internet Explorer 9+)

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
4


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Dodawanie JS do strony www


- `<script type="text/javascript" language="JavaScript 1.5">`
- `alert(12 > 6);`
- `</script>`



DAILY
 GROUP

W języku HTML za umieszczanie skryptów JS odpowiedzialny jest element `<script>` z argumentem `type text/javascript`, `application/javascript` oraz argumentem `language` o wartości `javascript`. Atrybut `language` jest jednak przestarzały i zaleca się go pomijać. Jest on pozostałością z czasów, kiedy istniał jeszcze język skryptowy VBScript żeby odróżnić skrypty napisane w tych językach

Człowiek - najlepsza inwestycja

Slajd
 5


B2E
 BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Zgodność ze starszymi przeglądarkami

```

<script type="text/javascript" language="JavaScript 1.5">
<!--
alert(12 > 6);
//-->
</script>
```



Przeglądarki nie obsługujące skryptów nie potrafią zinterpretować znacznika `<script>`. Podstawowym działaniem przeglądarki jest ignorowanie znaczników, których nie rozumieją. Takie działanie jest poprawne, gdy znacznik jest pojedynczym wyrażeniem, ale w znaczniku `<script></script>` można umieścić wiele wyrażeń. Starsze przeglądarki nie wiedzą, że mają się spodziewać znacznika zamykającego `</script>` więc po prostu wygenerują wszystkie linie występujące po otwarciu znacznika `<script>`.

Aby rozwiązać ten problem można otoczyć wyrażenia javascript znacznikami komentarza html `<!-- -->`.

linia zamykająca komentarz html rozpoczyna się znakiem komentarza JS (`//`), który oznacza polecenie zignorowania tej linii przez interpreter JavaScript, ale przeglądarki nie wspierające skryptów zinterpretują znacznik końca komentarza i rozpoczną generowanie strony od kolejnego znacznika.


Rozwiązanie to zapewniało zgodność ze starszymi przeglądarkami (obecnie bardzo starymi :) Obecnie można ten mechanizm pominąć, ale przytoczymy go tutaj dla porządku


Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI



 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
 6


B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript





JavaScript

Podłączenie zewnętrznego pliku

Zalecany sposób:

```
<script type="text/javascript" src="plik.js"></script>
```



Zaleca się jednak przechowywać skrypty w osobnych plikach i podłączać je do stron za pomocą atrybutu src

```
<script type="text/javascript" src="plik.js"></script>
```

Plik "plik.js" zawiera tylko definicję skryptów (bez dodatkowych znaczników <script>). Ogólnie przyjętą praktyką jest aby plik miał rozszerzenie ".js", chociaż nie jest to wymagane.

Zaletą takiego podejścia jest możliwość ponownego użycia skryptu na innej stronie.

Jeśli skrypt będzie zagnieżdżony w znaczniku HTML, każda strona będzie zawierała nadmiarową treść i modyfikacja skryptu będzie wymagała takich samych zmian w każdym pliku.

Z kolei podłączenie zewnętrznego skryptu to tylko jedna dodatkowa linia w każdym pliku html i zewnętrzny skrypt może być zmodyfikowany raz, aby zmiany były widoczne na wszystkich stronach, które go wykorzystują.

Plik ze skryptem zostanie umieszczony w pamięci podręcznej przeglądarki (cache) i wykorzystywany przy przechodzeniu między witrynami na stronie.

Dodatkową zaletą jest brak konieczności stosowania komentarzy i znaczników CDATA.

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Slajd
8



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Znacznik noscript

- `<noscript>`
- `<p>`
- Twoja przeglądarka nie obsługuje JavaScriptu. Aby zobaczyć stronę w pełnej funkcjonalności, zainstaluj inną przeglądarkę lub włącz opcję wyświetlania JS w przeglądarce
- `</p>`
- `</noscript>`


DAILY
 GROUP

Dodatkowym znacznikiem który możemy wykorzystać jest znacznik `<noscript></noscript>`. W przypadku nowoczesnych przeglądarek z wyłączoną obsługą JavaScript zostanie wyświetlona treść znajdująca się między znacznikami `<noscript>`. Możemy umieścić tam informację o konieczności włączenia obsługi skryptów, lub prośbę o wyświetlenie naszej strony w innej przeglądarce. Starsze przeglądarki również wyświetlą taką informację (wynika to z wcześniej wspomnianego mechanizmu działania przeglądarek - nieznany znacznik jest pomijany, a zawartość jest traktowana jako treść do wygenerowania i wysyłana do przeglądarki).

Człowiek - najlepsza inwestycja


B2E
BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Gdzie umieszczać skrypty?

- w nagłówku strony

```

<!DOCTYPE html....>
<html>
  <head>
    <title>Tytuł</title>
    <script src="plik.js"></script>
  </head>
  <body>
    <!-- treść strony -->
  </body>
</html>
    
```


DAILY
GROUP

Technicznie można umieścić znacznik `<script>` w dowolnym miejscu na stronie. Może to być sekcja `<head>` lub sekcja `<body>`. Skrypty znajdujące się w nagłówku strony zostaną wykonane jeszcze przed załadowaniem właściwej treści strony, natomiast w przypadku umieszczenia skryptu w ciele strony przed uruchomieniem skryptu zostanie wygenerowana część strony znajdująca się przed znacznikiem `<script>`

Klasyczne najlepsze praktyki wskazują umieszczenie skryptów w nagłówku strony. Zaletą takiego podejścia jest jak wspomnieliśmy wczytanie całego skryptu przed załadowaniem strony. Dodatkowo łatwiej jest innym programistom znaleźć znacznik `script`, co ułatwia debugowanie strony.

Natomiast wady takiego rozwiązania są następujące:

- ładowanie treści jest wstrzymane aż do momentu aż załadują i wykonają się wszystkie skrypty,
- skrypty nie mają dostępu do znaczników HTML w dokumencie, ponieważ nie został on jeszcze wczytany. Trzeba opóźnić wywołanie skryptów modyfikujących stronę, aż do momentu jej pełnego załadowania (można to zrobić za pomocą zdarzeń, które omówimy w późniejszych lekcjach).

Człowiek - najlepsza inwestycja




KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Slajd
 10


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript

Gdzie umieszczać skrypty?

- na końcu sekcji <body>


```


<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Tytuł</title>
  </head>
  <body>
    <!-- treść strony -->
    <script src="myscripts.js"></script>
  </body>
</html>
    
```


 DAILY
 GROUP

Niektórzy specjaliści od wydajności zalecają umieszczanie skryptów na końcu sekcji <body>. Ładowanie strony nie jest przerywane przez wczytywanie skryptów, w skrypcie nie trzeba czekać na zakończenie ładowania strony ponieważ cała treść jest już dostępna. Wadą są takie same :) strona jest dostępna dla użytkownika jeszcze zanim zostaną wgrane pliki JavaScript. Oznacza to, że jeśli np. na naszej witrynie zastosowaliśmy formularz, którego poprawne wypełnienie sprawdzamy za pomocą skryptu, to użytkownik może wysłać taki formularz zanim skrypt sprawdzający będzie dostępny. Aby rozwiązać taki problem możemy podzielić nasze skrypty na dwie grupy – jedną zawierającą ważne funkcje dołączyć do strony w nagłówku, a drugą z mniej ważnymi funkcjami w stopce strony.

 Slajd
 11



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript







Bezpieczeństwo JavaScript

- JavaScript nie zawiera żadnych mechanizmów bezpieczeństwa.
- Każdy skrypt ma takie same prawa – każdy może uzyskać dostęp do innego skryptu i zmodyfikować jego zmienne i funkcje.
- Skrypt może nadpisać nawet natywne funkcje JavaScriptu.

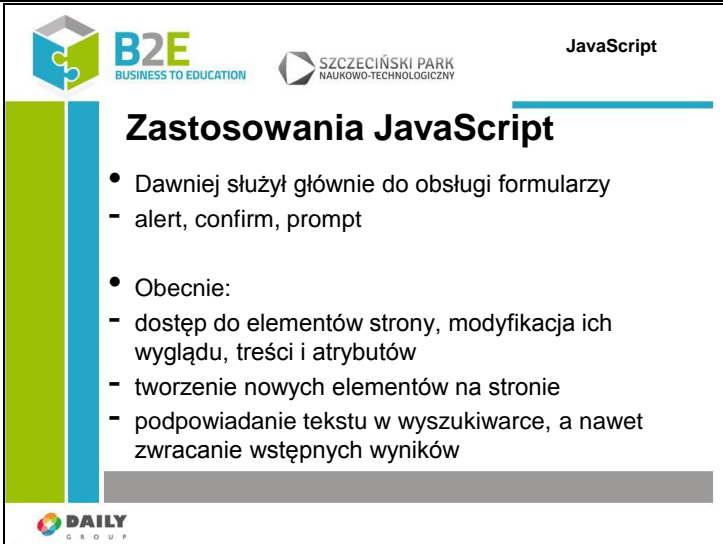

 DAILY
 GROUP

JavaScript nie zawiera żadnych mechanizmów bezpieczeństwa. Każdy skrypt ma takie same prawa – każdy może uzyskać dostęp do innego skryptu i zmodyfikować jego zmienne i funkcje. Skrypty mogą odczytywać pliki cookies, i wykorzystując prototyp funkcji można nadpisać zawartość każdej natywnej funkcji JavaScript. Dodatkowo JavaScript można wyłączyć, więc nie należy blokować dostępu

Człowiek - najlepsza inwestycja

		do strony za pomocą mechanizmów JavaScript.
Slajd 12	  <div style="text-align: right;">JavaScript</div> <h3>Dlaczego używać JavaScriptu?</h3> <ul style="list-style-type: none"> - uruchamiany po stronie klienta - ogranicza czas wykonywania strony (strona nie jest wysyłana na serwer i odsyłana do klienta po przetworzeniu) 	<p>W czasach, gdy ceny hostingu były stosunkowo wysokie, a prędkości połączenia z internetem niskie, możliwość wykonywania kodu po stronie klienta była nieoceniona. Można było reagować na działania użytkowników bez konieczności przesyłania strony do serwera. Strony z dobrze wykorzystanymi możliwościami JavaScript były bardziej interaktywne i szybsze.</p> <p>Obecnie prędkości i ceny są inne, ale brak konieczności odsyłania stron na serwer nadal powoduje że interakcja z nimi jest dużo szybsza i płynniejsza.</p>
Slajd 13	  <div style="text-align: right;">JavaScript</div> <h3>Do czego można wykorzystać JS</h3> <ul style="list-style-type: none"> - JavaScript jest prosty w implementacji - działa na komputerze użytkownika - dodaje dynamiczną zawartość na stronie (menu itp.) - może załadować treść do strony bez przesyłania jej na serwer (AJAX) - w skrypcie można sprawdzić, czy dana funkcjonalność jest dostępna w przeglądarce i odpowiednio zareagować, - javascript może być wykorzystany w celu poprawy błędów np. obsługi stylu css przez przeglądarkę 	<p>Do czego można wykorzystać JavaScript? JavaScript jest prosty w implementacji – wystarczy napisać prosty plik tekstowy i podłączyć go do strony WWW, działa po stronie klienta – zapewnia szybszą reakcję na działania użytkownika i ogranicza zużycie zasobów po stronie serwera, dodaje dynamiczną zawartość na stronie – nie trzeba przysyłać strony na serwer żeby wykonać jakąś zmianę w treści, może załadować treść w momencie gdy użytkownik jej potrzebuje – bez konieczności przeładowania całej strony (AJAX), w skrypcie można</p>

Człowiek - najlepsza inwestycja

		<p>sprawdzić, czy dana funkcjonalność jest dostępna w przeglądarce i odpowiednio zareagować, javascript może być wykorzystany w celu poprawy błędów np. obsłudze stylu css przez przeglądarkę</p>
Slajd 14		<p>Dawniej służył głównie do obsługi formularzy - alert, confirm, prompt</p> <p>Obecnie dzięki obsłudze DOM można odczytać dowolny element na stronie i zmieniać jego wygląd, zawartość i atrybuty. Możliwe jest także tworzenie nowych elementów.</p> <p>Tym samym nie musimy obecnie wykorzystywać funkcji alert, aby wyświetlić komunikat użytkownikowi. Możemy dodać dodatkowy element na stronie, w którym wpiszemy treść naszego komunikatu. JavaScript może zostać również wykorzystany do tworzenia odpowiedzi w oknach wyszukiwania, a nawet zwracania wstępnych wyników jeszcze w trakcie pisania zapytania.</p>

Slajd
15

B2E BUSINESS TO EDUCATION **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY **JavaScript**

Środowisko do pracy z JS

- JavaScript można uruchomić w dowolnej przeglądarce.
- Nie wymaga specjalistycznego środowiska programistycznego (można korzystać z dowolnego edytora tekstu)
- Edytory z kolorowaniem składni (np. notepad++)
- Środowiska programistyczne dla webdeveloperów Aptana Studio, Netbeans, Eclipse

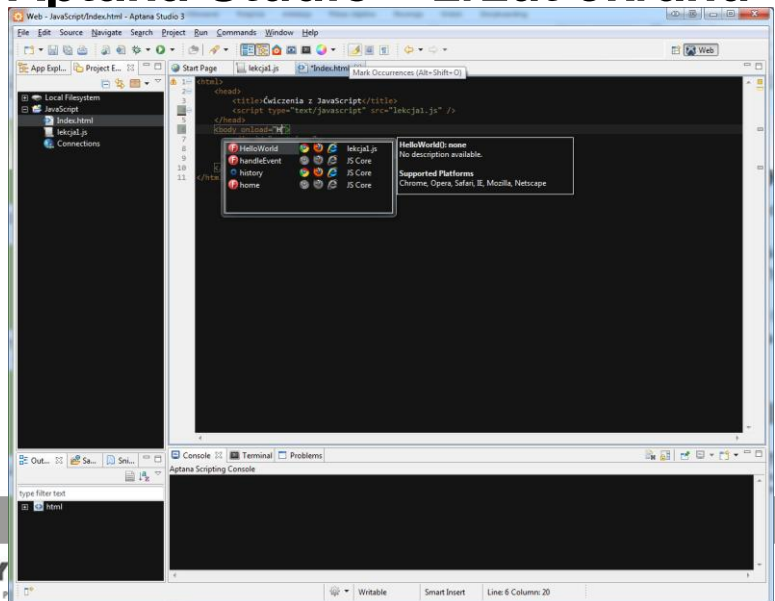
DAILY GROUP

Skrypty JS można uruchomić w dowolnej przeglądarce internetowej, a napisać w dowolnym edytorze tekstu. Dla łatwiejszego czytania kodu warto skorzystać z edytorów automatycznie kolorujących wpisywany tekst. Istnieją także wyspecjalizowane środowiska programistyczne dla webdeveloperów (obsługujące php, html i JavaScript). Przykładem takiego środowiska jest Aptana Studio – dostępna albo jako plugin do eclipse, albo jako wersja samodzielna.

Slajd
16

B2E BUSINESS TO EDUCATION **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY **JavaScript**

Aptana Studio – zrzut ekranu




DAILY GROUP

Na ekranie mamy zrzut ekranu z Aptana Studio – widać, że środowisko podpowiada nam przykładowe zdefiniowane funkcje, koloruje składnię, uzupełnia brakujące


Człowiek - najlepsza inwestycja

nawiasy itp.

Slajd
17



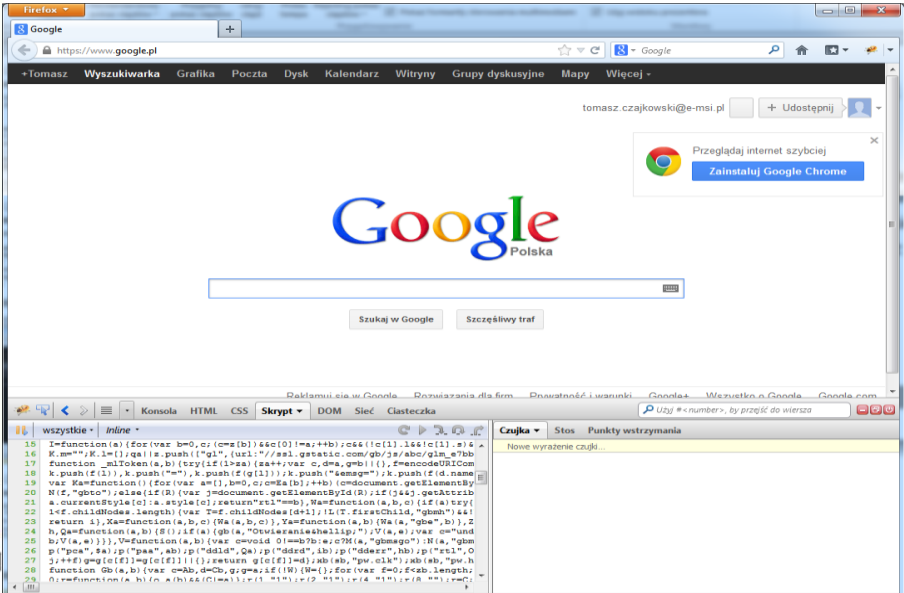
B2E
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Firebug – wtyczka dla developerów



Oprócz tworzenia i wyświetlania efektów działania skryptów, często potrzebujemy też narzędzia pozwalającego na sprawdzenie działania skryptu – np. konsoli wyświetlającej błędy, wartości zmiennych tymczasowych itp.

Można w tym celu wykorzystać dodatkowe narzędzia udostępniane przez przeglądarki internetowe (wbudowane w przeglądarkę, albo dodawane jako wtyczki). Przykładem takich narzędzi jest Firebug, dostępny dla przeglądarki Firefox (okrojone wersje istnieją też dla innych przeglądarek – np. Firebug lite dla Chrome) Przeglądarka Google Chrome ma wbudowane narzędzia dla programistów Web (dostępne w menu -> narzędzia -> narzędzia dla programistów, konsola JavaScript)

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Slajd
18



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Ćwiczenie


- Przygotuj plik index.html z podstawowymi znacznikami html i dodaj do niego znaczniki umożliwiające obsługę skryptów JS


DAILY
 GROUP

Utworzony plik będzie wykorzystywany na zajęciach jako podstawa do wykonywania ćwiczeń.

Slajd
19


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```

<html><head><title>Ćwiczenia JavaScript</title>
<script type="text/javascript">
<!--
// -->
</script>
</head>
<body>
<script type="text/javascript">
<!--
// -->
</script>
</body></html>
    
```


DAILY
 GROUP

Blok skryptu może być zagnieżdżony w nagłówku strony (skrypt zostanie wtedy wczytany i wykonany przed przetworzeniem treści strony) lub w sekcji <body> - skrypt zostanie wykonany po wczytaniu zdefiniowanej wcześniej zawartości strony. Oba rozwiązania są poprawne.

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Slajd
 20



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Ćwiczenie 2.


Wykorzystując plik utworzony w poprzednim ćwiczeniu napisz skrypt wyświetlający napis Witaj świecie. Do wyświetlenia komunikatu użyj funkcji alert().


DAILY
 GROUP

Komunikat Witaj świecie to najprostszy sposób na rozpoczęcia pracy z każdym językiem programowania.

 Slajd
 21


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Rozwiązanie

```

<html><head><title>Ćwiczenia JavaScript</title>
</head>
<body>
<script type="text/javascript">
<!--
alert("Witaj świecie");
//-->
</script>
</body></html>
    
```


DAILY
 GROUP


Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
 22



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Ćwiczenie 3.

Zmodyfikuj skrypt z poprzedniego ćwiczenia, tak aby na stronie pojawił się komunikat "Witaj imie". Zmienna imie utworz przez przypisanie wyniku wywołania funkcji prompt. Do wypisania danych na ekranie użyj funkcji `document.write()` dopisującą treść do strony html.


DAILY
 GROUP

Język JavaScript uruchamiany jest po stronie klienta (przeglądarki), więc umożliwia bezpośrednią komunikację z użytkownikiem (pozwala na wysyłanie komunikatów do użytkownika i reagowanie na wprowadzane przez niego odpowiedzi). Zmienne tworzymy przez jej zadeklarowanie a następnie przypisujemy jej wartość lub wynik obliczeń.

 Slajd
 23


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Rozwiązanie

```









<script type="text/javascript">
<!--
var imie = prompt("Podaj swoje imię: ");
document.write("Witaj "+imie);
//-->
</script>
    
```


DAILY
 GROUP

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd 24	<div>   <div>JavaScript</div> <hr/> <h2>Podsumowanie</h2> <ul style="list-style-type: none"> JavaScript jest bardzo dobrą technologią do wykorzystania na stronach WWW. Jest stosunkowo łatwym językiem do nauczania Bardzo dobrze współpracuje z innymi technologiami sieci web takimi jak html i css  </div>	JavaScript jest bardzo dobrą technologią do wykorzystania na stronach WWW. Jest stosunkowo łatwym językiem do nauczania Bardzo dobrze współpracuje z innymi technologiami sieci web takimi jak html i css
Slajd 25	<div>   <div>JavaScript</div> <hr/> <h2>Dziękuję za uwagę</h2> <div>   </div> <p><small>Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego</small></p>  </div>	

11.1.4 Ćwiczenia

Utwórz przykładową stronę www, zawierającą podstawowe znaczniki html i javascript. Utworzoną stronę będziemy wykorzystywać jako przykłady w innych lekcjach

11.1.5 Opis założonych osiągnięć ucznia

Po tej lekcji uczestnicy będą znać podstawowe zastosowania języka JavaScript związane z obsługą stron WWW, będą potrafili utworzyć prostą stronę internetową wykorzystującą JavaScript

Człowiek - najlepsza inwestycja

11.2 Lekcja 2 – Zmienne, typy danych i operatory

11.2.1 Cel lekcji

Celem lekcji jest zapoznanie uczestników z podstawowymi typami danych dostępnymi w języku JavaScript, sposobami definiowania zmiennych oraz z podstawowymi operatorami służącymi do manipulacji wartością zmiennych skryptu.

11.2.2 Sposoby osiągania celów kształcenia


Czynności nauczyciela	Czynności ucznia
podanie tematu zajęć, przypomnienie informacji z poprzednich lekcji	poznaje temat, odpowiada na pytania nauczyciela związane z materiałem z poprzednich zajęć
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Podsumowuje lekcję, ocenia pracę uczniów na zajęciach.	Dokonuje samooceny swojej pracy, omawia z nauczycielem rozwiązania ćwiczeń


11.2.3 Treść - slajdy z opisem

Slajd 1		<p>W poprzedniej lekcji wspomnieliśmy o zastosowaniach języka JavaScript i sposobach podłączania skryptów js w naszej witrynie. W tej lekcji zajmiemy się omówieniem podstawowych elementów służących do przechowywania danych oraz zobaczymy jakie podstawowe typy danych są dla nas dostępne. Zaczniemy od wprowadzenia pojęcia zmiennej.</p>
---------	---	---

Człowiek - najlepsza inwestycja

Slajd
2



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Zmienna


- Symboliczna nazwa dla wartości w aplikacji
- Musi zaczynać się literą, podkreśleniem lub znakiem \$
- Rozróżniana jest wielkość znaków
liczba != Liczba


DAILY
 GROUP

Zmiennych używa się jako symbolicznych nazw dla wartości w Twojej aplikacji. Każda zmienna ma swoją nazwę, która pozwala na identyfikację tej zmiennej w kodzie skryptu. Zmienna charakteryzuje się również typem, który określa rodzaj danych jakie przechowuje. Nazwy zmiennych, nazywane *identyfikatorami*, podporządkowane są pewnym regułom. *Identyfikator* JavaScript musi zaczynać się literą, podkreśleniem () lub znakiem dolara (\$); kolejne znaki mogą być cyframi (0-9). Ponieważ JavaScript rozróżnia duże/małe litery, litery oznaczają znaki od "A" do "Z" (wielkie litery) oraz znaki od "a" do "z" (małe litery).

Slajd
3


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Deklaracja zmiennych

Zmienną można zadeklarować na 2 sposoby:

- Zmienne lokalne: za pomocą słowa kluczowego var.
Na przykład:

```
var liczba = 10,
var nazwisko = "Kowalski"
```
- Zmienne globalne: za pomocą prostego przypisania wartości. Na przykład:


```
liczba = 10, nazwisko = "Kowalski"
```



DAILY
 GROUP

Zmienna zadeklarowana za pomocą słowa kluczowego var jest zmienną lokalną, natomiast bez słowa kluczowego zmienną globalną (więcej informacji na temat zmiennych lokalnych i globalnych będzie w późniejszych lekcjach)

Człowiek - najlepsza inwestycja

Slajd
 4



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Typy danych


- typ liczbowy
- typ łańcuchowy
- typ logiczny
- typ specjalny



DAILY
 GROUP

Typ danych określa rodzaj danych jaki przechowuje. W JavaScriptcie możemy podzielić typy na następujące rodzaje:

- typ liczbowy
- typ łańcuchowy
- typ logiczny
- typ specjalny

 Slajd
 5


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Typ liczbowy

- służy do reprezentacji liczb.
- liczby zapisywane są za pomocą literałów liczbowych, czyli ciągów znaków składających się na liczbę

```
var pi = 3.1415;
var licznik = 10;
```


DAILY
 GROUP

Typ liczbowy służy do reprezentacji liczb, przy czym nie ma występującego w klasycznych językach programowania rozróżnienia na typy całkowitoliczbowe i rzeczywiste (zmiennopozycyjne). Liczby zapisywane są za pomocą literałów (stałych napisowych, z ang. string constant, literal constant) liczbowych, czyli ciągów znaków składających się na liczbę

Obowiązują przy tym następujące zasady.

- Jeżeli ciąg cyfr nie jest poprzedzony żadnym znakiem lub jest poprzedzony znakiem +, reprezentuje wartość dodatnią, jeżeli natomiast jest poprzedzony




Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY




	<p>znakiem –, reprezentuje wartość ujemną.</p> <ul style="list-style-type: none"> • Jeżeli literał rozpoczyna się od cyfry 0, jest traktowany jako wartość ósemkowa. • Jeżeli literał rozpoczyna się od ciągu znaków 0x, jest traktowany jako wartość szesnastkowa (heksadecymalna). W zapisie wartości szesnastkowych mogą być wykorzystywane zarówno małe, jak i wielkie litery alfabetu, od A do F. • Literały mogą być zapisywane w notacji wykładniczej, w postaci X.YeZ, gdzie to część całkowita, Y — część dziesiętna, natomiast Z to wykładnik potęgi liczby 10. Zapis taki oznacza to samo, co $X.Y \cdot 10^Z$
Slajd 6	<div data-bbox="279 645 1005 1187">   <div>JavaScript</div> <hr/> <h2>Typ łańcuchowy</h2> <ul style="list-style-type: none"> • służy do reprezentacji ciągów znaków (napisów). • powinien być ujęty w znaki cudzysłowu, aczkolwiek dopuszczalne jest również wykorzystanie znaków apostrofu. <pre>var napis = "Witaj świecie!" var nazwisko = 'Kowalski';</pre>  </div> <div data-bbox="1029 645 1398 1187"> <p>Typ łańcuchowy (inaczej, typ string) służy do reprezentacji ciągów znaków (napisów). Ciągi te (inaczej, stałe napisowe) powinny być ujęte w znaki cudzysłowu, aczkolwiek dopuszczalne jest również wykorzystanie znaków apostrofu. Przykładowy ciąg ma postać: "abcdefg" lub 'abcdefg'</p> </div>

Człowiek - najlepsza inwestycja

Slajd
7



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY


B2E
BUSINESS TO EDUCATION





JavaScript


DAILY
GROUP

Ciągi mogą też zawierać poniższe znaki specjalne

- \b - backspace
- \n - nowa linia
- \r - powrót karetki
- \f - nowa strona
- \t - tabulacja
- \" - cudzysłów
- \' - apostrof
- \\ - lewy ukośnik

Człowiek - najlepsza inwestycja

Slajd 8	<div>   <div>JavaScript</div> </div> <h2>Typ logiczny</h2> <ul style="list-style-type: none"> • pozwala na określenie dwóch wartości logicznych: prawda i fałsz. • wartości tego typu są wykorzystywane przy konstruowaniu wyrażeń logicznych, porównywaniu danych itp. 	<p>Typ logiczny (boolean) pozwala na określenie dwóch wartości logicznych: prawda i fałsz. Wartość prawda jest w JavaScriptcie reprezentowana przez słowo true, natomiast wartość fałsz przez false. Wartości tego typu są używane przy konstruowaniu wyrażeń logicznych, porównywaniu danych, wskazywaniu, czy dana operacja zakończyła się sukcesem itp.</p>
Slajd 9	<div>   <div>JavaScript</div> </div> <h2>Typ specjalny</h2> <p>2 wartości:</p> <ul style="list-style-type: none"> • null • undefined 	<p>Możemy wyróżnić dwa rodzaje typów specjalnych: null i undefined. Choć podobne, nie są tożsame. typ null to specjalne słowo oznaczające wartość null (pustą), ponieważ JavaScript rozróżnia małe/duże litery, null to nie to samo co Null, NULL czy jakkolwiek inaczej. typ undefined to podstawowa właściwość, której wartość jest nieokreślona. W tym kontekście wartość undefined ma niezainicjalizowaną zmienną, zmienną, której jawnie przypisano wartość undefined, bądź też nieistniejącą właściwość obiektu.</p>

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Slajd
 10


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Zmienne a typy danych

- zmienne mogą zmieniać swój typ w trakcie działania programu

```
var mojaZmienna = "12345"
```

```
mojaZmienna = mojaZmienna - 345
```

wynik: 12000


DAILY
 GROUP

W JavaScriptcie w przeciwieństwie do wielu języków programowania zmienna może automatycznie zmieniać swój typ. Np. zmienna łańcuchowa może zostać zamieniona na zmienną liczbową lub odwrotnie w zależności od sposobu odwołania do zmiennej. Np. poniższa deklaracja przypisze zmiennej mojaZmienna wartość typu tekst "12345"


```
var mojaZmienna = "12345"
```


Pomimo tego, że ten łańcuch składa się z samych cyfr jest zmienną tekstową. Jednak JavaScript jest na tyle sprytny, że w poniższym przypisaniu zamieni typ znakowy na liczbę przed odejmowaniem i wynikową wartość (w tym przypadku liczba 12000) zapisze ponownie do zmiennej mojaZmienna

```
mojaZmienna = mojaZmienna - 345
```

Człowiek - najlepsza inwestycja

Slajd
 11


B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


DAILY
 GROUP

Niestety taka dynamiczna konwersja ma swoje wady. W przypadku przypisania `var mojaZmienna = "12345"`

`mojaZmienna = mojaZmienna + 678`

interpreter JS nie zmieni łańcucha "12345" na liczbę tylko zamieni liczbę 678 na łańcuch "678" i połączy oba łańcuchy co da w wyniku "12345678".

Aby zapobiec takiej sytuacji należy wskazać, że `mojaZmienna` jest typem liczbowym

`mojaZmienna = Number(mojaZmienna) + 678`

Taka instrukcja nazywana jest rzutowaniem na typ. W języku JavaScript dostępne są następujące funkcje do zmiany typu zmiennej:

`Boolean()` - zmienia wartość na prawdę lub fałsz

`Number()` - zmienia wartość na liczbową

`String()` - zmienia wartość na łańcuch znaków


Przy rzutowaniu na wartość logiczną łańcuch znaków o minimalnej długości 1, dowolny obiekt lub liczba większa od 0 będzie miała wartość "true", z kolei pusty łańcuch znaków, liczba 0 oraz typy specjalne null i undefined będą miały wartość "false"


Jeśli wartość rzutowana na typ liczbowy, zawiera cokolwiek oprócz cyfr to wynikiem jest NaN (not a number).

Funkcja `String()` zwraca dowolną wartość jako łańcuch znaków, nawet typy specjalne jak null i undefined.

Człowiek - najlepsza inwestycja

Slajd
 12


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Instrukcje

Instrukcja to polecenie do wykonania, zakończone średnikiem. Możemy np.


- wyświetlić okno dialogowe lub pobrać dane od użytkownika,


```
alert("Komunikat");
```

```
prompt("Podaj imię: ");
```
- pobrać element strony


```
document.getElementById("tekst");
```
- umieścić treść na stronie www


```
document.write("Komunikat");
```


DAILY
 GROUP

Instrukcję możemy potraktować jak polecenie do wykonania. Skrypt napisany w języku JavaScript to w istocie ciąg instrukcji, a zatem ciąg poleceń. Możemy zatem nakazać wyświetlenie okna dialogowego, umieścić jakąś treść na stronie WWW, zmodyfikować zawartość warstwy czy innego elementu witryny itp.

Co ważne, każdą instrukcję powinniśmy zakończyć znakiem średnika. To informacja dla interpretera, że jest to koniec instrukcji.

Przykładowe instrukcje:

```
var zmienna = 10;
```

```
var element = document.getElementById("content");
```

```
content.innerHTML = "Tekst";
```







Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI


UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY




Slajd 13	<div>   <div>JavaScript</div> <hr/> <h2>Przykłady instrukcji</h2> <p>Przykładowe instrukcje:</p> <pre>var zmienna = 10; var element = document.getElementById("content"); content.innerHTML = "Tekst";</pre>  </div>	
Slajd 14	<div>   <div>JavaScript</div> <hr/> <h2>Operatory</h2> <ul style="list-style-type: none"> • arytmetyczne, • porównywania (relacyjne), • logiczne, • przypisania,  </div>	<p>W JavaScriptcie, podobnie jak w innych językach programowania, występują operatory pozwalające na wykonywanie rozmaitych operacji. Operatory możemy podzielić na następujące grupy:</p> <ul style="list-style-type: none"> • arytmetyczne, • porównywania (relacyjne), • logiczne, • przypisania,

Człowiek - najlepsza inwestycja

Slajd
 15


B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


DAILY
 GROUP

Operatory arytmetyczne w JavaScript pozwalają na tworzenie wyrażeń numerycznych.

Operator % to operator modulo - podaje jako wynik resztę z dzielenia, operatory ++ i -- są uproszczonymi wersjami operatorów dodawania o 1 i odejmowania o 1



Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY





JavaScript

Operatory porównania

Operator	Opis	Przykład	Wynik
==	Równe	1 == 1	Prawda
===	Równe w wartości i typie	1 === '1'	Falsz
!=	Nie równe	1 != 2	Prawda
!==	Nie równe w wartości i typie	1 !== '1'	Prawda
>	Większe niż	1 > 2	Falsz
<	Mniejsze niż	1 < 2	Prawda
>=	Większe lub równe	1 >= 1	Prawda
<=	Mniejsze lub równe	2 <= 1	Falsz






Operator == i === różnią się sposobem dokładnością sprawdzania

warunek if (1 == '1') da w wyniku wartość "true" ponieważ '1' może być automatycznie zamienione na 1.

natomiast warunek if (1 === '1') da w wyniku wartość "false" ponieważ mimo że wartości mogą być uznane za takie same to zmienne są różnego typu

Slajd
17




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY


B2E
BUSINESS TO EDUCATION

JavaScript

Operatory logiczne

Operator	Opis	Przykład	Wynik
&&	I	1 == 1 && 2 == 2	Prawda
	Lub	1 == 1 2 == 3	Prawda
!	Negacja	!(1 == 1)	Falsz


DAILY
G R O U P

JavaScript wspiera 3 operatory logiczne, za pomocą których można rozszerzać instrukcję if.

Operatory w języku JavaScript sprawdzane są w kierunku od lewej do prawej. Tzn. jeśli w warunku zawierającym operator && na pierwszym miejscu będzie wyrażenie którego wartość nie będzie spełniona to sprawdzanie całego warunku zostanie przerwane i jako wynik będzie zwrócona wartość Falsz



Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY






SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Operatory przypisania

Operator	Opis	Przykład	Wynik w a
=	Proste przypisanie	a = 42	42
+=	Przypisanie z dodawaniem	a += 5	47
-=	Przypisanie z odejmowaniem	a -= 2	45
*=	Przypisanie z mnożeniem	a *= 2	90
/=	Przypisanie z dzieleniem	a /= 10	9
%=	Przypisanie z modulo	a %= 4	1



Przykładowo zamiast pisać `a = a + 5` można użyć prostszego `a += 5`.
 Powyższych operatorów można używać w połączeniu z innymi operatorami i zmiennymi

```
a = 10;
b = 25;
a += (b / 5)
```

wynikiem operacji jest 15 ($10 + (25/5)$)

Slajd
19




JavaScript

Kolejność operatorów

Priorytet	Operatory	Priorytet	Operatory
1	. [] new	10	&
2	()	11	^
3	++ --	12	
4	! ~ unary+ unary- typeof void delete	13	&&
5	* / %	14	
6	+ -	15	?:
7	<< >> >>>	16	= += -= *= /= %= <= >= >>= &= ^= !=
8	< <= > >= in instanceof	17	,
9	== != === !==		



Oprócz znajomości operatorów, niezbędna jest jeszcze wiedza na temat ich priorytetów, czyli kolejności wykonywania. Wiadomo np., że mnożenie jest „silniejsze” od dodawania, zatem najpierw mnożymy, potem dodajemy (kolejność tę można zmienić, stosując nawiasy okrągłe, dokładnie w taki sam sposób, w jaki zmienia się kolejność działań w matematyce). W JavaScriptcie jest podobnie — siła każdego operatora kolejność wykonywania jest ściśle określona.

 Slajd
20




JavaScript

Ćwiczenia





Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
 21



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Zadanie

Napisz skrypt wyświetlający na ekranie wynik działania podstawowych operacji na liczbach (dodawanie, odejmowanie, mnożenie, dzielenie). Liczby do działań pobierz od użytkownika



DAILY
 GROUP


```
var a = prompt("Podaj pierwszą liczbę: ");
var b = prompt("Podaj drugą liczbę: ");
```

```
a = Number(a);
b = Number(b);
```

```
var tekst = "Suma liczb " +
a + " i " + b + " wynosi " +
(a+b); // + "<br>";
document.writeln(tekst);
tekst = "Różnica liczb " + a +
" i " + b + " wynosi " + (a-
b); // + "<br>";
document.writeln(tekst);
tekst = "Iloczyn liczb " + a +
" i " + b + " wynosi " + (a
* b); // + "<br>";
document.writeln(tekst);
tekst = "Iloraz liczb " + a +
" i " + b + " wynosi " +
(a/b); // + "<br>";
document.writeln(tekst);
```

 Slajd
 22



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```
<script type="text/javascript">
var a = prompt("Podaj pierwszą liczbę: ");
var b = prompt("Podaj drugą liczbę: ");
a = Number(a);
b = Number(b);
var tekst = "Suma liczb " + a + " i " + b + " wynosi " + (a+b); // + "<br>";
document.writeln(tekst);
tekst = "Różnica liczb " + a + " i " + b + " wynosi " + (a-b); // + "<br>";
document.writeln(tekst);
tekst = "Iloczyn liczb " + a + " i " + b + " wynosi " + (a * b); // + "<br>";
document.writeln(tekst);
tekst = "Iloraz liczb " + a + " i " + b + " wynosi " + (a/b); // + "<br>";
document.writeln(tekst);
</script>
```



DAILY
 GROUP

Wynikiem działania funkcji prompt jest zmienna tekstowa zawierająca treść wpisaną


Człowiek - najlepsza inwestycja

przez użytkownika.
 Przed wykonaniem obliczeń matematycznych za pomocą funkcji Number musimy zmienić typ zmiennych na numeryczny.
 Do wyświetlenia wyników używamy np. funkcji document.writeln() (dodaje przejście do nowej linii po wyświetlanym tekście) lub funkcji document.write().

Slajd
23



B2E
 BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenia:


Ćwiczenie 1
 Napisz skrypt wyświetlający na ekranie napis Witaj i Twoje imię.

Ćwiczenie 2:
 Zmodyfikuj skrypt z poprzedniego ćwiczenia w taki sposób, aby przed wyświetleniem pojawiło się okno dialogowe z prośbą o podanie imienia. Wyświetl wpisane dane na ekranie




KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Człowiek - najlepsza inwestycja



UNIA EUROPEJSKA
 EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



```
document.write("witaj Tomek");
```

```
var element = document.getElementById("content");
element.innerHTML = "Witaj Tomek";
```

```
alert("Witaj Tomek");
```

Ćwiczenie 2

```
var imie = prompt("Podaj imię");
```

```
document.write("Witaj " + imie);
```

```
var element = document.getElementById("content");
element.innerHTML = "Witaj " + imie;
var tekst = "Witaj" + imie;
alert(tekst);
```

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



11.2.4 Ćwiczenia

Napisz skrypt obliczający wartość wprowadzonego wyrażenia arytmetycznego. Do obliczeń wykorzystaj funkcję eval()

11.2.5 Opis założonych osiągnięć ucznia

Uczestnik zna podstawowe typy danych i operatory języka JavaScript oraz potrafi wykorzystać je w praktyce

11.3 Lekcja 3 – Instrukcje warunkowe i pętle

11.3.1 Cel lekcji.

Celem lekcji jest wprowadzenie instrukcji pozwalających na sterowanie wykonywaniem programu. Poznamy różne warianty instrukcji warunkowych oraz pętli oraz zobaczymy praktyczne przykłady ich wykorzystania. Instrukcje wprowadzone w tej lekcji są podstawą każdego języka programowania, bez ich znajomości nie można tworzyć żadnych złożonych programów.










11.3.2 Sposoby osiągnięcia celów kształcenia

Czynności nauczyciela	Czynności ucznia
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Prowadzi z uczniami dyskusję na temat wybranych przez nich rozwiązań ćwiczeń	Prezentuje rozwiązania ćwiczeń nauczycielowi i pozostałym uczniom, tłumaczy wybrane przez siebie rozwiązanie
Podsumowuje lekcję, ocenia pracę uczniów na zajęciach.	Dokonyuje samooceny swojej pracy, omawia z nauczycielem rozwiązania ćwiczeń




11.3.3 Treść – slajdy z opisem

Slajd 1		<p>W poprzedniej lekcji wprowadziliśmy pojęcia zmiennej oraz instrukcji. Dla przypomnienia – zmienna to symboliczna nazwa dla wartości przechowywanych w trakcie działania programu, a instrukcja to polecenie sprawdzające lub ustawiające jej stan. W tej lekcji pokażemy podstawowe instrukcje warunkowe w języku JavaScript</p>
---------	--	---

Człowiek - najlepsza inwestycja


Slajd 2	<div>   <div>JavaScript</div> </div> <h2>Instrukcje warunkowe</h2> <p>Instrukcje warunkowe pozwalają na sterowanie przepływem programu wg. zdefiniowanych warunków</p> 	Instrukcje warunkowe pozwalają na sterowanie przepływem programu wg. zdefiniowanych warunków. Możemy podzielić wykonywanie programu na różne ścieżki i w zależności od wartości zmiennych wybierać te, które powinny zostać wykonane.
Slajd 3	<div>   <div>JavaScript</div> </div> <h2>Instrukcje warunkowe</h2> <p>Wyróżniamy następujące rodzaje instrukcji:</p> <ul style="list-style-type: none"> • if • if ... else • if else if ... • switch 	Wyróżniamy następujące rodzaje instrukcji: <ul style="list-style-type: none"> • if • if ... else • if else if ... • switch Pozwalają one na wykonywanie instrukcji w zależności od tego czy określony warunek został spełniony.
Slajd 4	<div>   <div>JavaScript</div> </div> <h2>Instrukcja if</h2> <pre> if (warunek) { instrukcje } lub jeśli jest tylko jedna instrukcja do wykonania if (warunek) instrukcja;</pre> 	Wyrażenia w nawiasach klamrowych zostaną wykonane tylko w przypadku, gdy warunek zostanie spełniony. Warunek musi być typu Boolean. Można go zbudować za pomocą operatorów poznanych we wcześniejszej lekcji. Nawiasy klamrowe wydzielają blok instrukcji wykonywanych po spełnieniu warunku. Po zamykającym nawiasie nie trzeba wstawiać średnika


Człowiek - najlepsza inwestycja

		kończącego polecenie JavaScript. W przypadku gdy w wyniku spełnienia warunku należy wykonać tylko 1 instrukcję nawiasy można pominąć, ale wymagany wtedy jest ; na końcu linii.
Slajd 5	   <div>JavaScript</div> <hr/> <h2>Instrukcja <i>if</i></h2> <p>Przykład:</p> <pre>var liczba = 21; if (liczba % 2 != 0) { document.writeln("Liczba nieparzysta"); }</pre>	

Instrukcja else rozszerza instrukcję if. Działa bardzo podobnie do if, ale zawarte w niej instrukcje są wykonywane tylko w przypadku, gdy warunek nie jest spełniony.

Slajd
 6


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Instrukcja *else*


```


if (warunek) {
    //instrukcje wykonywane, gdy warunek jest spełniony
}
else {
    // instrukcje wykonywane gdy warunek nie jest
    spełniony
}
    
```


DAILY
 GROUP

Instrukcja *else* rozszerza instrukcję *if*. Działa bardzo podobnie do *if*, ale zawarte w niej instrukcje są wykonywane tylko w przypadku gdy warunek nie jest spełniony.

 Slajd
 7


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript

Instrukcja *else*

Przykład:

```


liczba = 21;
if (liczba % 2 != 0) {
    document.writeln("Liczba nieparzysta");
}
else {
    document.writeln("Liczba parzysta");
}
    
```


DAILY
 GROUP


Instrukcja *else* rozszerza instrukcję *if*. Działa bardzo podobnie do *if*, ale zawarte w niej instrukcje są wykonywane tylko w przypadku gdy warunek nie jest spełniony.

Człowiek - najlepsza inwestycja

Slajd
 8



B2E
 BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Instrukcja else if

```

if (warunek1) {
    instrukcje1;
}
else if (warunek2) {
    instrukcje2;
}
.....
else if (warunekN) {
    instrukcjeN;
}
else {
    instrukcjeM;
}
    
```



Kolejna wersja instrukcji if pozwala na badanie wielu warunków. Po if może wystąpić wiele dodatkowych bloków else if.


Taka konstrukcja oznacza, że jeżeli warunek1 jest spełniony to wykonane zostaną instrukcje1, w przeciwnym wypadku gdy spełniony jest warunek2 to zostaną wykonane instrukcje2, itd. W przypadku, gdy wszystkie warunki są fałszywe, wykonywane są instrukcjeM z bloku else.

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
 9



B2E
 BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Instrukcja *switch*

```


switch (wyrażenie) {
  case wartość1:
    instrukcja1;
    break;
  case wartość2:
    instrukcja2;
    break;
  ...
  default : instrukcja;
}
```




Kolejna wersja instrukcji if pozwala na badanie wielu warunków. Po if może wystąpić wiele dodatkowych bloków else if.

Taka konstrukcja oznacza, że jeżeli warunek1 jest spełniony to wykonane zostaną instrukcje1, w przeciwnym wypadku gdy spełniony jest warunek2 to zostaną wykonane instrukcje2, itd. W przypadku, gdy wszystkie warunki są fałszywe, wykonywane są instrukcje z bloku else.

 Slajd
 10



B2E
 BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Instrukcja *switch*




```

switch (wyrażenie) {
  case wartość1:
    instrukcja1;
    break;
  case wartość2:
    instrukcja2;
    break;
  ...
  default : instrukcja;
}
```



Instrukcja wyboru switch (nazywana również instrukcją switch...case) pozwala w wygodny sposób sprawdzić ciąg warunków i wykonać różne instrukcje, w zależności od wyników porównywania. Najpierw program szuka wartości odpowiedniego wyrażenia i wykonuje załączoną instrukcję. Jeśli nie uda się znaleźć odpowiedniej wartości program szuka domyślnej instrukcji i ją wykonuje. Jeśli wartość została znaleziona to program

Człowiek - najlepsza inwestycja

		kontynuuje wykonywanie instrukcji aż do końca instrukcji switch lub do napotkania instrukcji break .
Slajd 11	<div>  B2E BUSINESS TO EDUCATION  SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY </div> <div>  DAILY GROUP </div> <div> <h2>JavaScript</h2> <hr/> <h1>Instrukcja <i>switch</i></h1> <p>Przykład:</p> <pre> kolor = "Biały"; switch (kolor) { case "Zielony": document.writeln("Wybrano zielony kolor"); break; case "Niebieski": document.writeln("Wybrano kolor niebieski"); break; case "Biały": document.writeln("Wybrano kolor biały"); break; default: document.writeln("Nieznany kolor"); } </pre> </div>	
	<p>W trakcie działania sprawdzane jest czy etykieta "Biały" znajduje się na zdefiniowanej liście etykiet instrukcji. Jeśli tak to wykonywanie programu przechodzi do tego miejsca. Jeśli pominiemy instrukcję break to oprócz napisu Wybrano kolor biały zobaczymy jeszcze napis Nieznany kolor.</p>	

Slajd
12**B2E**
BUSINESS TO EDUCATIONSZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenia

Popraw wcześniejszy przykład pobierający imię od użytkownika tak aby po naciśnięciu przycisku anuluj pojawił się tekst "Dlaczego nie chcesz podać swojego imienia?"



Popraw wcześniejszy przykład pobierający imię od użytkownika tak aby po naciśnięciu przycisku anuluj pojawił się tekst "Dlaczego nie chcesz podać swojego imienia?"

Człowiek - najlepsza inwestycja

**KAPITAŁ LUDZKI**
NARODOWA STRATEGIA SPÓJNOŚCIUNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

**B2E**
BUSINESS TO EDUCATIONSZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```
<script type="text/javascript">
<!--
var imie = prompt ("Podaj swoje imię:", "");
if (imie == null) {
document.write ("<h3>Dlaczego nie chcesz podać swojego
imienia?</h3>");
}
else {
document.write ("Witaj " + imie + "!");
}
//-->
</script>
```




Naciśnięcie przycisku anuluj w oknie wyświetlanym przez funkcję prompt zwraca jako wynik wartość null


Człowiek - najlepsza inwestycja

**KAPITAŁ LUDZKI**
NARODOWA STRATEGIA SPÓJNOŚCIUNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Slajd
14



B2E
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript


Ćwiczenie

Napisz skrypt pobierający od użytkownika 2 liczby i wyświetlający na ekranie większą z nich




DAILY
GROUP

Slajd
15



B2E
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY


JavaScript

Przykładowe rozwiązanie

```

<script type="text/javascript">
<!--
var a = prompt("Podaj pierwszą liczbę: ");
var b = prompt("Podaj drugą liczbę: ")


if (a>b)
    document.write("Liczba " + a " jest większa od liczby " +b);
else if (a == b)
    document.write("Liczby " + a +" i "+b+" są równe");
else
    document.write("Liczba " + b " jest większa od liczby " +a);
//-->
</script>
    
```




DAILY
GROUP

Człowiek - najlepsza inwestycja

Slajd
 16



B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenie

Napisz skrypt pobierający od użytkownika 2 liczby i wyświetlający wynik ich dzielenia. Zapewnij poprawną obsługę dzielenia przez 0.


DAILY
 GROUP


Celem ćwiczenia jest praktyczne zapoznanie się z instrukcjami warunkowymi. Do tego celu wykorzystamy algorytm obliczania równania kwadratowego $Ax^2 + Bx + C$. Równanie ma rozwiązanie, jeśli parametr Δ (delta) równy $B^2 - 4 \cdot A \cdot C$ jest większy od 0 lub mu równy. Jeśli Δ równa się 0, mamy jedno rozwiązanie równe $-B/(2 \cdot A)$, jeśli Δ jest większa od 0, mamy dwa rozwiązania: $x_1 = (-B + \sqrt{\Delta})/(2 \cdot A)$ i $x_2 = (-B - \sqrt{\Delta})/(2 \cdot A)$.


Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
 17



B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


DAILY
 GROUP

 Slajd
 18


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


DAILY
 GROUP

Napisz skrypt pobierający od użytkownika dzień tygodnia.

Człowiek - najlepsza inwestycja

Jeśli podana zostanie sobota, albo niedziela wyświetl komunikat "Miłego weekendu!", w przeciwnym wypadku wyświetl komunikat "Miłej pracy"

Slajd
19




JavaScript

Przykładowe rozwiązanie

```

<script type="text/javascript">
<!--
var dzien = prompt("Podaj dzień tygodnia: ");
switch(dzien){
    case "sobota" :
    case "niedziela":
        document.write("Miłego wypoczynku");
        break;
    default :
        document.write("Miłej pracy");
}
//-->
</script>
    
```



Instrukcja switch pozwala na grupowanie możliwych wariantów (pomijamy instrukcję break po poszczególnych opcjach)

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Slajd
 20


B2E
 BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenia

Napisz skrypt wyświetlający rozwiązania równania kwadratowego

$$Ax^2 + Bx + C.$$


DAILY
 GROUP


Celem ćwiczenia jest praktyczne zapoznanie się z instrukcjami warunkowymi. Do tego celu wykorzystamy algorytm obliczania równania kwadratowego $Ax^2 + Bx + C$. Równanie ma rozwiązanie, jeśli parametr Δ (delta) równy $B^2 - 4 \cdot A \cdot C$ jest większy od 0 lub mu równy. Jeśli Δ równa się 0, mamy jedno rozwiązanie równe $-B/(2 \cdot A)$, jeśli Δ jest większa od 0, mamy dwa rozwiązania: $x_1 = (-B + \sqrt{\Delta})/(2 \cdot A)$ i $x_2 = (-B - \sqrt{\Delta})/(2 \cdot A)$.


Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
21


B2E
 BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```

var A = 1, B = 1, C = -2, str = "";
if (A == 0) {
    str += "To nie jest równanie kwadratowe: A = 0!";
} else {
    delta = B * B - 4 * A * C;
    if (delta < 0) {
        str += "Delta < 0<br>";
        str += "To równanie nie ma rozwiązania w zbiorze liczb rzeczywistych!";
    } else if (delta == 0) {
        wynik = -B / 2 * A;
        str += "Rozwiązanie: x = " + wynik;
    } else {
        wynik = (-B + Math.sqrt(delta)) / 2 * A; str += "Rozwiązanie: x1 = " + wynik;
        wynik = (-B - Math.sqrt(delta)) / 2 * A; str += ", x2 = " + wynik;
    }
}
var element = document.getElementById("content");
element.innerHTML = str;
    
```


DAILY
 GROUP

Celem ćwiczenia jest praktyczne zapoznanie się z instrukcjami warunkowymi. Do tego celu wykorzystamy algorytm obliczania równania kwadratowego $Ax^2 + Bx + C$. Równanie ma rozwiązanie, jeśli parametr Δ (delta) równy $B^2 - 4 \cdot A \cdot C$ jest większy od 0 lub mu równy. Jeśli Δ równa się 0, mamy jedno rozwiązanie równe $-B/(2 \cdot A)$, jeśli Δ jest większa od 0, mamy dwa rozwiązania: $x_1 = (-B + \sqrt{\Delta})/(2 \cdot A)$ i $x_2 = (-B - \sqrt{\Delta})/(2 \cdot A)$.

 Slajd
22


B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript



JavaScript

Pętle


DAILY
 GROUP

Instrukcje warunkowe nie są jedynymi sposobami na kontrolę przepływu działania programu. Jeśli konieczne jest powtórzenie dowolnego fragmentu kodu to wykorzystać możemy pętle.


Slajd
23


B2E
 BUSINESS TO EDUCATION
 
SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY
 JavaScript

Rodzaje pętli w JavaScript

Pętle są instrukcjami pozwalającymi na wielokrotne wykonywanie tego samego kodu. JavaScript obsługuje następujące rodzaje pętli:



- for
- while
- do while



Pętle są instrukcjami pozwalającymi na wielokrotne wykonywanie tego samego kodu. JavaScript obsługuje następujące rodzaje pętli:

- for
- while
- do while


 Slajd
24


B2E
 BUSINESS TO EDUCATION
 
SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY
 JavaScript

Pętla *for*

Pętla for służy do wykonania instrukcji zawartych w nawiasach określonej liczbie razy.

```
for (wyrażenie_początkowe; warunek; wyrażenie modyfikujące) {
    instrukcje;
}
```



Wyrażenie początkowe ($i = 0$) inicjuje zmienną używaną jako licznik pętli, warunek ($i < 5$) określa warunek którego spełnienie zakończy wykonywanie pętli, wyrażenie modyfikujące ($i++$) zwiększa lub zmniejsza licznik liczby wykonań).

Powyższy przykład oznacza, dla zmiennej i równej początkowo 0 sprawdź, czy jest mniejsza od 5 i jeśli tak to wypisz jej wartość na wyjście a następnie podnieś wartość zmiennej i o 1 i ponownie sprawdź warunek. Kontynuuj aż warunek przestanie być prawdziwy.

Każde wykonanie pętli nazywamy iteracją lub przebiegiem, a zmienną zawierającą licznik pętli - zmienną iteracyjną.

Człowiek - najlepsza inwestycja

Slajd
25**B2E**
BUSINESS TO EDUCATIONSZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Pętla for

Przykład

```
for (i = 0; i<5; i++) {  
    document.write(i + "<br />");  
}
```





Dla zmiennej iteracyjnej i równej początkowo 0 sprawdź warunek – czy liczba jest mniejsza od 5. jeśli tak to wypisz na ekranie wartość tej liczby a następnie zwiększ ją o 1.

Człowiek - najlepsza inwestycja

**KAPITAŁ LUDZKI**
NARODOWA STRATEGIA SPÓJNOŚCIUNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Slajd
 26


B2E
 BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Pętla *while*

Pętla *while* służy do wielokrotnego wykonywania powtarzających się fragmentów kodu (na etapie pisania funkcji nie wiemy ile razy kod zostanie wykonany).

```
while (warunek) {
    instrukcje;
}
```


DAILY
 GROUP


W przeciwieństwie do pętli *for* nie znamy z góry liczby iteracji w tej pętli. Pętla może wykonać dowolną ilość razy (dopóki warunek jest spełniony), instrukcje zawarte w pętli mogą również nie zostać wykonane, jeśli warunek nie został spełniony przed pierwszą iteracją. Musimy też sami zadbać o zainicjalizowanie zmiennej iterującej i jej zwiększanie bądź zmniejszanie w każdej iteracji pętli.


Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
27

 **B2E**
BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Pętla *while*

Przykład

```
i = 0;
while (i <5) {
    document.writeln(i + "<br/>");
    i++;
}
```



 **DAILY**
GROUP

Do zmiennej 'i' przypisz wartość 0. Dopóki 'i' jest mniejsze od 5 wypisuj wartość 'i' na ekranie a następnie zwiększ i o 1.

Człowiek - najlepsza inwestycja

**KAPITAŁ LUDZKI**
NARODOWA STRATEGIA SPÓJNOŚCIUNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Slajd
28


 **B2E**
BUSINESS TO EDUCATION
  SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Pętla *do while*

W pętli *do ... while* warunek sprawdzany jest po wykonaniu zestawu instrukcji.

```
do {
    instrukcje;
}
while (warunek);
```

 **DAILY**
GROUP

Specjalnym rodzajem pętli *while* jest pętla *do while*.
Różnicą w porównaniu do pętli *while* jest fakt, że instrukcje zawsze zostaną wykonane przynajmniej raz - nawet jeśli warunek początkowy nie jest spełniony.

Człowiek - najlepsza inwestycja

Slajd
29


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Pętla *do while*

Przykład:


```


i = 0
do {
    document.writeln(i);
    i++;
} while (i<5)
    
```


DAILY
 GROUP

Na początku zmiennej "i" przypisz wartość "0". Wypisz wartość zmiennej "i" na ekranie i zwiększ ją o 1. Jeśli wartość zmiennej "i" jest mniejsza niż "5" wypisz jej wartość na ekranie i zwiększ o 1.

 Slajd
30


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript

Zagnieżdżanie pętli

W języku JavaScript definicje pętli możemy umieszczać wewnątrz innych pętli. np.

```

for (i = 0; i < 5; i++) {
    document.writeln("jestem w petli
zewnętrznej" + i);
    for (j = 0; j<5; j++) {
        document.writeln("jestem w petli
wewnętrznej " + j);
    }
}
    
```








DAILY
 GROUP

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 31	<div>   <div>JavaScript</div> </div> <h2>Zagnieżdżanie pętli</h2> <p>wynik:</p> <pre> Jestem w pętli zewnętrznej 1 Jestem w pętli wewnętrznej 1 Jestem w pętli wewnętrznej 2 Jestem w pętli zewnętrznej 2 Jestem w pętli wewnętrznej 1 Jestem w pętli wewnętrznej 2 Jestem w pętli zewnętrznej 3 Jestem w pętli wewnętrznej 1 Jestem w pętli wewnętrznej 2 </pre> <div>  </div>	Jak widzimy zagnieżdżanie pętli działa w następujący sposób: dla każdej iteracji pętli zewnętrznej wykonywane są wszystkie iteracje pętli wewnętrznej
Slajd 32	<div>   <div>JavaScript</div> </div> <h2>Przerywanie działania pętli</h2> <p>Do przerywania działania pętli wykorzystujemy 2 instrukcję:</p> <ul style="list-style-type: none"> • break – przerywa działanie całej pętli • continue – przerywa działanie bieżącej iteracji <div>  </div>	Do przerywania działania pętli wykorzystujemy 2 instrukcję: break – przerywa działanie całej pętli. Skrypt jest kontynuowany od kolejnej instrukcji poza pętlą. continue – przerywa działanie bieżącej iteracji. Przerywana jest bieżąca iteracja i uruchamiana jest kolejna. Należy pamiętać o ustawieniu zmiennej iteracyjnej (jeśli wykorzystujemy pętlę while lub do).

Człowiek - najlepsza inwestycja




KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI


UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Slajd
33



B2E
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenie


Ćwiczenie 6

Napisz skrypt wyświetlający na ekranie 20 razy tekst
"Będę się pilnie uczył :)"




DAILY
GROUP

Slajd
34



B2E
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązania

```


i = 0;
while (i<20) {
    document.writeln("Będę się pilnie uczył");
    i++;
}
for (i=0; i<20; i++) {
    document.writeln("Będę się pilnie uczył");
}
i = 0;
do {
    document.write("Będę się pilnie uczył");
    i++;
} while (i<20)
    
```




DAILY
GROUP

Człowiek - najlepsza inwestycja

Slajd
35



B2E
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY


JavaScript

Ćwiczenie


Napisz skrypt wyświetlający na ekranie tabliczkę mnożenia



Slajd
36



B2E
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```
for (i = 1; i <= 10; i++) {
  for (j = 1; j <= 10; j++) {
    document.write(i + " * " + j + " = " + i * j + "<br />");
  }
  document.write("<br />");
}
```



Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Slajd
 37



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript


Ćwiczenie

Napisz skrypt, który za pomocą pętli while i instrukcji continue wyświetli nieparzyste liczby z zakresu 10 – 30.


DAILY
 GROUP

 Slajd
 38


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```

var str = "";
i = 10;
while(i < 30){
    if(i % 2 == 0){
        i++;
        continue;
    }
    str += i;
    str += " ";
    i++;
}
var element = document.getElementById("element");
element.innerHTML = str;
    
```


DAILY
 GROUP

Rozwiązanie: w przypadku, gdy liczba jest parzysta (dzielenie modulo 2 daje wynik 0) pomijamy wypisywanie bieżącej wartości.

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
 39


B2E
 BUSINESS TO EDUCATION
 
SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY
 JavaScript

Dziękujemy za uwagę


 Człowiek - najlepsza inwestycja
 

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



11.3.4 Ćwiczenia

Ćwiczenia zostały przedstawione na slajdach 12-21 oraz 33-38.

W pierwszym bloku ćwiczeń zwracana jest uwaga na wykorzystanie instrukcji warunkowych, drugi ćwiczenie umożliwia praktyczne rozpoznanie sposobu działania pętli – wypisujemy w nich na ekran m.in. wartości z tabliczki mnożenia.

11.3.5 Opis założonych osiągnięć ucznia

Uczestnik po zakończeniu tej lekcji zna instrukcje warunkowe i pętle i potrafi wykorzystać je w praktyce.

11.4 Lekcja 4 – Tablice

11.4.1 Cel lekcji


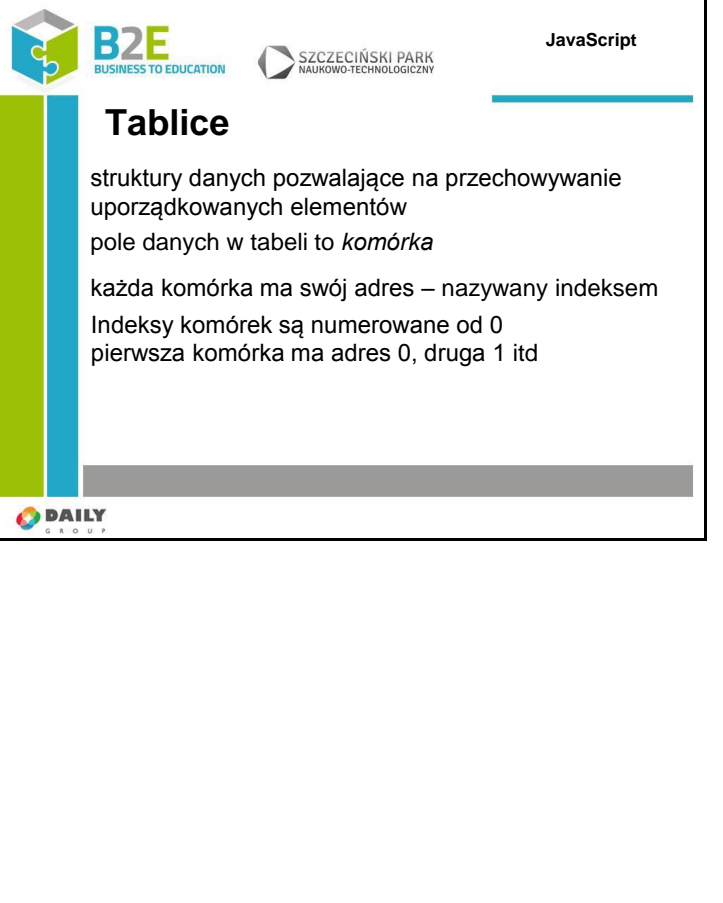
Celem lekcji pokazanie, czym są i do czego można wykorzystywać zmienne tablicowe w języku JavaScript.

11.4.2 Sposoby osiągnięcia celów kształcenia





Czynności nauczyciela	Czynności ucznia
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Prowadzi z uczniami dyskusję na temat wybranych przez nich rozwiązań ćwiczeń	Prezentuje rozwiązania ćwiczeń nauczycielowi i pozostałym uczniom, tłumaczy wybrane przez siebie rozwiązania
Podsumowuje lekcję, ocenia pracę uczniów na	Dokonyuje samooceny swojej pracy, omawia z

Człowiek - najlepsza inwestycja

11.4.3 Treść – slajdy z opisem

Slajd 1		W poprzednich lekcjach korzystaliśmy tylko z pojedynczych zmiennych. Często jednak potrzebujemy mechanizmu, który pozwoli nam na pogrupowanie zmiennych, albo nawet na zmienną do której możemy wpisać więcej niż jedną wartość. Do tego celu służą tablice.
Slajd 2		W JavaScriptcie możemy zarządzać danymi w sposób bardziej zaawansowany niż za pomocą zmiennych. Jednym z przykładów są tablice – można powiedzieć, że są to kolekcje zmiennych uporządkowane i pogrupowane razem. Poszczególne elementy w tabeli nazywamy komórkami, adres każdej komórki indeksem. Prostą tablicę można sobie wyobrazić jako szafkę, w której każda szuflada jest komórką, a numer tej szuflady indeksem komórki. Indeksy w tabelach JavaScript numerowane są od 0 – pierwsza komórka w tabeli ma indeks 0, druga 1 itd.

Człowiek - najlepsza inwestycja

Slajd 3	<div>   <div>JavaScript</div> <hr/> <h2>Tablice</h2> <p>Tworzenie nazw tablic</p> <ul style="list-style-type: none"> • rozpoczyna się od małej lub wielkiej litery, znaku _ lub znaku \$ • nie może zawierać operatorów matematycznych (+ -), znaków punktacyjnych (takich jak !, &) albo spacji • na drugim i dalszych miejscach może zawierać cyfry • słowa kluczowe języka nie mogą być użyte jako nazwa </div>	<p>Tworzenie nazw tablic rozpoczyna się od małej lub wielkiej litery, znaku _ lub znaku \$ nie może zawierać operatorów matematycznych (+ -), znaków punktacyjnych (takich jak !, &) albo spacji na drugim i dalszych miejscach może zawierać cyfry słowa kluczowe języka nie mogą być użyte jako nazwa</p>
Slajd 4	<div>   <div>JavaScript</div> <hr/> <h2>Tworzenie tablic</h2> <p>Dwa sposoby</p> <ul style="list-style-type: none"> • przez deklarację zmiennej w nawiasach kwadratowych var tablica = ["Janek", "Franek", "Bronek"]; • przez wywołanie konstruktora tablicy var tablica = new Array("Janek", "Franek", "Bronek"); </div>	<p>Tablicę możemy utworzyć na dwa sposoby – za pomocą nawiasów kwadratowych podając wewnątrz nawiasów komórki tablicy, lub wywołując konstruktor obiektu Array, który jest reprezentacją tablicy w JavaScript (o obiektach i konstruktorach powiemy sobie później). Efektem działania obu instrukcji będzie zmienna tablica zawierająca 3 imiona – Janek, Franek i Bronek.</p>







Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI


UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY




Slajd 5	<div data-bbox="284 190 997 728">   <div data-bbox="845 212 949 257">JavaScript</div> <h3 data-bbox="406 291 861 347">Przykłady tworzenia tablic</h3> <pre data-bbox="383 347 845 504"> var tablica = [1,2,3,4,5,6,7,8,9,10]; var tablica = [1, "Janek", 2, "Franek"]; var tablica = ["Janek", "Franek", "Bronek"]; var tablica = []; </pre>  </div>	<p>Wartość każdej komórki może być dowolnego typu. Przykładowo możemy utworzyć tablicę zawierającą liczby całkowite od 1 do 10, wcześniej widzieliśmy przykład deklaracji tablicy zawierającej ciągu znaków. W tablicy możemy też umieścić elementy różnego typu – np. liczby i ciągu znaków. Notacja z wykorzystaniem nawiasów kwadratowych umożliwia pomijanie elementów w tablicy – komórki pominięte będą miały wartość undefined. Możemy także utworzyć pustą tablicę i dodawać do niej elementy w trakcie działania programu.</p>
Slajd 6	<div data-bbox="284 996 997 1534">   <div data-bbox="845 1019 949 1064">JavaScript</div> <h3 data-bbox="406 1097 861 1153">Przykłady tworzenia tablic</h3> <pre data-bbox="383 1153 869 1265"> var tablica = new Array(); var tablica = new Array(1,2,3,4,5,6,7,8,9,10); var tablica = new Array(5); </pre>  </div>	<p>w konstruktorze tablicy możemy nie podawać żadnych argumentów – zostanie wtedy utworzona pusta tablica, możemy podać argumenty oddzielone przecinkami – zostaną one wtedy wpisane jako wartości komórek tablicy. Trzecia przykładowa konstrukcja ma trochę inne działanie. Zamiast spodziewanego utworzenia tablicy z jedną komórką o wartości 5, zostanie utworzona tablica pięcioelementowa w której komórki będą miały wartość undefined</p>

Człowiek - najlepsza inwestycja

Slajd
 7


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Odczyt i zapis danych

```

var zmienna = tablica[5];
document.write(tablica[5]);
if (tablica[5] == wartość) {
    instrukcje;
}
tablica[5] = wartość;
var indeks = 6;
tablica[indeks] = nowa_wartość;
    
```


DAILY
 GROUP

Odczyt danych z tablicy jest prosty. Aby przypisać wartość z danej komórki tablicy wystarczy podać jej indeks.

Nie musimy oczywiście korzystać z dodatkowych zmiennych aby wyświetlić lub wykorzystać wartość komórki. Możliwe jest bezpośrednie operowanie na komórkach tabeli.

Aby zapisać wartość do komórki tablicy wykonujemy operację przypisania na tablicy z wykorzystaniem indeksu komórki, którą chcemy zmodyfikować.


Wartości indeksu nie musimy podawać w trakcie pisania kodu – możemy wykorzystać do tego zmienną i ustawić wartość w trakcie działania programu.

Możemy jako indeks tablicy wykorzystać komórkę innej tablicy

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Tablice asocjacyjne

Rozważmy taką tablicę:

```
var wypłaty = new Array("Janek", 2000, "Franek", 3000, "Bronek", 2500);
var wypłaty = new Array();
wypłaty["Janek"] = 2000;
wypłaty["Franek"] = 3000;
wypłaty["Bronek"] = 2500;
```












Używanie indeksów numerycznych jest dobre, jeśli operujemy na niewielkiej liczbie danych.

Jeśli danych jest więcej lub jeśli dane mają dodatkowe znaczenie, użycie numerów aby pobrać wartość z tabeli zaczyna robić się kłopotliwe.

Rozważmy tablicę z wypłatami w następującej postaci: w pierwszej komórce jest identyfikator osoby (imię) a w kolejnej jego zarobki. Możliwe jest wyszukanie zarobków Franka w następujący sposób – sprawdź, czy wartość pierwszej komórki to „Franek” – jeśli tak to pobierz i wyświetl wartość kolejnej komórki. W przeciwnym wypadku przeskocz o 2 komórki i wykonaj sprawdzenie ponownie.



Na szczęście javascript wspiera przypisywanie nazw elementom komórek. Możemy zatem utworzyć następującą tablicę, gdzie jako indeksy pól podamy identyfikatory osób a jako wartość ich wypłatę. Wyświetlenie dochodów Franka to w tym przypadku proste zapytanie `wypłaty["Franek"]`.

JavaScript nie tworzy w tym przypadku nowych elementów tablicy tylko dodaje nowe właściwości (pary typu klucz – wartość) do obiektu przechowującego tablicę. Zapytanie o długość tablicy zwróci w tym przypadku 0 – bo utworzyliśmy pustą tablicę i dodaliśmy do niej 3 właściwości

Slajd 9	<div>   <div>JavaScript</div> </div> <h2>Tablice asocjacyjne</h2> <p>Tworzenie tablicy asocjacyjnej z wypełnieniem danymi</p> <pre>var wypłaty = new Array({ 'Janek' : 2000, 'Franek' : 3000, 'Broniek' : 2500 })</pre> <div>  </div>	<p>Tworząc tabelę asocjacyjną zainicjowaną początkowym zestawem danych do konstruktora obiektu przekazujemy definicję wartości, które chcemy utworzyć. Definicja zawiera pary klucz wartość (oddzielone dwukropkiem, a nie znakiem równości) umieszczone w nawiasach klamrowych</p>
Slajd 10	<div>   <div>JavaScript</div> </div> <h2>Tablice asocjacyjne</h2> <p>Na poprzednim ekranie utworzyliśmy 1 elementową tablicę z wartościami.</p> <p>Dostęp do poszczególnych elementów możliwy jest z wykorzystaniem nawiasów [] i podaniem indeksu elementu w tabeli oraz nazwy właściwości.</p> <p>np. wypłaty[0]['Janek']</p> <div>  </div>	<p>Na poprzednim ekranie utworzyliśmy 1 elementową tablicę z wartościami. Dostęp do poszczególnych elementów możliwy jest z wykorzystaniem nawiasów [] i podaniem indeksu elementu w tabeli oraz nazwy właściwości. np. wypłaty[0]['Janek'], gdzie 0 oznacza 1 element w tablicy.</p>
Slajd 11	<div>   <div>JavaScript</div> </div> <h2>Ćwiczenie 1</h2> <p>Utwórz tablicę przechowującą liczby od 1 – 100 w taki sposób, że w komórkach od 0-49 będą wartości 1-50 a w komórkach od 50-99 wartości 100 – 51.</p> <p>Wyświetl zawartość tabeli na ekranie.</p> <div>  </div>	<p>Ćwiczenie: Utwórz tablicę przechowującą liczby od 1 – 100 w taki sposób, że w komórkach od 0-49 będą wartości 1-50 a w komórkach od 50-99 wartości 100 – 51. Wyświetl zawartość tabeli na ekranie.</p>

Człowiek - najlepsza inwestycja

Slajd
12




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenie 1

```

for (i = 0; i<100; i++) {
  if (i<50) {
    tablica[i] = i+1;
  }
  else {
    tablica[i] = 100 - (i - 50);
  }
}
for(i = 0; i<100; i++) {
  document.write(tablica[i] + "<br />");
}
    
```


DAILY
GROUP

Slajd
13



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript


JavaScript


Tablice wielowymiarowe


DAILY
GROUP

Człowiek - najlepsza inwestycja

Slajd
 14


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Do tej pory wszystkie pokazane tablice miały tylko 1 wymiar. Można je było przedstawić za pomocą wektora z wartościami komórek tabeli.

Do pojedynczej wartości można było dostać się podając jej indeks.

Czy możliwe jest zatem podanie więcej niż jednego indeksu, aby wskazać element tablicy? JavaScript nie wspiera niestety

wielowymiarowych tablic, ale jest sposób na zasymulowanie takiej funkcjonalności.

W języku JavaScript wartością komórek tablicy mogą być inne tablice. Możemy utworzyć tabele zawierające w komórkach inne tabele, które mogą zawierać kolejne tabele itd.


Dostęp do elementów w takich tabelach jest następujący: `tablica[x][y]` – Z tablicy pobierz wartość komórki oznaczonej indeksem x a z niej wartość komórki oznaczonej Y


Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
 15


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Użycie pętli for




```

var tablica = [];
for (var i = 0; i < 100; i++) {
    tablica[i] = "wartość "+i;
}
var dlugosc_tablicy = tablica.length;
for (var i = 0; i < dlugosc_tablicy; i++) {
    document.write(tablica[i] + "<br />");
}
  
```

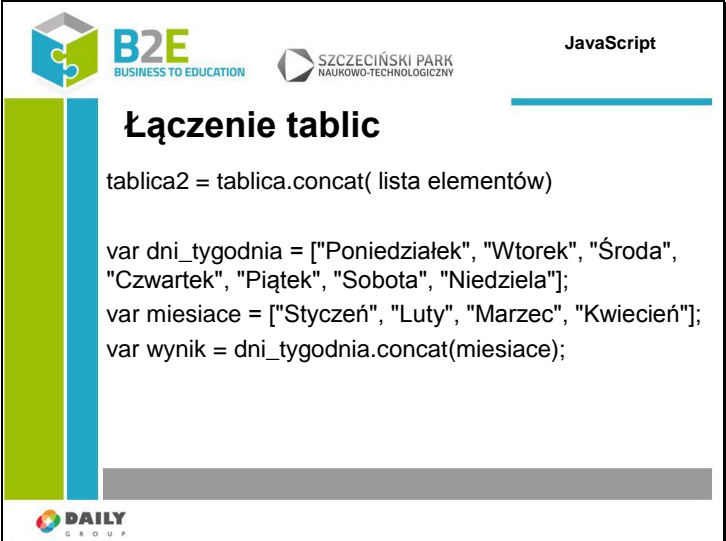
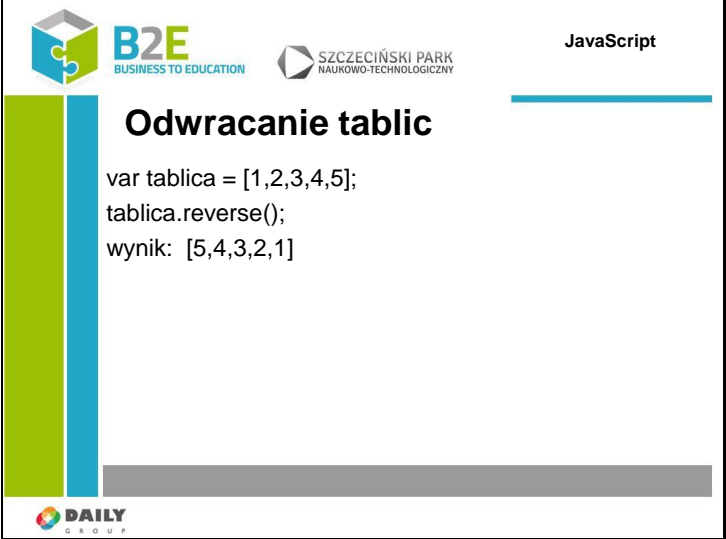

DAILY
 GROUP










Bezpośrednie odwołania do elementów tablicy za pomocą indeksu są wystarczające tylko w przypadku, gdy tablica zawiera niewiele elementów. Jeśli danych jest dużo, lub z góry nie wiemy ile (częsta sytuacja, gdy nasz program otrzyma dane z zewnątrz) wymagane będzie wprowadzenie automatycznego przetwarzania tablicy. W takim celu możemy wykorzystać pętle. Pierwsza przedstawiona pętla dodaje 100 elementów do nowo utworzonej tablicy. Odczyt danych z tablicy możliwy jest w następujący sposób. Pętla for może zostać wykorzystana w przypadku gdy znamy długość tablicy. W przypadku tablic z indeksem numerycznym długość tablicy możemy pobrać za pomocą właściwości length. Właściwość ta przyjmuje wartość najwyższego indeksu w tablicy + 1. Tworząc pętlę warto obliczyć długość tablicy przed wywołaniem pętli. Przypisując liczbę elementów w tablicy do zmiennej operacja obliczenia wartości właściwości tablica.length wykonywana jest tylko raz, gdybyśmy w definicji pętli ustawili warunek `i < tablica.length` długość tablicy sprawdzana byłaby za każdą iteracją.

Człowiek - najlepsza inwestycja


Slajd 16	<div>   JavaScript </div> <h2>Pętla for in</h2> <pre>for (var indeks in tablica) { document.write(tablica[indeks] + "
"); }</pre> 
-------------	---







Człowiek - najlepsza inwestycja

		<p>elementy</p> <ul style="list-style-type: none"> - join – zwraca łańcuch znaków z zawartością wszystkich komórek oddzielonych znakiem separatora
Slajd 18		<p>Wynikiem połączenia tablic jest nowa tablica składająca się z wszystkich elementów tablicy dni_tygodnia oraz wszystkich elementów tablicy miesiace.</p>
Slajd 19		<p>Do odwracania tablic służy funkcja reverse. Nie przyjmuje ona żadnych argumentów i w działa na tablicy, na rzecz której została wywołana (nie jest tworzona tablica wynikowa).</p>

Slajd 20	<div>   <div>JavaScript</div> </div> <h2>Dodawanie i usuwanie elementów</h2> <ul style="list-style-type: none"> • pop - tablica.pop() • push – tablica.push(lista elementów) • shift – tablica.shift() • unshift – tablica.unshift(lista elementów) <div>  </div>	
Slajd 21	<div>   <div>JavaScript</div> </div> <h2>Sortowanie tablicy</h2> <p>Aby posortować tablicę należy wywołać na niej funkcję sort()</p> <pre>var liczby = [5,7,3,5,2,7,5]; liczby.sort(); wynik: [2,3,5,5,5,7,7]</pre> <div>  </div>	<p>Sortowanie tablicy to uporządkowanie jej elementów. Funkcja zmienia porządek elementów w tablicy nadpisując oryginalną wersję.</p> <p>Tablice zawierające liczby zostaną uporządkowane rosnąco, natomiast tablice zawierające ciągi znaków – alfabetycznie.</p>
Slajd 22	<div>   <div>JavaScript</div> </div> <h2>Sortowanie cd.</h2> <pre>tablica.sort(nazwa_funkcji);</pre> <p>Funkcja musi zwracać:</p> <ul style="list-style-type: none"> • wartość mniejszą od 0, jeśli pierwszy argument jest mniejszy od drugiego • wartość 0, jeśli argumenty są równe, • wartość większą od 0, jeśli pierwszy argument jest większy od drugiego <div>  </div>	<p>Jeśli będziemy chcieli posortować tablice w sposób odmienny od domyślnego do wywołania metody sort, należy przekazać nazwę funkcji porównującej 2 elementy. Taka funkcja będzie otrzymywać 2 elementy tablicy, jako wynik musi zwracać:</p> <ul style="list-style-type: none"> • wartość mniejszą od 0, jeżeli pierwszy argument jest mniejszy od drugiego • wartość równą 0,

Człowiek - najlepsza inwestycja

		<p>jeżeli argumenty są równe</p> <ul style="list-style-type: none"> wartość większą od 0, jeżeli pierwszy argument jest większy od drugiego <p>Aby odwrócić kolejność sortowania możemy albo zmodyfikować funkcję porównującą, albo na wyniku sortowania zastosować metodę <code>reverse()</code></p>
Slajd 23	 <p>JavaScript</p> <h3>Sklejanie wartości (join)</h3> <pre>var tablica = ["Janek", "Franek", "Bronek"]; document.write(tablica.join(" i "));</pre> <p>wynik: Janek i Franek i Bronek bez separatora: Janek,Franek,Bronek</p>	<p>Czasem przydatna jest możliwość zmiany wszystkich elementów tablicy w jeden łańcuch znaków. Można do tego wykorzystać metodę <code>join()</code> jako argument podając separator rozdzielający poszczególne elementy tablicy. W podanym przykładzie jako separator wykorzystaliśmy łańcuch składający się ze znaku spacji na początku i końcu oraz spójnika <code>i</code>. Wynikiem działania skryptu będzie tekst: Janek i Franek i Bronek. Jeśli nie podamy żadnego separatora domyślnie zostanie przyjęty znak <code>,</code></p>

Slajd 24	<div>   <div>JavaScript</div> </div> <h2>Dzielenie ciągu na elementy</h2> <pre>var tekst= "Janek i Franek i Bronek"; document.write(tekst.split(" i "));</pre> <p>wynik: Janek,Franek,Bronek</p> <div>  </div>	<p>Odwrótną metodą jest metoda <code>split()</code> – operująca na ciągu znaków. Wynikiem działania jest tablica elementów powstałych z podzielenia ciągu wejściowego na fragmenty oddzielone znakiem separatora.</p>
Slajd 25	<div>   <div>JavaScript</div> </div> <h2>Ćwiczenie</h2> <p>Utwórz tablicę przechowującą 50 kolejnych wartości całkowitych i wyświetl jej zawartość na ekranie. W obu operacjach użyj pętli typu <code>while</code>.</p> <div>  </div>	

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Slajd
26


B2E
 BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Przykładowe rozwiązanie


```

var tablica = new Array();
i = 0;
while (i<50) {
    tablica[i] = i+1;
    i++;
}
j = 0;
while (j<tablica.length) {
    document.writeln(tablica[j]);
    j++;
}
    
```


DAILY
 GROUP

 Slajd
27



B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenia

Wykonaj sortowanie tablicy zawierającej wartości całkowite, tak aby najpierw znalazły się wartości niepodzielne przez 3 w porządku malejącym, a następnie wartości podzielne przez 3 w porządku rosnącym.


DAILY
 GROUP


Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
 28



B2E
 BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


DAILY
 GROUP

 Slajd
 29


B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript




DAILY
 GROUP

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Slajd
30

 **B2E**
BUSINESS TO EDUCATION
  **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY



JavaScript

Ćwiczenia

Z tablicy zawierającej dni tygodnia przesunij niedzielę na początek.

 **DAILY**
GROUP


Slajd
31

 **B2E**
BUSINESS TO EDUCATION
  **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```
var dni = ["Poniedziałek", "Wtorek", "Środa", "Czwartek", "Piątek", "Sobota", "Niedziela"];
var niedziela = dni.pop();
dni.unshift(niedziela);
```

 **DAILY**
GROUP

11.4.4 Ćwiczenia

Ćwiczenie zostało zaprezentowane na slajdzie 10 oraz na slajdach 25 - 31.

Człowiek - najlepsza inwestycja



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

11.4.5 Opis założonych osiągnięć ucznia

Po ukończeniu lekcji uczestnik będzie znał sposoby tworzenia tablic w języku JavaScript, a także podstawowe metody służące do pracy z tablicami

11.5 Lekcja 5 – Obiekty i funkcje

11.5.1 Cel lekcji

Język JavaScript jest językiem obiektowym – w tej lekcji poznamy definicję obiektu oraz funkcji.





11.5.2 Sposoby osiągania celów kształcenia

Czynności nauczyciela	Czynności ucznia
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Prowadzi z uczniami dyskusję na temat wybranych przez nich rozwiązań ćwiczeń	Prezentuje rozwiązania ćwiczeń nauczycielowi i pozostałym uczniom, tłumaczy wybrane przez siebie rozwiązanie
Podsumowuje lekcję, ocenia pracę uczniów na zajęciach.	Dokonuje samooceny swojej pracy, omawia z nauczycielem rozwiązania ćwiczeń

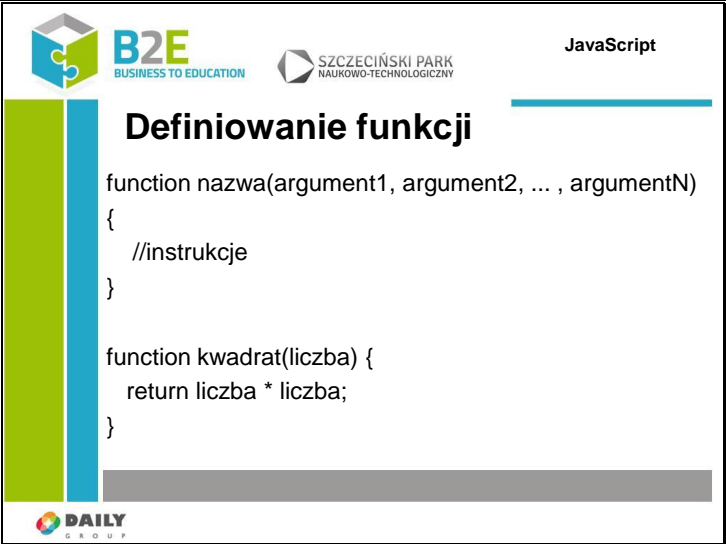
11.5.3 Treść – slajdy z opisem

Slajd 1	 <p>The slide is a title slide for a JavaScript lesson. It features a large green and blue vertical bar on the left. The text 'JavaScript' is prominently displayed in the center, with 'Obiekty i funkcje' below it. Logos for B2E, Szczeciński Park Naukowo-Technologiczny, and DAILY are at the top. At the bottom, there are logos for 'KAPITAŁ LUDZKI' and 'UNIA EUROPEJSKA'.</p>	<p>We wcześniejszych lekcjach wprowadziliśmy pojęcia zmiennej, pokazaliśmy istniejące typy danych w języku JavaScript i omówiliśmy tablice jako specjalny rodzaj zmiennych. JavaScript pozwala nam na tworzenie własnych typów danych zwanych obiektami i zbiorów instrukcji, które mogą wykonywać jakąś część programu zwanych funkcjami. W tej lekcji przybliżymy te pojęcia.</p>
------------	--	---









Człowiek - najlepsza inwestycja

Slajd 2	<div>   <div>JavaScript</div> </div> <h2>Co to jest funkcja</h2> <p>Funkcja jest zestawem instrukcji, które wykonują określone zadanie lub obliczają wartość.</p> <p>Funkcje są sekcjami kodu, które można wywoływać z innych miejsc w kodzie.</p>	<p>Funkcja to jeden z podstawowych bloków w JavaScript. Funkcja jest zestawem instrukcji, które wykonują określone zadanie lub obliczają wartość.</p> <p>Funkcja wykonuje zdefiniowane w ciele funkcji instrukcje a następnie zwraca wynik do miejsca, z którego została wywołana.</p> <p>Aby wykorzystywać funkcje trzeba ją wcześniej zdefiniować.</p>
Slajd 3	<div>   <div>JavaScript</div> </div> <h2>Definiowanie funkcji</h2> <p>Definicja funkcji (zwana też deklaracją funkcji) składa się ze słowa kluczowego function i następujących elementów:</p> <ul style="list-style-type: none"> • nazwy funkcji • listy argumentów w nawiasach, oddzielonych przecinkami) • instrukcji definiujących funkcję umieszczonych w nawiasach klamrowych. 	<p>Definicja funkcji (zwana też deklaracją funkcji) składa się ze słowa kluczowego function i następujących elementów:</p> <ul style="list-style-type: none"> • nazwy funkcji • listy argumentów w nawiasach, oddzielonych przecinkami) • instrukcji definiujących funkcję umieszczonych w nawiasach klamrowych. <p>Nazwa może się składać z liter, cyfr oraz znaków podkreślenia i dolara, nie może jednak zaczynać się od cyfry. We wszystkich współczesnych implementacjach JavaScriptu może też zawierać znaki spoza alfabetu łacińskiego (np. polskie znaki diakrytyczne), choć z reguły się ich nie stosuje.</p>

Człowiek - najlepsza inwestycja

		<p>Funkcja może, ale nie musi zawierać słowa kluczowego return (zwracającego wartość funkcji). Gdy w definicji funkcji brakuje instrukcji return wartością zwracaną przez funkcję jest "undefined"</p>
<p>Slajd 4</p>	 <pre> function nazwa(argument1, argument2, ... , argumentN) { //instrukcje } function kwadrat(liczba) { return liczba * liczba; } </pre>	<p>Funkcja posiada 1 argument nazwany liczba i zwraca za pomocą instrukcji return wartość argumentu pomnożonego przez samego siebie. Typy podstawowe (np, number) są przekazywane do funkcji za pomocą wartości. Oznacza to, że jeśli funkcja zmodyfikuje wartość parametru to zmiana ta nie będzie widoczna poza funkcją, przekazanie do funkcji typu złożonego (np. tablicy, lub zdefiniowanego przez użytkownika obiektu) spowoduje przekazanie tylko referencji do obiektu. Wszystkie modyfikacje wykonane przez funkcję będą widoczne także poza nią.</p>

Człowiek - najlepsza inwestycja


Slajd 5	<div>   JavaScript </div> <h2>Argumenty funkcji</h2> <ul style="list-style-type: none"> - brak z góry zdefiniowanej liczby - służą do przekazania wartości do funkcji w celu dowolnego przetwarzania i wykonywania na nich instrukcji <div>  </div>	Argumenty funkcji <ul style="list-style-type: none"> - brak z góry zdefiniowanej liczby - służą do przekazania wartości do funkcji w celu dowolnego przetwarzania i wykonywania na nich instrukcji
Slajd 6	<div>   JavaScript </div> <h2>Zwracanie wartości przez funkcję</h2> <ul style="list-style-type: none"> - słowo kluczowe return <pre>function nazwa (argumenty) { //instrukcje return wartość; }</pre> <div>  </div>	Jeśli w ciele funkcji wystąpi instrukcja return to przetwarzanie funkcji zostaje przerwane i do miejsca wywołania zwracana jest wartość występująca po słowie return .
Slajd 7	<div>   JavaScript </div> <h2>Zasięg (widoczność) zmiennych</h2> <p>Zasięg zmiennej to miejsce w którym zmienna jest ważna i można się do niej bezpośrednio odwoływać.</p> <p>Zmienne mogą być:</p> <ul style="list-style-type: none"> - globalne, - lokalne <div>  </div>	Kiedy zadeklarujemy zmienną poza ciałem funkcji jest ona nazywana zmienną globalną, ponieważ jest dostępna z poziomu każdego innego fragmentu kodu w bieżącym dokumencie, Jeśli zadeklarujemy zmienną w ciele funkcji jest ona nazywana zmienną lokalną - widoczną tylko w obrębie tej funkcji (i ewentualnych podfunkcji w niej zdefiniowanych)


Człowiek - najlepsza inwestycja

		<p>Aby zadeklarować zmienną lokalną należy poprzedzić ją słowem kluczowym var. Wartość takiej zmiennej zostanie zapomniana po opuszczeniu funkcji, a jeśli istniała wcześniej zmienna globalna o takiej samej nazwie, zachowa ona swoją wartość sprzed wywołania funkcji. Zmienna zadeklarowana w funkcji bez wykorzystania słowa kluczowego var jest zmienną globalną.</p>
Slajd 8		
Slajd 9		<p>JavaScript zawiera kilka predefiniowanych funkcji o zasięgu globalnym</p> <ul style="list-style-type: none"> • eval • isNaN • parseInt oraz parseFloat • Number i String

Człowiek - najlepsza inwestycja

Slajd
10

 **B2E**
BUSINESS TO EDUCATION


 **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY

JavaScript

Funkcja eval

Funkcja eval przekazuje łańcuch znaków do kompilatora i wykonuje rezultat

```
eval(wyrażenie);
```


 **DAILY**
GROUP


Funkcja eval przekazuje łańcuch znaków do kompilatora i wykonuje rezultat

```
eval(wyrażenie);
```

Nie zaleca się nadmiernie wykorzystywać tej funkcji, ponieważ utrudnia, a czasem nawet uniemożliwia sprawdzanie poprawności kodu z wykorzystaniem narzędzi typu Jslint. Dodatkowo funkcja eval zużywa czas na przekształcenie wyrażenia.

Slajd
11

 **B2E**
BUSINESS TO EDUCATION

 **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY


JavaScript

Funkcja eval - przykłady

```
eval("x=10;y=20;document.write(x*y)");
document.write("<br>" + eval("2+2"));
document.write("<br>" + eval(x+17));
```










wynikiem będą:

200
4
27

 **DAILY**
GROUP

```
eval("x=10;y=20;document.write(x*y)");
document.write("<br>" + eval("2+2"));
document.write("<br>" + eval(x+17));
wynikiem będą:
200
4
27
```


Człowiek - najlepsza inwestycja

Slajd 12	<div>   <div>JavaScript</div> </div> <h2>Funkcja isNaN</h2> <p>Sprawdza czy wartość jest liczbą. <code>isNaN(wartość);</code></p> <p>Zwraca true jeśli przekazana wartość nie jest liczbą</p> 	Funkcje <code>parseFloat</code> i <code>parseInt</code> zwracają NaN, gdy podana jako argument wartość nie jest liczbą. Wykorzystując funkcję sprawdzającą <code>isNaN</code> możemy w kodzie programu sprawdzić czy konwersja się udała i w zależności od wyniku odpowiednio zareagować.
Slajd 13	<div>   <div>JavaScript</div> </div> <h2>Funkcja isNaN - przykłady</h2> <pre>document.write(isNaN(5-2)+ "
"); document.write(isNaN(0)+ "
"); document.write(isNaN("Hello")+ "
"); document.write(isNaN("2005/12/12")+ "
");</pre> <p>wynikiem będzie:</p> <pre>false false true true</pre> 	
Slajd 14	<div>   <div>JavaScript</div> </div> <h2>Funkcje parseInt i parseFloat</h2> <p>Funkcje konwertujące łańcuch znaków na liczby całkowite i zmiennoprzecinkowe.</p> <pre>parseFloat(str); parseInt(str [, base]);</pre> 	Funkcja <code>parseFloat</code> próbuje zwrócić liczbę zmiennoprzecinkową z podanego łańcucha znaków. Jeśli natrafi na znak inny niż + lub -, cyfrę(0-9) lub . zwraca wartość uzyskaną do danego momentu i ignoruje resztę ciągu. Jeśli pierwszy znak w łańcuchu nie może być skonwertowany do liczby funkcja zwraca NaN. Funkcja <code>parseInt</code> konwertuje pierwszy


Człowiek - najlepsza inwestycja

argument (str) na liczbę całkowitą zgodną z wybraną bazą. Np. baza 10 zwróci liczbę jako dziesiętną 8 jako ósemkową, 16 - szesnastkową itd. Jeśli podczas przetwarzania funkcja trafi na znak nie znajdujący się w podanej bazie przerwie wykonywanie i zwróci wartość uzyskaną do tego momentu. Jeśli pierwszy znak w ciągu nie może być zmieniony na liczbę funkcja zwraca NaN

Slajd
15



B2E
 BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

parseFloat - przykłady


```


document.write(parseFloat("10") + "<br>");
document.write(parseFloat("10.33") + "<br>");
document.write(parseFloat("34 45 66") + "<br>");
document.write(parseFloat(" 60 ") + "<br>");
document.write(parseFloat("40 lat") + "<br>");
document.write(parseFloat("On ma 40 lat") + "<br>");
            
```



Człowiek - najlepsza inwestycja

Slajd
 16


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


parseFloat - przykłady cd.


wynikiem będą:

- 10
- 10.33
- 34
- 60
- 40 (!!!!)
- NaN


DAILY
 GROUP

 Slajd
 17



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

parseInt - przykłady

```
document.write(parseInt("10") + "<br>");
document.write(parseInt("10.33") + "<br>");
document.write(parseInt("34 45 66") + "<br>");
document.write(parseInt("40 lat") + "<br>");
document.write(parseInt("On ma 40 lat") + "<br>");
document.write(parseInt("10",10)+ "<br>");
document.write(parseInt("010")+ "<br>");
document.write(parseInt("10",8)+ "<br>");
document.write(parseInt("0x10")+ "<br>");
document.write(parseInt("10",16)+ "<br>");
```


DAILY
 GROUP

Tylko pierwsza liczba w łańcuchu jest zwracana,
 spacje na początku i końcu wyrażenia są dopuszczalne
 Starsze przeglądarki ustawiały domyślną bazę na ósemkową jeśli łańcuch będzie
 rozpoczynał się od 0.
 W wersji ECMAScript 5 domyślną bazą są liczby dziesiętne

Człowiek - najlepsza inwestycja

Slajd
18


B2E
BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript


parseInt - przykłady cd.


wyniki:

- 10
- 10
- 34
- 40 (!!!)
- NaN
- 10
- 8 (!!!)
- 8
- 16
- 16


DAILY
GROUP

Slajd
19


B2E
BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Funkcja Number i String

Funkcje pozwalające na konwersję obiektu do liczby lub łańcucha znaków

```
var obiekt;
obiekt = Number(obiekt);
obiekt = String(obiekt);
```


DAILY
GROUP

Funkcja Number wykorzystuje metodę valueOf() obiektu a funkcja String metodę toString().

Człowiek - najlepsza inwestycja




KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI


UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
 20


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Ćwiczenia


Ćwiczenie 1

Napisz funkcję, której 2 pierwsze argumenty będą operandami, a trzeci argument będzie oznaczał rodzaj działania. W zależności od rodzaju działania funkcja powinna zwracać sumę, różnicę, iloraz lub iloczyn dwóch pierwszych argumentów


DAILY
 GROUP

 Slajd
 21


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```
function math(a,b, op) {
  switch (op) {
    case "+": return a+b; break;
    case "-": return a-b; break;
    case "*": return a*b; break;
    case "/": return a/b; break;
    default: return "Nieznana operacja";break; }
}

var a = 5, b = 10;
document.writeln(math(a,b, "+"));
document.writeln(math(a,b, "-"));
document.writeln(math(a,b, "*"));
document.writeln(math(a,b, "/"));
document.writeln(math(a,b, "^"));
```



DAILY
 GROUP

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
22



B2E
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript


Ćwiczenie

Napisz funkcję, która przyjmuje 2 parametry liczbowe i zwraca większy z nich.




DAILY
GROUP

Slajd
23



B2E
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```
function wiekszy(a,b) {
  if (a > b) return a;
  else return b;
}

var a = 10;
var b = 20;
document.writeln("Wiekszy z pary " + a + " i " + b + " to "+wiekszy(a,b));
```



DAILY
GROUP

Człowiek - najlepsza inwestycja




KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI


UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
 24



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript


Ćwiczenia

Napisz funkcję przyjmującą dwa argumenty i zwracającą ich różnicę, gdy pierwszy jest nieujemny, lub ich sumę, gdy pierwszy jest ujemny. Obliczenie sumy i różnicy powinno zostać wykonane przez pomocnicze funkcje wewnętrzne.


DAILY
 GROUP

 Slajd
 25



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie


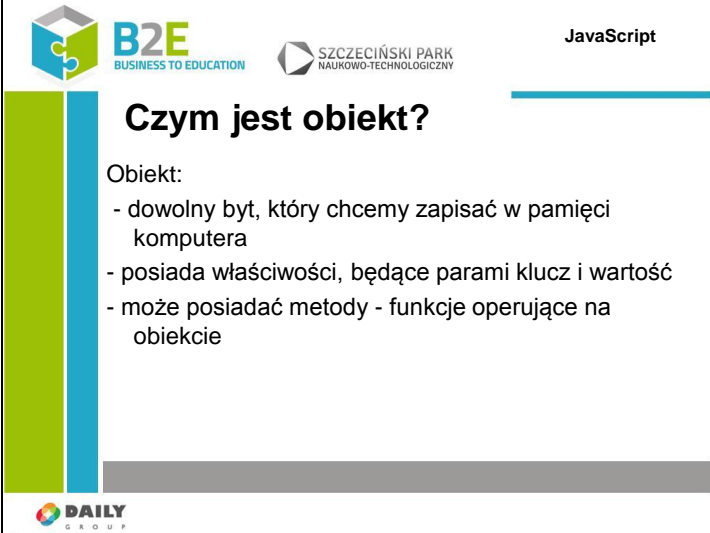
```
function wypisz(a,b) {
    function suma(a1,b1) {
        return a1 + b1;
    };
    function roznica(a1,b1) {
        return a1 - b1;
    };
    if (a>0)
        return suma(a,b);
    else return roznica(a,b);
}
document.writeln(wypisz(10, 2));
document.writeln(wypisz(-2, -4));
```


DAILY
 GROUP







Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd 26		
Slajd 27		<p>Obiekty są kolekcjami właściwości, gdzie każda właściwość ma nazwę oraz wartość. Nazwa może być dowolnym łańcuchem, również pustym. Wartość właściwości może być dowolną, dozwoloną wartością JavaScriptu, z wyjątkiem undefined. Obiekty JavaScriptu są bezklasowe. Nie ma żadnych ograniczeń co do nazw nowych właściwości lub ich wartości. Obiekty są wygodne do przechowywania i organizowania danych. Obiekty mogą zawierać inne obiekty.</p>

Człowiek - najlepsza inwestycja

Slajd 28	<div>   <div>JavaScript</div> <h2>Obiekty w JavaScript</h2> <ul style="list-style-type: none"> - zmienne - funkcje - tablice - obiekty zdefiniowane przez użytkownika  </div>	<p>W języku JavaScript proste typy to liczby, łańcuchy, typy logiczne oraz typy specjalne (null i undefined). Wszystkie pozostałe elementy, wliczając funkcje, zmienne, tablice to obiekty</p>
Slajd 29	<div>   <div>JavaScript</div> <h2>Tworzenie prostych obiektów</h2> <ul style="list-style-type: none"> - JSON - JavaScript Object Notation - obiekt należy umieścić między nawiasami klamrowymi <pre>{ definicja obiektu }</pre>  </div>	<p>Do tworzenia prostych obiektów można wykorzystać notację JSON (JavaScript Object Notation) - notacja obiektów JavaScript. Definicję obiektu należy umieścić w nawiasach klamrowych. Definicja składa się z właściwości i przypisanych im wartości. Nazwa właściwości powinna być ciągiem znaków w cudzysłowie, natomiast wartość może być łańcuchem znaków, liczbą, obiektem, tablicą itd. Kolejne właściwości oddzielamy od siebie przecinkiem.</p>










Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY












Slajd 30	<div>   <div>JavaScript</div> </div> <h2>JSON - przykład</h2> <pre> { "producent": "Fiat", "model": "Punto", "rocznik": 2010 } </pre> 	Powyższa definicja opisuje przykładowy obiekt samochód. Zawiera 3 właściwości: producent, model i rocznik.
Slajd 31	<div>   <div>JavaScript</div> </div> <h2>Obiekty JSON w skryptach</h2> <pre> var obiekt = { //definicja } np. var samochod = { "producent": "Fiat", "model": "Punto" } </pre> 	Obiekt JSON można przypisać do zmiennej albo wykorzystać funkcję eval.
Slajd 32	<div>   <div>JavaScript</div> </div> <h2>Obiekty JSON w skryptach</h2> <pre> var zmienna = nazwaObiektu.nazwaWłaściwości np. var model_samochodu = samochod.model; </pre> 	Po utworzeniu obiektu możemy uzyskać dostęp do jego właściwości. W tym celu korzystamy z operatora . (znak kropki)







Człowiek - najlepsza inwestycja

Slajd 33	<div>   <div>JavaScript</div> </div> <h3>Bezpośrednie przypisywanie wartości</h3> <p>Możemy przypisać wartości do obiektu już po jego utworzeniu, np.</p> <pre>samochod.kolor = "Biały";</pre> <pre>samochod["cena"] = 50000;</pre> <div>  </div>	<p>Jeśli przy operacji przypisania odwołamy się do nieistniejącej właściwości zostanie ona utworzona. Dostęp do właściwości obiektu możliwy jest za pomocą <code>.</code> lub nawiasu klamrowego. W nawiasie podajemy nazwę właściwości, której wartość chcemy zmodyfikować.</p>
Slajd 34	<div>   <div>JavaScript</div> </div> <h3>Odczyt i zapis za pomocą pętli</h3> <pre>for (var nazwa in nazwaObiektu) { instrukcje }</pre> <div>  </div>	<p>Specjalna wersja pętli <code>for</code> działająca na właściwościach obiektu. Za każdą iteracją pod zmienną nazwa zostanie podstawiona kolejna właściwość obiektu <code>nazwaObiektu</code>. Pętla będzie wykonywana aż odczytane zostaną wszystkie właściwości, lub zostanie przerwana za pomocą instrukcji <code>return</code> lub <code>break</code>;</p> <p>Pętla pozwala w prosty sposób przejść po właściwościach obiektu bez konieczności posiadania informacji o ich nazwach i liczbie.</p>


Człowiek - najlepsza inwestycja


Slajd 35	<div>   <div>JavaScript</div> <hr/> <h2>Metody obiektów</h2> <p>Metoda to właściwość obiektu, której wartość jest funkcją</p> <pre> obiekt.nazwa_funkcji = function() { instrukcje } </pre>  </div>	Metody wykonują zwykłe operacje na danych zawartych w danym obiekcie, więc muszą mieć dostęp do jego właściwości. Służy do tego słowo kluczowe <code>this</code> . Odróżnia się w ten sposób właściwość obiektu od innych zmiennych.
Slajd 36	<div>   <div>JavaScript</div> <hr/> <h2>Metody cd.</h2> <pre> var obiekt = { //definicje właściwości nazwa_funkcji : function () { instrukcje } } </pre>  </div>	Definicje metody można umieścić w notacji JSON, tak samo jak każdą inną właściwość obiektu. W ten sposób możemy utworzyć obiekt zawierający zarówno właściwości jak i funkcje. Metody obiektu mogą przyjmować argumenty, definiujemy je tak samo jak argumenty w funkcji./
Slajd 37	<div>   <div>JavaScript</div> <hr/> <h2>Metody - przykład</h2> <pre> var samochod = { "producent": "Fiat", "model": "Punto" "wyswietl": function() { alert(this.producent + " " + this.model); } } </pre>  </div>	Tworzymy obiekt samochód zawierający właściwości producent i model oraz metodę wyświetlającą komunikat z informacją o wartościach właściwości obiektu.

Człowiek - najlepsza inwestycja

Slajd 38	  <div>JavaScript</div> <hr/> <h1>JavaScript</h1> <p>Konstruktory i prototypy</p> 	
Slajd 39	  <div>JavaScript</div> <hr/> <h2>Konstruktory</h2> <pre>function Samochod(producent, model, rocznik) { this.producent = producent; this.model = model; this.rocznik = rocznik; } var samochod = new Samochod("Fiat", "Punto", 2010);</pre> 	<p>Tworzenie obiektów za pomocą JSON nie jest jedyną dostępną możliwością. Możemy również utworzyć obiekt za pomocą operatora new, której jako parametr podamy specjalną funkcję zwaną konstruktorem. Operator new tworzy nowy pusty obiekt a następnie przekazuje go funkcji będącej jego argumentem. Zadaniem tej funkcji jest zainicjalizowanie obiektu domyślnymi wartościami.</p>

Człowiek - najlepsza inwestycja



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Prototypy


- specjalna właściwość konstruktora
- pozwala na zdefiniowanie elementów wspólnych dla wszystkich obiektów tworzonych za pomocą tego konstruktora
- dostęp
`nazwa_funkcji_konstruktora.prototype.nazwa_metody`



DAILY
 GROUP

Na jednym z wcześniejszych slajdów utworzyliśmy prosty obiekt zawierający 2 właściwości i metodę. Jednak definiowanie metod przy każdorazowym tworzeniu obiektu jest niewygodne i nieefektywne. Poznaliśmy też konstruktory obiektu, więc moglibyśmy umieścić definicję metod w konstruktorze, aby nie być zmuszonym do ich ciągłego wpisywania. Takie rozwiązanie mimo, że wygląda dobrze nadal jest nieefektywne. Każda instancja obiektu będzie zawierała własne właściwości oraz definicje metod. Właściwości zwykle są różne dla każdego obiektu danego typu, ale definicje metod są takie same, więc mogłyby być utworzone tylko raz.

W języku JavaScript w tym celu wykorzystuje się tzw. prototyp obiektu. Użycie operatora `new` powoduje nie tylko stworzenie pustego obiektu i przekazanie go funkcji będącej konstruktorem, ale także przygotowanie prototypu obiektu. Wartością prototypu jest wartość właściwości funkcji o nazwie `prototype`. Kiedy do prototypu dodamy nową funkcję lub właściwość będzie ona dostępna dla wszystkich obiektów tworzonych za pomocą tego konstruktora.

Slajd
 41


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Prototypy cd.

```

function Samochod(producent, model, rocznik) {
    this.producent = producent;
    this.model = model;
    this.rocznik = rocznik;
}
function Samochod.prototype.wyswietl = function() {
    alert(this.producent + " " + this.model);
}
    
```



DAILY
 GROUP


Na jednym z wcześniejszych slajdów utworzyliśmy prosty obiekt zawierający 2 właściwości i metodę. Jednak definiowanie metod przy każdorazowym tworzeniu obiektu jest niewygodne i nieefektywne. Poznaliśmy też konstruktory obiektu, więc moglibyśmy umieścić definicję metod w konstruktorze, aby nie być zmuszonym do ich ciągłego wpisywania. Takie rozwiązanie mimo, że wygląda dobrze nadal jest nieefektywne. Każda instancja obiektu będzie zawierała własne właściwości oraz definicje metod. Właściwości zwykle są różne dla każdego obiektu danego typu, ale definicje metod są takie same, więc mogłyby być utworzone tylko raz.

W języku JavaScript w tym celu wykorzystuje się tzw. prototyp obiektu. Użycie operatora `new` powoduje nie tylko stworzenie pustego obiektu i przekazanie go funkcji będącej konstruktorem, ale także przygotowanie prototypu obiektu. Wartością prototypu jest wartość właściwości funkcji o nazwie `prototype`. Kiedy do prototypu dodamy nową funkcję lub właściwość będzie ona dostępna dla wszystkich obiektów tworzonych za pomocą tego konstruktora.

Człowiek - najlepsza inwestycja

Slajd
 42


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


DAILY
 GROUP


Wszystkie właściwości obiektu są publiczne, tzn. można je odczytać, modyfikować i usuwać w dowolnym miejscu skryptu.


Parametry i zmienne utworzone w konstruktorze są prywatne. Są one dołączone do obiektu, ale nie są dostępne na zewnątrz funkcji, nie są również dostępne dla publicznych metod obiektu. Aby z nich skorzystać należy wykorzystać prywatne metody – funkcje zdefiniowane w konstruktorze obiektu.

Zmienna `that`, jest wykorzystywana w celu udostępnienia publicznych właściwości obiektu dla prywatnych metod – `this` w prywatnej metodzie będzie oznaczało funkcję konstruktora i zmienne w nim zdefiniowane.

Człowiek - najlepsza inwestycja

Slajd
 43


B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript




DAILY
 G R O U P

Metoda uprzywilejowana ma dostęp do prywatnych zmiennych i metod i jest dostępna dla publicznych metod i innych funkcji. Metody te są dodawane w konstruktorze i podłączane do obiektu this.

Wywołanie funkcji pobierz zwróci zawartość obiektu pierwsze 3 razy kiedy zostanie wywołana. Po tym zawsze będzie zwracać null. Metoda wywołuje prywatną metodę dec, która działa na prywatnej zmiennej secret. Metoda pobierz jest dostępna dla innych publicznych metod i funkcji, ale nie pozwala na bezpośredni dostęp do prywatnych zmiennych.

Człowiek - najlepsza inwestycja

Slajd
44



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Właściwości prywatne i publiczne

Publiczne:


```
function Konstruktor(...) {
    this.nazwaWlasciwosci = wartosc;
}
Konstruktor.prototype.nazwaWlasciwosci = wartosc;
```

Prywatne:



```
function Konstruktor(...) {
    var that = this; var nazwaWlasciwosci = wartosc;
    function nazwaFunkcji(...) {.....}
}
```

Uprzywilejowane

```
function Konstruktor(...) {
    this.nazwaFunkcji = function (...) { ..... }
}
```


DAILY
GROUP


Slajd
45



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenie 1


Napisz kod konstruktora obiektu, będącego reprezentacją koła na płaszczyźnie. W konstruktorze umieść metody zwracające pole i obwód koła.



DAILY
GROUP

Napisz kod konstruktora obiektu, będącego reprezentacją koła na płaszczyźnie. W konstruktorze umieść metody zwracające pole i obwód koła.

Człowiek - najlepsza inwestycja

Slajd
 46



B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript


Ćwiczenie 1 - rozwiązanie

```
function Kolo(x, y, r) {
    this.x = x;
    this.y = y;
    this.r = r;
    this.pole = function() {
        return 3,14 * r * r;
    }
    this.obwod = function() {
        return 2 * 3,14 * r;
    }
}
```


DAILY
 GROUP

 Slajd
 47



B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenie 2

Napisz kod konstruktora obiektu reprezentującego prostokąt. W prototypie obiektu umieść metody zwracające pole i obwód prostokąta.


DAILY
 GROUP

Napisz kod konstruktora obiektu reprezentującego prostokąt. W prototypie obiektu umieść metody zwracające pole i obwód prostokąta.

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
 48



JavaScript

Ćwiczenie 2 - rozwiązanie

```
function Prostokat (a, b) {
    this.a = a;
    this.b = b;
}
Prostokat.prototype.pole = function() {
    return a * b;
}
Prostokat.prototype.obwod = function() {
    return 2 * a + 2 * b;
}
```


Człowiek - najlepsza inwestycja


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



11.5.4 Ćwiczenia

Ćwiczenia do lekcji zostały zaprezentowane na slajdach w poprzednim punkcie.

Wymagają one od uczestnika umiejętności definiowania obiektów w JavaScript, a także ich konstruktorów, prototypów i funkcji, które mogą być wykorzystane jako metody obiektu.

11.5.5 Opis założonych osiągnięć ucznia

Po ukończeniu tej lekcji uczestnik wie czym są obiekty w języku JavaScript, potrafi je tworzyć. Wie także jak grupować instrukcje w funkcje i w jaki sposób można je wykorzystywać.

11.6 Lekcja 6 – Model dokumentu – Document Object Model

11.6.1 Cel lekcji

Celem lekcji jest wprowadzenie definicji modelu dokumentu, jako reprezentacji strony internetowej, która może być przetwarzana przez skrypty i programy.


11.6.2 Sposoby osiągnięcia celów kształcenia

Czynności nauczyciela	Czynności ucznia
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Prowadzi z uczniami dyskusję na temat	Prezentuje rozwiązania ćwiczeń nauczycielowi i

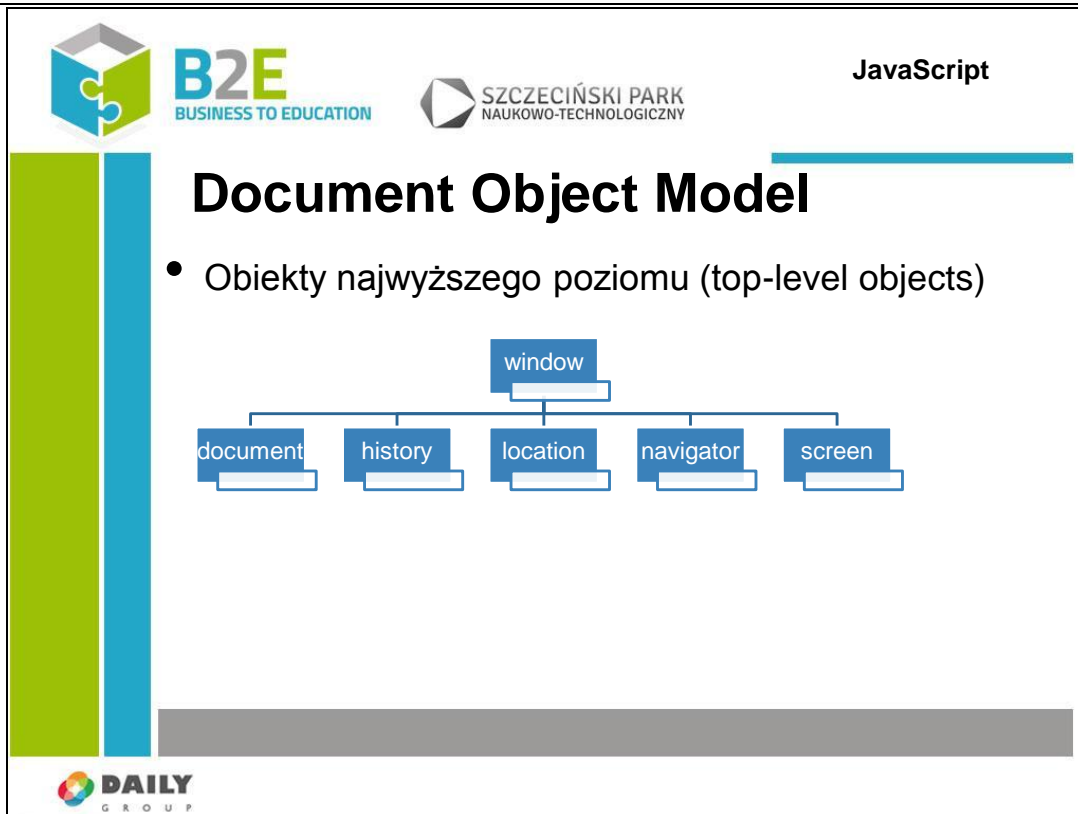
Człowiek - najlepsza inwestycja

wybranych przez nich rozwiązań ćwiczeń	pozostałym uczniom, tłumaczy wybrane przez siebie rozwiązanie
Podsumowuje lekcję, ocenia pracę uczniów na zajęciach.	Dokonuje samooceny swojej pracy, omawia z nauczycielem rozwiązania ćwiczeń

11.6.3 Treść – slajdy z opisem

Slajd 1		<p>Każda strona www składa się z wielu elementów (linków, paragrafów, obrazów itp.) Każdy z tych obiektów może mieć swoje właściwości (jak rozmiar, kolor) i można wykonywać na nich różne operacje. Dodatkowo dostępne są jeszcze obiekty opisujące okno przeglądarki i dokumentu załadowanego w bieżącym oknie. Do wygodnego zarządzania wszystkimi obiektami stworzono model dokumentu – Dokument Object Model, który określa hierarchię obiektów na stronie, ich właściwości oraz sposób zachowania.</p>
------------	--	--

Człowiek - najlepsza inwestycja




Obsługą standardu zajmuje się organizacja W3C (World Wide Web Consortium – <http://www.w3.org>), ale poszczególne przeglądarki w różnym stopniu respektują zalecenia standardu.


Document Object Model zmienia każdy element witryny www w hierarchie obiektów, z których każdy posiada swoje własne właściwości. Właściwości opisują wszystkie atrybuty obiektu takie jak HTML, który zawiera, jego wysokość, szerokość itd. Najbardziej zewnętrznym elementem hierarchii (jej korzeniem) jest obiekt window, który jest bieżącym oknem przeglądarki, zakładką, ramką pływającą (iframe) lub oknem typu popup.

Poniżej znajdują się następujące obiekty:

- dokument, który jest reprezentacją dokumentu (X)HTML,
- obiekt history zawiera historię odwiedzin stron podczas danej sesji przeglądarki,
- obiekt location reprezentuje adres URL aktualnego dokumentu,
- obiekt navigator przechowuje informacje o przeglądarce, jej nazwę, wersję itp.,
- obiekt screen.

Slajd
 3



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Obiekt window


- główny i domyślny obiekt w DOM
- reprezentuje okno lub zakładkę w przeglądarce
- zmienne globalne w JavaScript są w praktyce właściwościami tego obiektu


DAILY
 GROUP

Obiekt window reprezentuje okno przeglądarki (lub jeśli korzystamy z przeglądarek typu Firefox – jego zakładkę). Jest obiektem domyślnym tzn. że jego metody i właściwości możemy wywoływać bezpośrednio – bez konieczności podawania jego nazwy.

 Slajd
 4



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe metody obiektu window

- alert(treść) – wyświetla komunikat zawarty w argumencie treść
- confirm(treść) – wyświetla komunikat zawarty w argumencie treść oraz przyciski OK i Anuluj
- prompt(treść [,opis]) – wyświetla okno dialogowe z polem pozwalającym wprowadzić odpowiedź
- back() – odpowiednik przycisku wstecz w przeglądarce
- close() – zamyka okno
- open(URL, nazwa [właściwości, [,zamiana]])- wyświetla nowe okno o określonej nazwie.

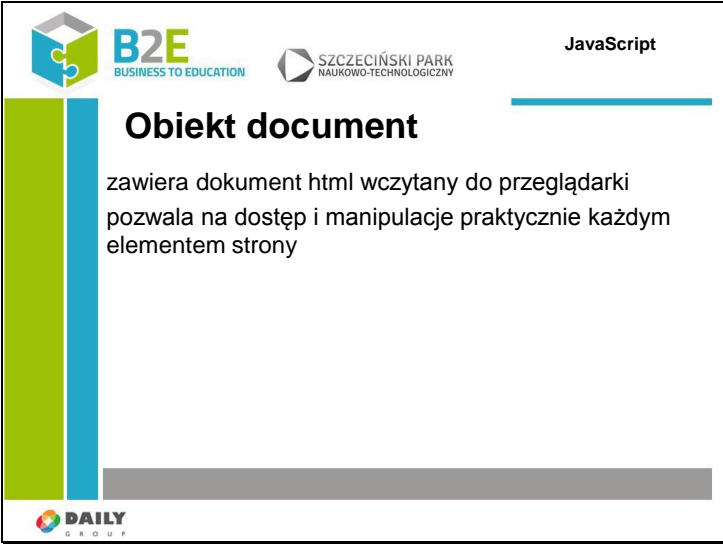
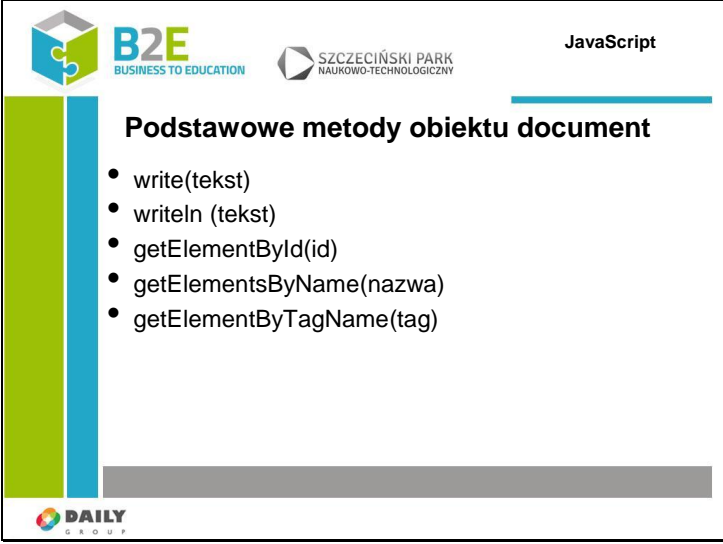

DAILY
 GROUP

Metoda alert wyświetla okienko z komunikatem i przyciskiem OK, metoda confirm pozwala nam uzyskanie potwierdzenia od użytkownika – wyświetlone zostanie okno z komunikatem i przyciski OK, i Anuluj. Wynikiem wywołania metody będzie true – jeśli wciśnięto OK lub false jeśli wciśnięto Anuluj.

Metoda prompt wyświetla komunikat zawarty w parametrze treść i pole tekstowe na wprowadzenie odpowiedzi. W polu tekstowym zostanie wyświetlony tekst wprowadzony jako opis, wynikiem działania jest treść wpisana przez użytkownika

Metoda back() cofnie nas o 1 stronę w historii przeglądarki

Człowiek - najlepsza inwestycja

	<p>Metoda close() zamyka okno przeglądarki</p> <p>Metoda open() otwiera adres Url w nowym oknie przeglądarki o podanej nazwie i właściwościach (np. rozmiarze)</p>	
Slajd 5		<p>Obiekt document reprezentuje dokument html wczytany do okna przeglądarki oraz zawiera szereg właściwości i metod pozwalających na jego modyfikację. Poprzez ten obiekt można otrzymać dostęp praktycznie do każdego elementu strony i za jego pomocą można tymi elementami manipulować.</p>
Slajd 6		<ul style="list-style-type: none"> •write(tekst) – wpisuje w dokumencie tekst zawarty w argumencie tekst •writeln (tekst) – tak samo jak write, ale na końcu dokleja znak nowej linii •getElementById(id) – zwraca odnośnik do obiektu o identyfikatorze id •getElementsByName(nazwa) – zwraca listę obiektów o atrybucie nazwa równym przekazanej nazwie •getElementByTagName(tag) – zwraca listę obiektów o danym znaczniku

Człowiek - najlepsza inwestycja



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Obiekt history

Właściwości

- current
- length
- next
- previous

Metody:

- back()
- forward()
- go(parametr)


DAILY
G R O U P

Przechowuje historię stron odwiedzanych przez użytkownika w danej sesji przeglądarki. Zawiera metody pozwalające na przemieszczanie się między odwiedzanymi stronami


history.current – zawiera adres URL bieżącej strony,
history.length – liczba stron przechowywanych w historii
właściwości next i previous zawierają adresy stron dostępnych pod przyciskami wstecz i dalej


Podstawowe metody obiektu to:
back() i forward() – wczytują odpowiednio poprzednią i następną stronę w historii

go- wczytuje stronę podaną jako parametr. Jeśli parametr jest liczbą to jest traktowany jako pozycja względem aktualnej strony w historii przeglądania. Jeśli parametr jest ciągiem znaków to traktowany jest jako adres URL

Człowiek - najlepsza inwestycja

Slajd
8


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript

Obiekt location

Zawiera adres URL aktualnego dokumentu

Metody:

- assign(url)
- reload(force)
- replace(url)



DAILY
 GROUP


Zawiera adres URL aktualnego dokumentu oraz metody pozwalające na manipulację tym adresem. We właściwościach obiektu znajdują się informacje składowe adresu (host, nazwa hosta, port). Podstawowe metody:

- assign(url) – powoduje wczytanie dokumentu o adresie wskazywanym w argumencie URL
- reload(force) – wymusza ponowne wczytanie strony. Jeśli parametr force jest ustawiony na true to strona jest ponownie wczytywana z serwera, w przeciwnym wypadku może zostać wczytana z pamięci cache
- replace(url) – powoduje wczytanie dokumentu o adresie wskazywanym w argumencie URL, bieżąca strona nie zostanie zapamiętana w historii

Człowiek - najlepsza inwestycja

Slajd
9


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Obiekt navigator


Zawiera informacje o przeglądarce i systemie operacyjnym na którym działa

Zawiera właściwość userAgent, którą można wykorzystać do określenia typu przeglądarki

```
document.write(navigator.userAgent)
```


Przykładowe skrypty określające typ przeglądarki to:


- whichbrowser (www.whichbrowser.net)
- UAParser.js (<http://faisalman.github.io/ua-parser-js>)


DAILY
 GROUP

Za pomocą właściwości userAgent obiektu navigator możemy określić typ przeglądarki i w zależności od wyniku sterować naszym skryptem. Niestety właściwość ta zawiera oprócz nazwy przeglądarki także jej wersję oraz system na którym jest uruchomiona. Utrzymywanie takiego skryptu wymaga czasu – dlatego w internecie dostępne są gotowe skrypty, które możemy wykorzystać. Posiadają one zwykle wsparcie społeczności programistów i jeśli pojawia się jakiś błąd jest on szybko usuwany.

Slajd
10



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Dostęp do elementów strony


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Tester JavaScript</title>
<script type="text/javascript">
</script>
<body>
  <div id="content">
    Witaj!
  </div>
</body>
</html>
```



DAILY
 GROUP

Strona www jest wczytywana w przeglądarce jako struktura hierarchiczna (drzewo), której elementy są zbudowane ze znaczników html. Wykorzystując JavaScript możemy wyszukiwać i modyfikować elementy, usuwać i tworzyć nowe.

Człowiek - najlepsza inwestycja

Slajd
 11


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Dostęp do elementów strony







metoda getElementById()

```
var content = document.getElementById("content");
var tresc = content.innerHTML;
content.innerHTML = tresc;
```

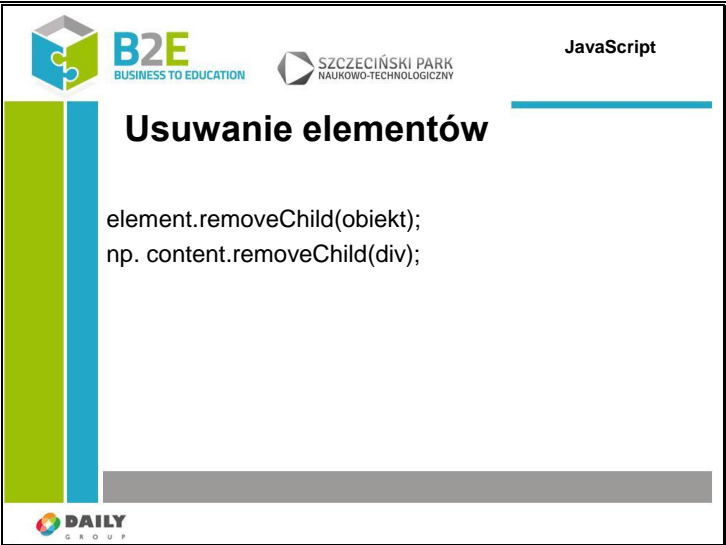

DAILY
 GROUP

Strona www jest wczytywana w przeglądarce jako struktura hierarchiczna (drzewo), której elementy są zbudowane ze znaczników html. Wykorzystując JavaScript możemy wyszukiwać i modyfikować elementy, usuwać i tworzyć nowe. Do znalezienia danego obiektu najlepiej wykorzystać metodę getElementById obiektu document. Zwraca ona referencje do obiektu w drzewie dokumentu o podanym id. Taką referencje najlepiej przypisać do zmiennej i traktować jako obiekt. Do modyfikacji elementu można wykorzystać właściwość innerHtml. Większość elementów drzewa DOM posiada taką właściwość. innerHTML to zawartość obiektu w postaci HTML. Możemy odczytać zawartość wyszukanego przez nas elementu (w naszym przypadku będzie to napis „Witaj!”) lub zmodyfikować.

Człowiek - najlepsza inwestycja


Slajd 12	<div>   <div>JavaScript</div> </div> <h3>Bezpośrednia manipulacja węzłami dokumentu</h3> <p>każdy element nie będący tekstem zawiera kolekcję węzłów potomnych (childNodes)</p> <pre>var zmienna = content.childNodes[0] zmienna.nodeValue = "Nowa treść";</pre> <div>  </div>	<p>W innerHTML możemy umieścić treść będącą kodem HTML, po przypisaniu tej wartości przeglądarka sama przebuduje drzewo DOM dokumentu. Czasami jednak wymagana jest większa kontrola nad węzłami dokumentu. W DOM każdy węzeł nie będący tekstem zawiera kolekcję węzłów potomnych. Np. węzeł <body> zawiera węzeł potomny <div>, który zawiera węzeł potomny z tekstem Witaj. Dostęp do elementów potomnych realizowany jest za pomocą tablicy childNodes. W zmiennej zmienna uzyskamy 1 element potomny węzła content. Dostęp do wartości tego obiektu możliwy jest przez właściwość nodeValue.</p>
Slajd 13	<div>   <div>JavaScript</div> </div> <h3>Tworzenie elementów przez skrypt</h3> <p>Węzły powiązane z elementami html</p> <pre>var div = document.createElement("div");</pre> <p>Węzły tekstowe</p> <pre>var tekst = document.createTextNode("Treść");</pre> <p>Łączenie:</p> <pre>div.appendChild(tekst); content.appendChild(div);</pre> <div>  </div>	<p>Za pomocą JavaScriptu możemy nie tylko modyfikować istniejące ale także dodawać nowe elementy do strony. Aby tego dokonać trzeba wyróżnić kilka typów węzłów: węzły powiązane z elementami html tworzymy za pomocą metody createElement obiektu document – jako argument podajemy nazwę tworzonego znacznika. Natomiast węzły zawierające tylko tekst tworzymy za pomocą innej metody obiektu document - createTextNode. Tak utworzone węzły łączymy ze sobą za pomocą</p>


Człowiek - najlepsza inwestycja

		<p>metody <code>appendChild</code></p> <p>Żeby taki połączony węzeł był widoczny musimy połączyć go z innym elementem naszej strony.</p>
<p>Slajd 14</p>		<p>Skoro możemy modyfikować i dodawać elementy to pewnie istnieje też sposób na usuwanie elementów. Służy do tego metoda <code>removeChild</code>, której jako parametr przekazujemy obiekt do usunięcia. element jest węzłem z którego usuwamy węzeł potomny. Usunięcie obiektu jest tylko odłączeniem go od elementu nadrzędnego, a nie usunięciem z pamięci. Węzeł nadal istnieje a referencje do niego otrzymaliśmy w wyniku działania metody <code>removeChild</code>.</p>

Człowiek - najlepsza inwestycja

Slajd
 15



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript


Ćwiczenia

Umieść w kodzie HTML warstwę zdefiniowaną za pomocą znacznika `<div>`. Napisz skrypt, który doda do tej warstwy elementy, jakie w postaci HTML miałyby postać: `<i>Szkolenie JavaScript</i>`. Nie używaj właściwości `innerHTML`.


DAILY
 GROUP

 Slajd
 16


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Przykładowe rozwiązanie

```

var str = "Szkolenie JavaScript";
var element = document.getElementById("content");
var bEl = document.createElement("b");
var iEl = document.createElement("i");
var iElTextNode = document.createTextNode(str);
bEl.appendChild(iEl);
iEl.appendChild(iElTextNode);
element.appendChild(bEl);
    
```


DAILY
 GROUP

Człowiek - najlepsza inwestycja

Slajd 17	<div>   <div>JavaScript</div> </div> <h2>Ćwiczenia</h2> <p>Napisz skrypt, który doda do warstwy div 5 elementów typu <p> z dowolną treścią.</p> <div>  </div>	
Slajd 18	<div>   <div>JavaScript</div> </div> <h2>Przykładowe rozwiązanie</h2> <pre>var content = document.getElementById("content"); for (i = 0; i<5; i++) { var p = document.createElement("p"); p.innerHTML = "Tekst"; content.appendChild(p); }</pre> <div>   <p>Człowiek - najlepsza inwestycja</p> <p>Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego</p>  </div>	

11.6.4 Ćwiczenia

Napisz skrypt wyszukujący na stronie element o identyfikatorze "test". Zmień kolor jego tła na zielony.

Napisz skrypt umieszczający treść pierwszego paragrafu strony w znaczniku <h1>

11.6.5 Opis założonych osiągnięć ucznia

Uczestnik po zakończeniu lekcji posiada podstawowe informacje na temat drzewa dokumentu (modelu DOM). Potrafi wyszukiwać w nim elementy i je modyfikować.

Człowiek - najlepsza inwestycja

11.7 Lekcja 7 – Zdarzenia

11.7.1 Cel lekcji

Interakcja na stronach www realizowana jest przez zdarzenia. „Coś” się na stronie wydarzyło. Celem lekcji jest wprowadzenie pojęcia zdarzenie i pokazanie w jaki sposób można reagować na zmiany na naszej witrynie.

11.7.2 Sposoby osiągnięcia celów kształcenia

Czynności nauczyciela	Czynności ucznia
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Prowadzi z uczniami dyskusję na temat wybranych przez nich rozwiązań ćwiczeń	Prezentuje rozwiązania ćwiczeń nauczycielowi i pozostałym uczniom, tłumaczy wybrane przez siebie rozwiązanie
Podsumowuje lekcję, ocenia pracę uczniów na zajęciach.	Dokonuje samooceny swojej pracy, omawia z nauczycielem rozwiązania ćwiczeń

11.7.3 Treść – slajdy z opisem

Slajd 1		<p>Zdarzeniem w JavaScript może być kliknięcie myszą, przesunięcie kursora nad element, opuszczenie strony itp. W JS możemy zdefiniować specjalne funkcje zwane funkcjami obsługi zdarzeń (z ang. event handler). W tej lekcji zobaczymy jakie typy zdarzeń występują oraz jak włączyć ich obsługę.</p>
------------	---	---

Człowiek - najlepsza inwestycja


B2E
 BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Obsługa zdarzeń (event handler)

- Metody rejestracji zdarzeń:
 - inline (model podstawowy)
`<znacznik onzdarzenie="instrukcje;">`
 - oddzielenie kodu JS od html
`var element = document.getElementById("content");`
`element.onclick = instrukcje;`
-



Reagowanie na zdarzenia zachodzące na stronie zwiększa jej dynamikę i atrakcyjność. Kiedy coś się dzieje na stronie przeglądarka generuje zdarzenie (obiekt) a następnie bada czy jest zarejestrowana funkcja obsługi tego zdarzenia. Jeśli tak to funkcja ta jest wywoływana.

Istnieje kilka metod przypisywania rejestracji zdarzeń w JavaScriptcie

- inline, zwany też podstawowym – najstarszy, obsługiwany podobnie przez większość przeglądarek.

Zdarzenie zwykle utożsamiane z nazwą funkcji obsługującej, która jest jednocześnie nazwą atrybutu znacznika html, do którego podpinamy zdarzenie. Ten sposób ma kilka wad

- po pierwsze miesza kod JavaScript z treścią strony,
- nie przekazuje automatycznej referencji do obiektu, który wywołał zdarzenie (można to obejść przekazując referencje `this` jako argument wywołania funkcji)
- oddzielenie kodu JS od html – umieszczenie wielu instrukcji JavaScript bezpośrednio w znaczniku jest możliwe, ale jeśli chcemy wykorzystać nasz skrypt ponownie to musimy skopiować jego treść i podłączyć do nowego elementu. Możemy to uprościć i przenieść nasz kod do funkcji a w znacznikach umieścić tylko wywołania naszych funkcji. Jednak gdy będziemy chcieli zmienić obsługę części zdarzeń na naszej stronie będziemy musieli przeszukiwać dokument html i wyszukiwać elementy, które wywołują naszą funkcję. Dlatego możliwe jest podłączenie obsługi zdarzenia bezpośrednio w kodzie skryptu. Możemy do tego wykorzystać poznaną wcześniej metodę `getElementById`, aby zlokalizować dany element na stronie i umieścić w nim obsługę zdarzenia. Aby usunąć obsługę zdarzenia przypisujemy mu wartość `null`.

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY




JavaScript

Obsługa zdarzeń (event handler)

Metody rejestracji zdarzeń:

- nowy model rejestracji zdarzeń (addEventListener)

```
var element = document.getElementById('przycisk');
element.addEventListener('click', przenies, false);
element.addEventListener('click', wypiszTekst, false);
element.addEventListener('click', function() {
    this.style.color = 'blue';
}, false);
```

- removeEventListener()

```
element.removeEventListener('click', przenies, false);
```



Wadą wcześniej przedstawionych sposobów jest możliwość podłączenia tylko jednej funkcji pod dane zdarzenie. Jeśli w trakcie działania przypiszemy do zdarzenia nową funkcję to nadpiszemy starszą.

Możemy to obejść używając „nowego” modelu rejestracji zdarzeń z wykorzystaniem metody `addEventListener()`;

Przyjmuje ona 3 parametry: typ zdarzenia, funkcja do wywołania parametrów włączający lub wyłączający bąbelkowe wywołania zdarzeń (z reguły jest on ustawiony na wartość `false`)

Bąbelkowe wywołanie zdarzeń:


(jeśli jest ustawione na `true`, zdarzenie jest wywoływane dla wszystkich elementów nadrzędnych – czyli jeśli klikniemy w obszarze `div`, który jest częścią innego obszaru to zdarzenie `onClick` zostanie przekazane do obu elementów!. Jeśli oba elementy zawierają funkcję obsługującą to zdarzenie to obie funkcje zostaną wywołane). Zdarzenie „wędruje” od elementu który je wywołał w górę dokumentu (aż do elementu `document`) tak jak bąbelki powietrza wędrują w górę butelki.


W elemencie przycisk przypisujemy 3 funkcje obsługujące kliknięcie: `przenies`, `wypiszTekst`, i anonimową funkcję zmieniającą kolor na niebieski. Od tego momentu kliknięcie na elemencie uruchomi wszystkie 3 funkcje.

Aby usunąć obsługę zdarzenia z elementu wykorzystujemy metodę `removeEventListener()`, której jako argument podajemy takie same parametry jak metodzie `addEventListener`.

Niestety nie możemy jednak usunąć anonimowych funkcji!

Slajd
 4


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Problemy z nowym modelem


starsze wersje przeglądarki Internet Explorer stosują swój własny model


```
var element = document.getElementById('przycisk');
element.attachEvent('onClick', funkcja);
element.detachEvent('onClick', funkcja);
```


DAILY
 GROUP

Wcześniejsze wersje Internet Explorera (≤ 8) stosują własny model. Zamiast zdarzenia `addEventListener` i `removeEventListener` są zdarzenia `attachEvent` i `detachEvent`. Jako parametry tych funkcji podajemy nazwę zdarzenia z przedrostkiem `on` np. `onClick` a jako 2 nazwę funkcji obsługującej zdarzenie.

 Slajd
 5


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Słowo kluczowe this

JavaScript udostępnia słowo kluczowe `this`, które jest referencją do obiektu, wywołującego metodę.

```
function zmienKolor(kolor) {
    this.style.color = kolor;
}
element.zmienKolor('#808080');
```




DAILY
 GROUP

JavaScript udostępnia słowo kluczowe `this`, które jest referencją do obiektu, wywołującego metodę. Przykładowo jeśli chcemy stworzyć funkcję, która zmieni kolor elementu możemy napisać ją w następujący sposób i wywołać.

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY

JavaScript

Dodatkowe opcje


- blokowanie domyślnej akcji (preventDefault())


```
element.addEventListener('click',function (e) {
                alert('Ten link nigdzie nie przeniesie.');
```

```
                e.preventDefault();
            },false)
```
- blokowanie nasłuchu innych zdarzeń (stopPropagation())


```
a.addEventListener('click',function (e) {
                alert('Kliknięto link');
```

```
                e.preventDefault();
                e.stopPropagation()
            },false);
```



Elementy na stronie mają w większości przypadków przypisane domyślne akcje. Np. link przenosi w inne miejsce, formularz jest wysyłany na serwer itp.







Dodane przez nas zdarzenia zostaną wykonane jako pierwsze, ale później zostanie wykonana domyślna czynność.

Aby temu zapobiec możemy skorzystać z metody `e.preventDefault()`, gdzie `e` to obiekt zdarzenia.

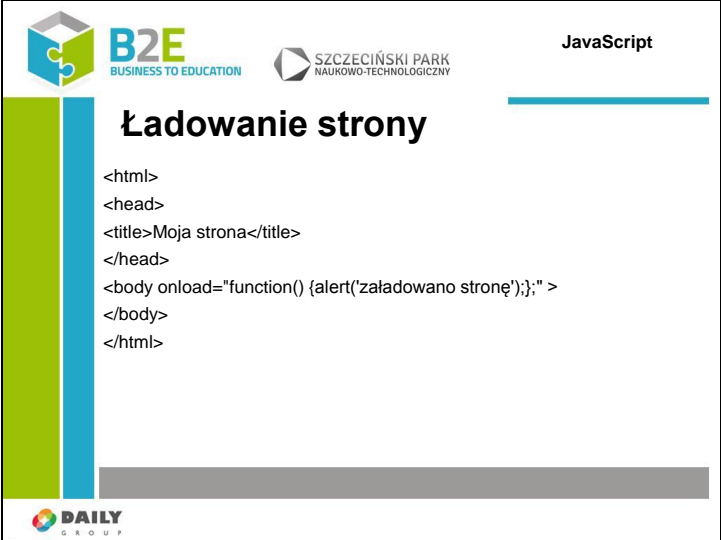
Jak wspomniałem wcześniej zdarzenie po wystąpieniu przekazywane jest w górę dokumentu – aż do znacznika `document` i jeśli po drodze napotka na metodę obsługującą dane zdarzenie to metoda ta zostanie wywołana.

Aby wstrzymać takie działanie możemy wykorzystać metodę `stopPropagation()`



Człowiek - najlepsza inwestycja

<p>Slajd 7</p>	<div>  B2E BUSINESS TO EDUCATION  SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY </div> <div> <div>JavaScript</div> <hr/> <h2>Lista zdarzeń</h2> <ul style="list-style-type: none"> - onblur - onchange - onclick - ondblclick - onfocus - onkeypress - onload </div> <div>  DAILY GROUP </div>	<p>Lista typowych zdarzeń obsługiwanych przez przeglądarkę:</p> <ul style="list-style-type: none"> - onblur: zdarzenie uruchamiane gdy element straci focus (zmienimy aktywny element na inny) - onchange: element traci focus i zmieniła się zawartość elementu - onclick: zdarzenie, które powstaje kiedy element został kliknięty - ondblclick: zdarzenie po dwukrotnym kliknięciu myszą - onfocus: element otrzymuje fokus (jest aktywny) - onkeypress: zdarzenie powstaje kiedy klawisz zostaje wciśnięty i puszczony (są jeszcze zdarzenia onkeydown i onkeyup – wywoływane po naciśnięciu i puszczeniu przycisku) - onload: przeglądarka zakończyła ładowanie strony lub ramki
<p>Slajd 8</p>	<div>  B2E BUSINESS TO EDUCATION  SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY </div> <div> <div>JavaScript</div> <hr/> <h2>Lista zdarzeń cd</h2> <ul style="list-style-type: none"> - onmousedown - onmousemove - onmouseout - onmouseover - onmouseup - onresize </div> <div>  DAILY GROUP </div>	<p>Lista typowych zdarzeń obsługiwanych przez przeglądarkę:</p> <ul style="list-style-type: none"> - onmousedown: zdarzenie powstaje gdy klawisz myszy zostanie naciśnięty nad elementem - onmousemove: kursor myszy jest przesuwany nad elementem - onmouseout: kursor myszy opuścił element - onmouseover: kursor myszy wszedł w element - onmouseup: przycisk myszy został zwolniony

Człowiek - najlepsza inwestycja

		nad elementem - onresize: zmienił się rozmiar okna
Slajd 9	 <p>Ładowanie strony</p> <pre> <html> <head> <title>Moja strona</title> </head> <body onload="function() {alert('załadowano stronę');}" > </body> </html> </pre>	<p>Komunikat „Załadowano stronę” pojawi się dopiero po wczytaniu całej strony do przeglądarki. Jest to bardzo przydatne, jeśli nasza funkcja ma działać na elementach strony. W tym przypadku mamy pewność, że wszystkie elementy są już wczytane i utworzone w przeglądarce a nasza metoda będzie mogła z nich skorzystać. Gdybyśmy umieścili wywołanie naszej metody poza zdarzeniem zostałyby wywołana w momencie wczytania przez przeglądarkę, bez oczekiwania na zakończenie wczytywania strony.</p>

Człowiek - najlepsza inwestycja


JavaScript

Obsługa kliknięć

- onclick


```
<div id="przycisk"
            style="background-color: red; width:100px; height: 100px;"
            onclick="alert('Witaj!');" >
```
- ondblclick


```
<div id="przycisk"
            style="background-color: red; width:100px; height: 100px;"
            ondblclick="alert('Witaj 2x');" >
```



Najczęściej wykorzystywane zdarzenie. Metodę obsługi zdarzenia możemy przypisać do większości elementów strony.

W tym przykładzie utworzyliśmy element div o rozmiarze 100 na 100px i podłączyliśmy do niego zdarzenie click z instrukcją wywołującą okno z komunikatem. Zdarzenie dblclick jest wywoływane po dwukrotnym kliknięciu na element. Jeśli element ma podłączoną metodę onclick to zostanie ona wywołana zamiast oczekiwanej przez nas metody ondblclick

Człowiek - najlepsza inwestycja




KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Slajd
 11


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript

Reakcja na ruch myszą

zdarzenia onmouseover i onmouseout


```


<div id="przycisk"
    style="background-color: red; width:100px; height: 100px;"
    onmouseover="alert('kursor nad elementem!');" >
</div>
<div id="przycisk"
    style="background-color: red; width:100px; height: 100px;"
    onmouseout="alert('kursor opuścił element!');" >
</div>
    
```


DAILY
 GROUP

Zdarzenia onmouseover i onmouseout służą do wykrywania czy kursor myszy znalazł się lub opuścił dany element html.

 Slajd
 12



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenia



Napisz skrypt w taki sposób, aby pierwsze kliknięcie zmieniało tekst znajdujący się na warstwie na dowolny inny, drugie przywracało jego oryginalną zawartość, trzecie ponownie go zmieniało itd.


DAILY
 GROUP

Napisz kod strony zawierającej jedną warstwę (element div). Kliknięcie tej warstwy powinno spowodować utworzenie nowej warstwy, która również będzie reagować na kliknięcie

Człowiek - najlepsza inwestycja


Slajd
 13



JavaScript

Przykładowe rozwiązanie



```

<div id="content" style="background-color:silver;width:200px;height:60px;text-align:center;"
onclick="divClick();">To jest nasza strona </div>
<script type="text/javascript">
<!--
var str = "Nowy tekst";
function divClick()
{
    var element = document.getElementById("content");
    var tempStr = element.innerHTML;
    element.innerHTML = str;
    str = tempStr;
}
// -->
</script>
    
```




Napisz kod strony zawierającej jedną warstwę (element div). Kliknięcie tej warstwy powinno spowodować utworzenie nowej warstwy, która również będzie reagować na kliknięcie.

 Slajd
 14



JavaScript

Ćwiczenia

Napisz kod strony zawierającej jedną warstwę (element div). Kliknięcie tej warstwy powinno spowodować utworzenie nowej warstwy, która również będzie reagować na kliknięcie.



Napisz kod strony zawierającej jedną warstwę (element div). Kliknięcie tej warstwy powinno spowodować utworzenie nowej warstwy, która również będzie reagować na kliknięcie

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
 15

B2E
 BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```
<div id="content" style="background-color:silver;width:200px;height:20px;text-align:center;" >div>
<script type="text/javascript">
function div1Click() {
    alert('Pierwsza warstwa została kliknięta');
    if(div2Exists) return;
    var divEl = document.createElement("div");
    divEl.addEventListener('click', function(e) {
        alert("Druga warstwa została kliknięta");}, true);
    var str = "To jest druga warstwa.";
    var divElTextNode = document.createTextNode(str);
    divEl.appendChild(divElTextNode);
    var content = document.getElementById("content");
    content.appendChild(divEl);
    div2Exists = true;
}
var div1 = document.getElementById("content");
div1.addEventListener('click', div1Click, false);
// -->
</script>
```


 Slajd
 16

B2E
 BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenia

Napisz skrypt, który zamieni wprowadzane znaki w polu formularza na duże.








Napisz skrypt, który zamieni wprowadzane znaki w polu formularza na duże.

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd 17	<div>  B2E BUSINESS TO EDUCATION </div> <div>  SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY </div> <div style="text-align: right;">JavaScript</div> <hr/> <h3 style="text-align: center;">Przykładowe rozwiązanie</h3> <pre><input type="text" name="imie" onkeyup="this.value = this.value.toUpperCase();" ></pre> <div style="text-align: center;">  Czołówek - najlepsza inwestycja </div> <div style="text-align: center;">  UNIA EUROPEJSKA FUNDUSZ SPOŁECZNY </div> <p style="text-align: center; font-size: small;">Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego</p> <div style="text-align: center;">  DAILY GROUP </div>	Można wykorzystać zdarzenie onkeyup – przez zmienną "this" mamy dostęp do wartości pola tekstowego
-------------	---	--

11.7.4 Ćwiczenia

Ćwiczenie zostało zaprezentowane na slajdzie 12. Wymaga ono od uczestnika opanowania sposobu definiowania obsługi zdarzeń i tworzenia nowych obiektów na stronie

11.7.5 Opis założonych osiągnięć ucznia

Uczestnik potrafi wymienić podstawowe zdarzenia na stronie, potrafi napisać funkcję obsługującą dowolne z tych zdarzeń.

11.8 Lekcja 8 – Obsługa błędów

11.8.1 Cel lekcji

Celem lekcji jest przedstawienie mechanizmów obsługi błędów programu w języku JavaScript. Zaprezentowane zostaną podstawowe instrukcje służące do zapewnienia poprawnego wykonania programu.

11.8.2 Sposoby osiągania celów kształcenia

Czynności nauczyciela	Czynności ucznia
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Prowadzi z uczniami dyskusję na temat wybranych przez nich rozwiązań ćwiczeń	Prezentuje rozwiązania ćwiczeń nauczycielowi i pozostałym uczniom, tłumaczy wybrane przez siebie rozwiązania
Podsumowuje lekcję, ocenia pracę uczniów na zajęciach.	Dokonyje samooceny swojej pracy, omawia z nauczycielem rozwiązania ćwiczeń


Czołówek - najlepsza inwestycja


11.8.3 Treść – slajdy z opisem

<p>Slajd 1</p>		<p>W każdym programie występują błędy. Nawet najlepsi programiści popełniają błędy. W języku JavaScript oprócz typowych błędów popełnianych przez programistę, możliwe są także błędy związane z różną interpretacją standardu ECMAScript w przeglądarkach internetowych. Np. funkcja, która zadziała w przeglądarce Google Chrome, niekoniecznie musi działać tak samo w przeglądarce Firefox. Do obsługi błędów w języku JavaScript wykorzystywane są przeniesione z języka C++ instrukcje try catch, oraz obiekty typu Error. Program może wskazać nieprawidłowe działanie przez zgłoszenie (rzucenie) wyjątku.</p>
--------------------	---	--

Człowiek - najlepsza inwestycja

Slajd
 2


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Zgłaszanie błędów

Instrukcja throw

```
throw wyrażenie;  
throw "Wystąpił błąd!";
```


Sieć CSS JS Wpisy dziennika


11:36:41.270 file:///C:/Users/Tomek/Desktop/tester3.html
 11:36:41.223 ✖ uncaught exception: Wystąpił błąd!


DAILY
 GROUP

Wyjątki (ang exceptions) to konstrukcje służące do obsługi błędów. Obsługują one sytuacje, w których program nie zadziałał tak jak założył programista. Np. użytkownik przekazał tekst w polu przeznaczonym na liczbę; nie zadziałało pobranie danych itp. Wyjątki są informacją od programu, że dana instrukcja się nie powiodła. Część wyjątków zgłaszana jest przez funkcje systemowe (np. konwersje typów, próby odczytania nieistniejącej właściwości itp.), ale nic nie stoi na przeszkodzie aby zgłaszać własne typy wyjątków w tworzonych przez nas funkcjach i metodach. Służy do tego instrukcja throw, której postać to: throw wyrażenie. Spowoduje to wypisanie na konsoli przeglądarki tekstu zawartego w wyrażeniu.

 Slajd
 3


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Zdarzenie onerror

Najprostszą metodą obsługi błędu to podłączenie funkcji do zdarzenia onError


```
window.onerror = function(komunikat, url, linia) {  
    instrukcje;  
}
```



DAILY
 GROUP

Najprostszą metodą obsługi błędu to podłączenie funkcji do zdarzenia onError. Zdarzenie onError przekazuje 3 parametry do funkcji obsługującej: komunikat o błędzie, adres url strony oraz numer linii w której wystąpił błąd.

Człowiek - najlepsza inwestycja

Slajd
 4


B2E
 BUSINESS TO EDUCATION

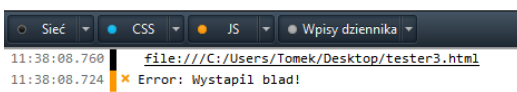

 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript

Obsługa błędów

Obiekt Error


```
throw new Error();
throw new Error("Wystąpił błąd!");
```





DAILY
 GROUP

Zamiast jednak stosować typy proste przy tworzeniu wyjątków należy skorzystać z obiektu opisującego błąd. W języku JavaScript jest to obiekt Error. Jako parametr wywołania konstruktora możemy podać treść komunikatu o błędzie. Utworzony obiekt ma 2 główne właściwości: name (to nazwa konstruktora wykorzystanego do utworzenia obiektu) oraz message – treść komunikatu podana w konstruktorze. Obiekt Error zawiera metodę toString(), która może się różnić w zależności od przeglądarki – Mozilla Firefox wyświetla domyślnie właściwości name i message oddzielone dwukropkiem.

 Slajd
 5


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Obsługa błędów







Instrukcje do przechwytywania błędów:

- try ... catch
- try ... catch ... finally
- zdarzenie onerror


DAILY
 GROUP

Wcześniejsze przykłady pokazywały sposoby zgłaszania wyjątków, które nie były w żaden sposób obsługiwane w naszym skrypcie. Wynikiem było więc wypisanie wyjątku na konsoli przeglądarki. Zamiast wypisywać komunikaty o błędach należy odpowiednio zareagować w kodzie naszego programu. W języku JavaScript do obsługi wyjątków wykorzystywane są instrukcje zapożyczone z innych języków

Człowiek - najlepsza inwestycja

		programowania takich jak C++ i Java. Są to instrukcje try catch, try catch z dodatkową klauzulą finally, oraz zdarzenie onerror.
Slajd 6	  <div>JavaScript</div> <h3>Instrukcja try ... catch</h3> <p>Instrukcja try ... catch ma następującą postać</p> <pre>try { instrukcje; } catch (wyjątek) { obsługa wyjątku }</pre> 	<p>Instrukcja try catch ma następującą postać:</p> <p>w bloku try umieszczamy kod, który chcemy wykonać i w którym może wystąpić błąd, następnie umieszczamy instrukcję catch z parametrem, w którym zostanie przekazany obiekt naszego wyjątku. Dzięki temu będziemy mieć dostęp do typu wyjątku, i komunikatu, który został w nim zdefiniowany.</p>
Slajd 7	  <div>JavaScript</div> <h3>Instrukcja finally</h3> <p>Instrukcje try catch można rozszerzyć o opcjonalną instrukcję finally.</p> <pre>try { instrukcje; } catch (wyjątek) { kod obsługi wyjątku; } finally { instrukcje; }</pre> 	<p>Instrukcje try catch możemy rozszerzyć o opcjonalną instrukcję finally. Dodanie tej klauzuli spowoduje że instrukcje wpisane w tej klauzuli zostaną wykonane zawsze, nawet gdy zostanie zgłoszony wyjątek.</p>

Slajd
 8

JavaScript

Zagnieżdżanie instrukcji try catch

```
try{
    //instrukcje, które mogą spowodować wyjątek 1
    try{
        //instrukcje, które mogą spowodować wyjątek 2
    }
    catch (wyjątek2){
        //obsługa wyjątku 2
    }
}
catch (wyjątek1){
    //obsługa 1 wyjątku
}
```



Instrukcje try catch można zagnieżdżać. Tzn, wewnątrz instrukcji try możemy umieścić kolejny blok obsługujący inny wyjątek naszego programu.

 Slajd
 9

JavaScript

Propagacja wyjątków

- W przypadku wystąpienia wyjątku jest on przekazywany do najbliższego bloku obsługi błędów. Jeśli funkcja nie zawiera obsługi błędów to wyjątek przekazywany jest do miejsca wywołania funkcji.
- Po obsłużeniu wyjątku działanie programu jest wznowiane.
- Jeśli program nie zawiera obsługi błędu, działanie całego skryptu jest zatrzymywane i informacja o błędzie wpisywana jest na konsoli przeglądarki




W momencie wystąpienia wyjątku wstrzymywane jest wykonywanie kodu skryptu, a sterowanie zostanie przekazane do najbliższego bloku obsługi wyjątku. Jeśli definicja funkcji nie zawiera obsługi błędów wyjątek jest przekazywany do miejsca wywołania funkcji. Po obsłużeniu błędu wykonywanie programu jest kontynuowane, a jeśli w całym skrypcie nie znajdzie się blok obsługi wyjątku to działanie skryptu zostanie przerwane i komunikat o błędzie będzie wypisany na konsoli.


Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
 10


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY




JavaScript


DAILY
 GROUP

Oprócz typu Error ECMAScript v3 definiuje jeszcze 6 innych: EvalError, RangeError, ReferenceError, SyntaxError, TypeError i URIError. Są dostępne od wersji 1.5 JavaScriptu. Sposób tworzenia obiektów tych typów jest taki sam jak obiektu Error.


- EvalError jest wykorzystywany w przypadku nieprawidłowego użycia funkcji eval.
- RangeError jest wykorzystywany, gdy wartość numeryczna przekracza dopuszczalny zakres.
- ReferenceError jest wykorzystywany przy próbie odczytu nieistniejących zmiennych.
- SyntaxError jest wykorzystywany po wykryciu błędu składniowego, np. przez metodę eval oraz konstruktory Function i RegExp.
- TypeError jest wykorzystywany, gdy wartość jest typu innego niż oczekiwany. Może tak być przy próbie dostępu do właściwości o wartości null bądź undefined, użycia operatora new i argumentu niebędącego konstruktorem, próbie wywołania nieistniejącej metody obiektu itp.
- URIError jest wykorzystywany przez metody kodujące bądź dekodujące adresy URI,


Człowiek - najlepsza inwestycja

		gdy wykryte zostaną nieprawidłowo sformowane dane.
Slajd 11	<div>  B2E BUSINESS TO EDUCATION  SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY </div> <div> <div>JavaScript</div> <hr/> <h2>Ignorowanie błędów</h2> <pre> try { var ajax = new XMLHttpRequest() } catch(e1) { try { ajax = new ActiveXObject("Msxml2.XMLHTTP") } catch(e2) { try { ajax = new ActiveXObject("Microsoft.XMLHTTP") } catch(e3) { ajax = false } } } </pre> <div>  DAILY GROUP </div> </div>	
	<p>W klauzuli catch nie musimy tylko informować użytkownika o napotkanym błędzie, tylko dodać np. inne instrukcje i kontynuować wykonanie programu. Jest to mechanizm przydatny przy tworzeniu wersji skryptu działającej w wielu przeglądarkach. W przykładzie pokazano utworzenie obiektu AJAX dla różnych przeglądarek. Jeśli korzystamy z nowszych przeglądarek wykonana zostanie instrukcja pierwsza, natomiast jeśli korzystamy z przeglądarki internet explorer 7 to 1 instrukcja zgłosi błąd i w obsłudze błędu spróbuje wykonać instrukcję tworzącą obiekt żądania za pomocą metody specyficznej dla internet explorera.</p>	

Człowiek - najlepsza inwestycja

Slajd
 12



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Ćwiczenie


Napisz skrypt pobierający od użytkownika informacje o dacie, w kolejnych pytaniach o rok, miesiąc i dzień (wykorzystaj instrukcję prompt, lub przygotowany przez siebie formularz). Zgłoś wyjątki, jeśli wprowadzone dane są niepoprawne (podane są wartości tekstowe, lub liczby wychodzą poza zakres).


DAILY
 GROUP

Napisz skrypt pobierający od użytkownika informacje o dacie (wykorzystaj instrukcję prompt, lub przygotowany przez siebie formularz). Zgłoś wyjątki jeśli wprowadzone dane są niepoprawne. Wyjątki powinny być zgłoszone w przypadku podanie nieprawidłowej cyfry (w przypadku miesiąca spoza zakresu 1-12 itd.), w przypadku wpisania w pole tekstu. Jeśli data zostanie wprowadzona poprawnie wypisz ją na ekranie.

 Slajd
 13



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie







```
function pobierzDane() {
    try {
        var dzien = prompt("Podaj dzień");
        if (isNaN(dzien) || Number(dzien)<0 || Number(dzien) > 31) throw new
        Error("Nieprawidłowy dzień");
        var miesiac = prompt("Podaj miesiac");
        if (isNaN(miesiac) || Number(miesiac)<0 || Number(miesiac) > 12) throw new
        Error("Nieprawidłowy miesiac");
        var rok = prompt("Podaj rok");
        if (isNaN(rok) || Number(rok)<2000 || Number(rok) > 2020) throw new
        Error("Nieprawidłowy rok");
        alert(rok+"-"+miesiac+"-"+dzien); }
    catch (error) {
        alert(error.toString()); }
}
```


DAILY
 GROUP

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


<div>Slajd 14</div>	<div><div><div><div><div>SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY</div></div></div><div>JavaScript</div></div><div><div><div>Ćwiczenie</div><div>Zmodyfikuj poprzedni przykład dodając własny wyjątek i dodaj jego wykorzystanie.</div></div><div><div></div></div></div></div>	<div>Zmodyfikuj poprzedni przykład dodając własny wyjątek i dodaj jego wykorzystanie.</div>
<div>Slajd 15</div>	<div><div><div><div><div>SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY</div></div></div><div>JavaScript</div></div><div><div><div>Przykładowe rozwiązanie</div><div><pre>function Bład(type,msg) { this.type = type; this.message = msg; this.toString = function() { return "Nasz blad to " + this.message+ "typ: "+ this.type; } } var dzien = prompt("Podaj dzień"); if (isNaN(dzien) Number(dzien)<0 Number(dzien) > 31) throw new Bład("Nasz", "Nieprawidłowy dzień");</pre></div><div><div>Człowiek - najlepsza inwestycja</div><div><div><div>KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI</div></div><div><div>UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY</div></div></div><div>Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego</div></div><div><div></div></div></div></div></div>	

11.8.4 Ćwiczenie

Napisz skrypt pobierający od użytkownika informacje o dacie (wykorzystaj instrukcję prompt, lub przygotowany przez siebie formularz). Zgłoś wyjątki jeśli wprowadzone dane są niepoprawne.

Wyjątki powinny być zgłoszone w przypadku podanie nieprawidłowej cyfry (w przypadku miesiąca spoza zakresu 1-12 itd.), w przypadku wpisania w pole tekstu.

Jeśli data zostanie wprowadzona poprawnie wypisz ją na ekranie.

Ćwiczenie obrazuje praktyczny sposób wykorzystania instrukcji poznanych w czasie lekcji.

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



11.8.5 Opis założonych osiągnięć ucznia

Uczestnik potrafi dodać do skryptu metody obsługi błędów.

11.9 Lekcja 9 – Wykorzystanie języka JavaScript


11.9.1 Cel lekcji

Po wprowadzeniu podstawowych elementów języka JavaScript czas pokazać do czego można ten język wykorzystać. W lekcji zostaną przedstawione podstawowe zastosowania języka JavaScript – takie jak walidacja przesyłanych formularzy, modyfikacja elementów css, operacje na łańcuchach znaków.

11.9.2 Sposoby osiągnięcia celów kształcenia

Czynności nauczyciela	Czynności ucznia
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Prowadzi z uczniami dyskusję na temat wybranych przez nich rozwiązań ćwiczeń	Prezentuje rozwiązania ćwiczeń nauczycielowi i pozostałym uczniom, tłumaczy wybrane przez siebie rozwiązanie
Podsumowuje lekcję, ocenia pracę uczniów na zajęciach.	Dokonyuje samooceny swojej pracy, omawia z nauczycielem rozwiązania ćwiczeń

11.9.3 Treść – slajdy z opisem

Slajd 1		<p>W lekcji omówimy najczęstsze sposoby wykorzystania języka JavaScript.</p> <ul style="list-style-type: none"> - zmiana elementów css - operacje na łańcuchach znaków - wykorzystanie do walidacji formularzy <p>Zobaczemy, jak wykorzystać zaprezentowane na wcześniejszych lekcjach elementy języka JavaScript.</p>
------------	--	---

Człowiek - najlepsza inwestycja

Slajd
 2


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

CSS – atrybut style

- `getAttribute`
 - `var obiekt = document.getElementById('przycisk');`
 - `var styl = obiekt.getAttribute("style");`
- `setAttribute`
 - `obiekt.setAttribute("style", "background-color: blue");`
- `removeAttribute`
 - `obiekt.removeAttribute("style");`


DAILY
 GROUP

Style CSS to ważny element każdej nowoczesnej witryny WWW. Za pomocą JavaScriptu możliwa jest dynamiczna zmiana właściwości arkuszy styl. Style CSS danego elementu przechowywane są w atrybucie „style” danego elementu. Dostęp do tego atrybutu możliwy jest z wykorzystaniem metod `getAttribute`, `setAttribute` i `removeAttribute`

Metoda `getAttribute(„style”)` pobiera wartość atrybutu `style` i umieszcza w zmiennej `styl`. Aby ustawić styl dla obiektu wywołujemy metodę `setAttribute` z dwoma parametrami. Pierwszym jest nazwa atrybutu (w naszym przypadku `style`) a drugim jego wartość.

Usunięcie stylu możliwe jest za pomocą metody `removeAttribute`.

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY





JavaScript

Obiekt style

każdy obiekt reprezentujący element strony zawiera właściwość style.

Jest ona odwzorowaniem atrybutu html style

```

<div style="
  background-color: blue;
  width: 100px;
  height: 100px;">
</div>
            
```

```

element.style.backgroundColor = "blue";
element.style.width = "100px";
element.style.height = "100px";
            
```




Wykorzystanie atrybutu style jest wygodne jeśli chcemy zmodyfikować całą jego zawartość. Jeśli jednak chcemy zmienić tylko jedną właściwość (np. kolor tła) pobranie całego atrybutu będzie wymagało od nas obsłużenia także innych właściwości, ustawionych dla tego elementu. Będziemy musieli wyszukać w atrybucie aktualny kolor tła, a przy zapisie zmodyfikowanej wartości pamiętać także o pozostałych ustawionych właściwościach.


Na szczęście w JavaScript każdy element ma swój obiekt style (który jest odwzorowaniem atrybutu style).

Jeśli w zmiennej obiekt przechowujemy referencje do elementu witryny do dostęp do obiektu style możliwy jest przez wypisanie obiekt.style.

Atrybuty css danego elementu są odwzorowywane do właściwości obiektu style w następujący sposób: jeśli atrybut elementu nazywał się background-color to jego reprezentacja w JavaScript będzie wyglądała następująco:

backgroundColor


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript



DAILY
 GROUP


Do tej pory zmienialiśmy tylko konkretne atrybuty stylu przypisanego do danego elementu. Stosując JavaScript możemy również zmienić wartość atrybutu class elementu html, a co za tym idzie zmienić cały styl przypisany do tego elementu. Atrybut class zamieniany jest na właściwość className obiektu odpowiadającego danemu znacznikowi.

Aby zmodyfikować właściwość class elementu należy wyszukać go za pomocą metody document.getElementById() a następnie do jego właściwości className przypisać nową wartość.

Człowiek - najlepsza inwestycja

Slajd
 5


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Właściwość className


```


<html><head><style type="text/css">
    .zielony { background-color: green; }
    .niebieski { background-color: blue; }
</style>
<script type="text/javascript">
    function zmienKlase() {
        var element = document.getElementById("przycisk");
        element.className="niebieski"; }
</script></head><body>
<div id="przycisk" class="zielony" style="width: 100px; height:
100px;"></div>
<input type="button" value="zmień" onclick="zmienKlase();"></input>
</body></html>
    
```


DAILY
 GROUP

Aby zmodyfikować właściwość class elementu, należy wyszukać go za pomocą metody `document.getElementById()`, a następnie do jego właściwości `className` przypisać nową wartość.

 Slajd
 6



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenia

Umieść na stronie przycisk. Niech po kliknięciu zmienia się jego kolor tła i kolor tekstu.
 Wykorzystaj obie metody (modyfikacje obiektu style) i zmianę klasy css


DAILY
 GROUP

Ćwiczenie

Umieść na stronie przycisk. Niech po kliknięciu zmienia się jego kolor tła i kolor tekstu.
 Wykorzystaj obie metody (modyfikacje obiektu style) i zmianę klasy css.


Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
7



JavaScript



Przykładowe rozwiązanie

```

<html><head><style type="text/css">
    .zielony { background-color: green; }
    .niebieski { background-color: blue; }
</style>
<script type="text/javascript">
    function zmienKlase() {
        var element = document.getElementById("przycisk");
        element.className="niebieski";
    }
</script></head>
<body>
<div id="przycisk" class="zielony" style="width: 100px; height: 100px;" ></div>
<input type="button" value="zmień" onclick="zmienKlase();" ></input>
</body></html>
    
```




Slajd
8



JavaScript

Ćwiczenia

Umieść na stronie przycisk. Niech najechanie na niego zmieni kolor jego tła i tekstu.



Ćwiczenie

Umieść na stronie przycisk. Niech najechanie na niego zmieni kolor jego tła i tekstu.

Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Slajd
 9

JavaScript

Przykładowe rozwiązanie

```
<html><head><style type="text/css">
    .zielony { background-color: green; }
    .niebieski { background-color: blue; }
</style><script type="text/javascript">
function zmien() {
    var element = document.getElementById("przycisk");
    element.style.backgroundColor = 'blue';
    element.style.color = 'white'; }
function wroc() {
    var element = document.getElementById("przycisk");
    element.style.backgroundColor = 'green';
    element.style.color = 'black';}
</script></head><body>
<div id="przycisk" class="zielony" style="width: 100px; height: 100px;"
onmouseover="zmien()" onmouseout="wroc()" ></div>
</body></html>
```


 Slajd
 10

JavaScript

Ćwiczenia

Napisz skrypt, który zmieni kolor tła aktualnie zaznaczonego pola w formularzu na żółty i z powrotem na biały po jego odznaczeniu



Ćwiczenie

Napisz skrypt, który zmieni kolor tła aktualnie zaznaczonego pola w formularzu na żółty i z powrotem na biały po jego odznaczeniu.



Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
11




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY


B2E
BUSINESS TO EDUCATION



JavaScript


Przykładowe rozwiązanie

```
function zaznacz(x) {
  x.style.background="yellow";
}
function odznacz(x) {
  x.style.background="white";
}
(...)
<input type="text" onfocus="zaznacz(this)"
onblur="odznacz(this)">
```


DAILY
GROUP

Slajd
12




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY


B2E
BUSINESS TO EDUCATION

JavaScript

Przetwarzanie tekstów


- długość tekstu
- przetwarzanie ciągów



DAILY
GROUP

W czasie pisania programów wielokrotnie będziemy potrzebowali wywołać jakieś operacje na ciągach znaków, sprawdzić ich długość, sprawdzić czy zawiera określony ciąg znaków, albo zbadać jaka jest dowolna litera w łańcuchu. Do przetwarzania tekstów można wykorzystać różne dostarczone w języku metody.

Człowiek - najlepsza inwestycja

Slajd
 13


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Jak sprawdzić długość tekstu?

Każdy łańcuch znaków w JavaScript jest obiektem. Obiekt string zawiera m.in. właściwość `length` – liczbę zawartych w nim znaków

```
var tekst = "Witaj świecie!";
var dlugosc = tekst.length;
alert("Długość tekstu to: "+dlugosc);
```


DAILY
 GROUP

Każdy łańcuch znaków w JavaScript jest obiektem. Obiekt string zawiera m.in. właściwość `length` – liczbę zawartych w nim znaków. Powyższy przykład po uruchomieniu pokaże komunikat: "Długość tekstu to 14".

 Slajd
 14


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript




Przetwarzanie ciągów

• <code>charAt</code>	<code>tekst.charAt(indeks)</code>
• <code>charCodeAt</code>	<code>tekst.charCodeAt(indeks)</code>
• <code>concat</code>	<code>tekst.concat(tekst2...tekstN)</code>
• <code>fromCharCode</code>	<code>String.fromCharCode(kod1..kodN)</code>
• <code>indexOf</code>	<code>tekst.indexOf(wartość [,indeks])</code>
• <code>lastIndexOf</code>	<code>tekst.lastIndexOf(wartość [, indeks])</code>
• <code>match</code>	<code>tekst.match(wyrażenieReg)</code>
• <code>replace</code>	<code>tekst.replace(wyrażenieReg, nowy)</code>








DAILY
 GROUP

`charAt(indeks)` – zwraca znak znajdujący się pod wybranym indeksem
`charCodeAt(indeks)` - zwraca kod znaku znajdującego się pod wybranym indeksem (w JS 1.2 – kod ISO Latin1 w późniejszych Unicode)
`concat` – łączy łańcuchy znakowe. Wynikiem jest łańcuch powstały z połączenia tekstu oryginalnego i wszystkich tekstów podanych jako parametry
`fromCharCode` – zwraca łańcuch znaków złożony z podanych jako parametry kodów.
`indexOf` – zwraca indeks wystąpienia łańcucha wartość w łańcuchu tekst. Jeśli podamy parametr indeks to przeszukiwanie rozpoczyna się od tego indeksu.
`lastIndexOf` – zwraca indeks ostatniego wystąpienia łańcucha wartość w łańcuchu tekst.
`match` – zwraca część ciągu tekst pasującego do

Człowiek - najlepsza inwestycja

		wyrażenia regularnego, jeśli nie znajdzie dopasowania zwraca null replace – zwraca ciąg znaków, w którym fragmenty opisane przez wyrażenie zostały zamienione na nowy tekst														
Slajd 15	<div><div>B2E BUSINESS TO EDUCATION</div><div>SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY</div><div>JavaScript</div></div> <div><h2>Przetwarzanie ciągów</h2><table><tr><td>• search</td><td>tekst.search(wyrażenieReg)</td></tr><tr><td>• slice</td><td>tekst.slice(start [,koniec])</td></tr><tr><td>• split</td><td>tekst.split([separator [,limit]])</td></tr><tr><td>• substr</td><td>tekst.substr(indeks [,liczba])</td></tr><tr><td>• substring</td><td>tekst.substring(od, do)</td></tr><tr><td>• toLowerCase</td><td>tekst.toLowerCase()</td></tr><tr><td>• toUpperCase</td><td>tekst.toUpperCase()</td></tr></table></div> <div></div>	• search	tekst.search(wyrażenieReg)	• slice	tekst.slice(start [,koniec])	• split	tekst.split([separator [,limit]])	• substr	tekst.substr(indeks [,liczba])	• substring	tekst.substring(od, do)	• toLowerCase	tekst.toLowerCase()	• toUpperCase	tekst.toUpperCase()	
• search	tekst.search(wyrażenieReg)															
• slice	tekst.slice(start [,koniec])															
• split	tekst.split([separator [,limit]])															
• substr	tekst.substr(indeks [,liczba])															
• substring	tekst.substring(od, do)															
• toLowerCase	tekst.toLowerCase()															
• toUpperCase	tekst.toUpperCase()															

Człowiek - najlepsza inwestycja

<p>Slajd 16</p>	<div>  B2E BUSINESS TO EDUCATION  SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY JavaScript </div> <h2>Przetwarzanie ciągów - przykłady</h2> <ul style="list-style-type: none"> concat <pre>var t1 = "test".concat("java", "script"); "testjavascript"</pre> indexOf <pre>var indeks = "testjavascript".indexOf("va"); 6</pre> match <pre>var tekst = "testjavascript".match("java"); java</pre> 	
<p>Slajd 17</p>	<div>  B2E BUSINESS TO EDUCATION  SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY JavaScript </div> <h2>Przetwarzanie ciągów - przykłady</h2> <ul style="list-style-type: none"> replace <pre>var tekst = "Witaj %IMIE%!"; tekst.replace("%IMIE%", "Tomek"); "Witaj Tomek!"</pre> split <pre>var tablica = "a,b,c,d".split(","); [a,b,c,d]</pre> 	<p>replace - Metoda zamienia wystąpienie zgodne z wyrażeniem regularnym podanym jako 1 argument na tekst podany jako 2 argument.</p> <p>split - dzieli tekst na fragmenty względem separatora podanego jako argument, opcjonalnie można podać maksymalną liczbę dopasowanych elementów</p>


B2E
 BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Wprowadzanie danych

- formularze

```
<form name="formularz" action="submit.php" method="POST">
```

```
.....
```

```
<input type="submit" value="wyślij" />
```

```
</form>
```

- sprawdzanie poprawności danych

```
<form name="formularz" action="submit.php"
```

```
onsubmit="return sprawdz()" method="POST">
```

```
.....
```

```
<input type="submit" value="wyślij" />
```

```
</form>
```



Formularz to element strony, który tworzymy za pomocą znacznika form. Dostęp do formularzy możliwy jest przez tablicę forms obiektu document. Formularz służy do przesłania zestawu danych od użytkownika do serwera. Warto wykorzystać JavaScript aby dokonać wstępnej analizy danych wprowadzonych do formularza. Sprawdzenie formularza powinno odbywać się również po stronie serwera, ale nie ma sensu niepotrzebnie obciążać serwera jeśli dane w formularzu są błędnie wprowadzone. Zobaczmy w jaki sposób możemy wykorzystać napisaną przed chwilą funkcję.

Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



B2E
 BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Sprawdzanie poprawności danych

```
<form name="logowanie" action="login.php" method="POST">
  <input type="text" name="login">
  <input type="password" name="password">
  <input type="button" value="Zaloguj"
onclick="sprawdz()" ">
</form>

function sprawdz() {
  var loginForm = document.forms["logowanie"];
  if (loginForm.login == "" || loginForm.password == "")
  {
    alert("Nie podano nazwy użytkownika lub
hasła");
    return; }
  loginForm.submit();
}
```



Możemy sprawdzić formularz na dwa sposoby.

Po pierwsze możemy nie umieszczać w nim przycisku typu submit tylko zwykły przycisk typu button i w funkcji obsługującej zdarzenie onclick umieścić sprawdzenie danych i wysyłanie formularza jeśli dane są poprawne.

Nasz przykładowy formularz zawiera 2 pola tekstowe do wprowadzenia nazwy użytkownika i hasła oraz przycisk "Zaloguj". Po kliknięciu przycisku sprawdzane jest czy pola login i password zostały wypełnione (są różne od znaku pustego) i jeśli tak to formularz jest wysyłany przez wywołanie metody submit(). Jeśli którekolwiek z pól jest puste, wyświetlony jest komunikat i działanie funkcji zostanie przerwane.



Slajd
 20


B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript



DAILY
 GROUP

Możemy sprawdzić formularz na dwa sposoby.

Drugim sposobem jest podłączenie obsługi zdarzenia submit do formularza. Ogólna postać jest funkcji jest bardzo podobna. W obsłudze zdarzenia submit wpisaliśmy wywołanie `return sprawdz()`. Formularz zostanie przesłany do serwera tylko w przypadku gdy funkcja `sprawdz` zwróci jako wynik `true`. Stanie się tak, jeśli pola `login` i `password` zostaną wypełnione.

 Slajd
 21


B2E
 BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


DAILY
 GROUP

Człowiek - najlepsza inwestycja

Slajd
22


JavaScript

Przykład

```



<form name="logowanie" >
<input type="text" name="pesel"
onchange="sprawdzPesel();" >
</form>

function sprawdzPesel() {
    var form1 = document.forms["logowanie"];
    if (form1.pesel.value.length != 11) {
        alert("Nieprawidłowy numer PESEL");
        form1.pesel.focus();
    }
}
    
```



Jeśli wpisujemy liczbę znaków różną od 11, pojawi się komunikat i pole pesel zostanie ustawione jako aktywne.
 Co się jednak stanie, gdy ponownie opuścimy pole formularza (bez dokonywania poprawek)?
 Q: Jak możemy temu zapobiec?
 A: wykorzystując zdarzenie onBlur


 Slajd
23

JavaScript

Ćwiczenie

Napisz funkcję javascript sprawdzającą poprawność wpisanego w formularzu adresu e-mail









Ćwiczenie:
 Napisz funkcję javascript sprawdzającą poprawność wpisanego w formularzu adresu e-mail

Jako poprawny adres email uznajemy adres:

- zawiera tylko 1 znak @
- znak @ nie jest pierwszym elementem ciągu
- po znaku @ występuje cyfra lub litera.
- w ciągu po @ występuje znak . ale nie jest na ostatniej pozycji

Do napisania algorytmu

Człowiek - najlepsza inwestycja

		można wykorzystać następujące metody (indexOf, lastIndexOf, match, search)
Slajd 24	<div>   <div>JavaScript</div> <h2>Przykładowe rozwiązanie</h2> <pre>function sprawdzEmail(email) { var malpa = email.indexOf("@"); var kropka = email.lastIndexOf("."); if (malpa < 1 kropka < malpa+2 kropka+1 >= email.length) { alert("Nieprawidłowy adres email"); return false; } }</pre>  </div>	
Slajd 25	<div>   <div>JavaScript</div> <h2>Wyrażenia regularne</h2> <ul style="list-style-type: none"> • Wzorce opisujące łańcuchy tekstów • Pozwalają na określenie czy podany ciąg znaków pasuje do wzorca. • Mogą wyszukiwać w tekście wystąpienia wzorca. <p>W JavaScript reprezentowane są przez obiekt RegExp</p>  </div>	<p>Wyrażenia regularne to wzorce opisujące łańcuchy tekstów. Wyrażenia regularne mogą określać zbiór pasujących łańcuchów albo wskazywać istotne części łańcucha znaków. Istnieją algorytmy pozwalające sprawdzić czy podany ciąg znaków pasuje do wzorca. Za pomocą wyrażeń regularnych można wyszukiwać w tekście wystąpienia wzorca. Wyrażenia regularne są reprezentowane w JavaScript za pomocą obiektów typu RegExp.</p>

Człowiek - najlepsza inwestycja




KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Slajd
 26


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Obiekt RegExp

Tworzenie obiektów RegExp

- `var zmienna = new RegExp(wzorzec, atrybuty)`
- `var zmienna = /wzorzec/atributy`

```
var wzorzec = /JavaScript/
wzorzec.test("To jest kurs JavaScript");
```


DAILY
 GROUP

Wyrażenia regularne są reprezentowane w JavaScript za pomocą obiektów typu RegExp. Obiekt może zostać utworzony na dwa sposoby

- przez jawne wykorzystanie konstruktora obiektu (`var zmienna = new RegExp(wzorzec, atrybuty)`)
- przez przypisanie wyrażenia do zmiennej

Utwórzmy prosty wzorzec – niech to będzie ciąg znaków Javascript. Aby przetestować czy w tekście znajduje się ustalony przez nas wzorzec wywołujemy metodę test obiektu RegExp. Metoda zwraca true jeśli wzorzec występuje w tekście i false w przeciwnym przypadku. Obiekt RegExp ma jeszcze kilka innych metod:

 Slajd
 27


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Metody obiektu RegExp




- `test(tekst)` `var wynik = wzorzec.test(tekst);`
- `compile(wyrażenie [, atrybuty])`
- `exec(tekst)` `var wynik = wzorzec.exec(tekst);`


DAILY
 GROUP

Metoda test bada czy w zadanym jako argument tekście znajduje się wzorzec, zdefiniowany przez obiekt typu RegExp. Wynikiem jest true lub false

Metoda compile() dokonuje wewnętrznej kompilacji wyrażenia, co pozwala na szybsze jego przetwarzanie. Wyrażenie może zawierać argumenty i – pomijanie wielkości liter, g – globalne, m – przetwarzanie ma dotyczyć ciągu

Człowiek - najlepsza inwestycja

		<p>zawierającego wiele wierszy.</p> <p>hec – metoda przeszukuje tekst w poszukiwaniu wzorca. Zwraca tablicę z wynikami dopasowań.</p>
Slajd 28	<div>    </div> <h2>Tworzenie wyrażeń</h2> <h3>Atrybuty (flagi) wyrażeń regularnych</h3> <ul style="list-style-type: none"> • g • i • m <pre>var wzorzec = new RegExp("javascript", "i"); var wzorzec = /javascript/i;</pre>	JavaScript

Wspomnieliśmy przed chwilą, że kompilowane wyrażenie może zawierać atrybuty, oraz że obie metody tworzenia wyrażeń także takie miejsce mają.

W wyrażeniach regularnych możemy podawać atrybuty (zwane też flagami) pozwalające na zmianę zachowania wyrażenia.

Dostępne są następujące parametry:

- g – globalny – oznaczający że przetwarzany ma być cały ciąg (bez tego znacznika) zwracany będzie tylko pierwszy dopasowany fragment. Parametr o bardzo dużym znaczeniu przy poszukiwaniu wszystkich wystąpień wzorca w tekście, można go pominąć jeśli interesuje nas tylko czy tekst zawiera fragment opisany wzorcem
- i – oznaczający że wielkość liter w tekście ma być ignorowana. Bez podania tego znacznika domyślną wartością jest false – treść w tekście musi mieć taką samą wielkość liter jak ta zdefiniowana we wzorcu
- m – oznaczający, że przetwarzanie ma dotyczyć tekstu wielowierszowego. W trybie tym znak początku i końca wzorca (^\$) jest wstawiany przed i po znaku nowej linii (\n).







Człowiek - najlepsza inwestycja






KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Slajd 29	<div>   <div>JavaScript</div> </div> <h2>Tworzenie wyrażeń</h2> <p>Wyrażenia budowane są ze znaków zwykłych (litery alfabetu i cyfry) i znaków specjalnych (większość znaków interpunkcyjnych) takich jak:</p> <p><code>^ \$. * + ? = ! : \ / () [] { }</code></p> <p>Aby wykorzystywać znaki specjalne w wyrażeniach należy poprzedzić je znakiem lewego ukośnika (<code>\</code>)</p>	<p>Wyrażenia budowane są ze znaków zwykłych oraz znaków specjalnych. Znaki zwykle to wszystkie litery alfabetu i cyfry. Znaki specjalne to większość znaków interpunkcyjnych takich jak: <code>^ \$. * + ? = ! : \ / () [] { }</code>. Aby z nich korzystać należy podać przed nimi znak <code>\</code> (lewy ukośnik)</p>
Slajd 30	<div>   <div>JavaScript</div> </div> <h2>Tworzenie wyrażeń</h2> <p>W wyrażeniach regularnych wprowadzono pojęcie klasy znakowej.</p> <p>Klasa znakowa to zdefiniowany zestaw znaków, aby ją zbudować korzystamy z operatora zakresu <code>[]</code>.</p> <pre>var wyrażenie = /[a-zA-Z]/</pre>	<p>W wyrażeniach regularnych wprowadzono pojęcie klasy znakowej. Klasa znakowa to zdefiniowany zestaw znaków, aby ją zbudować korzystamy z operatora zakresu <code>[]</code>. Istnieją klasy predefiniowane które zawierają definicje często wykorzystywanych klas</p>
Slajd 31	<div>   <div>JavaScript</div> </div> <h2>Klasy znakowe</h2> <ul style="list-style-type: none"> <code>[...]</code> Dowolny znak znajdujący się w nawiasie <code>[^...]</code> Dowolny znak nieznajdujący się w nawiasie <code>.</code> Dowolny znak <code>\d</code> – dowolna cyfra <code>\D</code> – dowolny znak inny niż cyfra <code>\s</code> – dowolny biały znak (tabulator, spacja) <code>\S</code> – dowolny znak niebędący białym znakiem <code>\w</code> – dowolny znak wyrazu złożonego ze znaków ASCII <code>\W</code> – dowolny znak niebędący znakiem słowa złożonego ze znaków ASCII 	<p><code>[...]</code> Dowolny znak znajdujący się w nawiasie – lista znaków podana w nawiasie jest listą znaków, które mogą wystąpić na tej pozycji w tekście</p> <p><code>[^...]</code> Dowolny znak nieznajdujący się w nawiasie – odwrotność – na pozycji w tekście nie mogą wystąpić znaki z nawiasu. Operator <code>^</code> musi być pierwszym po nawiasie</p> <p><code>.</code> Dowolny znak – na tej pozycji może być dowolny</p>

Człowiek - najlepsza inwestycja

		<p>znak oprócz znaku nowej linii</p> <p>\d – dowolna cyfra – uproszczony zapis [0-9]</p> <p>\D – dowolny znak inny niż cyfra – odwrotność powyższego [^0-9]</p> <p>\s – dowolny biały znak (tabulator, spacja)</p> <p>\S – dowolny znak niebędący białym znakiem</p> <p>\w – dowolny znak wyrazu złożonego ze znaków ASCII – dowolna cyfra, litera lub znak podkreślenia [0-9a-zA-Z_]</p> <p>\W – dowolny znak niebędący znakiem słowa złożonego ze znaków ASCII [^0-9a-zA-Z_]</p>
Slajd 32	  <div style="text-align: right;">JavaScript</div> <hr/> <h2 style="text-align: center;">Powtórzenie wzorca</h2> <p>Często zdarza się sytuacja, że fragment wzorca powinien się powtórzyć. Istnieją odpowiednie elementy, które umieszczone w wyrażeniu oznaczają ile razy może pojawić się dany element znajdujący się przed nim.</p> <ul style="list-style-type: none"> • ? • * • + <p style="text-align: center;">ab?c+d*</p> 	<p>Często zdarza się sytuacja, że fragment wzorca powinien się powtórzyć. Istnieją odpowiednie elementy, które umieszczone w wyrażeniu oznaczają ile razy może pojawić się dany element, znajdujący się przed nim.</p> <p>? – oznacza że element może pojawić się 0 lub jeden raz</p> <p>* - oznacza że element może pojawić się dowolną ilość razy (zero lub więcej)</p> <p>+ - oznacza, że element może pojawić się jeden lub więcej razy</p> <p>W powyższym wyrażeniu dopasowane zostaną fragmenty tekstu, które: zaczynają się od znaku "a", po nich może ale nie musi wystąpić litera "b", następnie będzie co najmniej jedna litera c po których może wystąpić dowolna liczba liter d.</p>







Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI


UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY




Slajd 33	<div>   JavaScript </div> <h2>Powtórzenie wzorca</h2> <p>Dodatkowo dostępne są następujące operatory</p> <ul style="list-style-type: none"> {n} – element ma się powtórzyć n razy {n,} – element ma się powtórzyć co najmniej n razy {n,m} – element ma się powtórzyć co najmniej n razy, ale nie więcej niż m razy <p>Przykład: kod pocztowy</p> <pre>var kod = /\d{2}-\d{3}/ /[0-9][0-9]-[0-9][0-9][0-9]/</pre> <div>  </div>	Dodatkowo dostępne są następujące operatory {n} – element ma się powtórzyć n razy {n,} – element ma się powtórzyć co najmniej n razy {n,m} – element ma się powtórzyć co najmniej n razy, ale nie więcej niż m razy
Slajd 34	<div>   JavaScript </div> <h2>Kwantyfikatory zachłanne i leniwe</h2> <ul style="list-style-type: none"> - zachłanne – próbują dopasować się do jak najdłuższych podciągów znaków <pre>var wyrażenie = /<.*>/ var wynik = wyrażenie.exec("aaa
 bbb
 ccc"); wyrażenie = /<[^>]*>/</pre> <ul style="list-style-type: none"> - leniwe – powstają przez dodanie dodatkowego znaku zapytania (??, *?, +?, {n}?, {n,}?, {n,m}?) <div>  </div>	Wszystkie opisane wcześniej kwantyfikatory określone są jako zachłanne- będą próbowały dopasować się do jak najdłuższych podciągów. Rozważmy taki przykład – wyrażenie ma wyszukiwać znaczniki html w tekście. Wiemy że znaczniki to fragmenty tekstu ujęte w nawiasy trójkątne. Spodziewamy się wyników ale wyrażenie dopasowało jeszcze tekst bbb . Fragment ten także pasuje do naszego wyrażenia, ale nie jest poprawnym znacznikiem html. Aby to poprawić należałoby użyć następującego wyrażenia: /<[^>]*>/ określi ono, że między nawiasami nie może znaleźć się nawias zamykający. Innym sposobem jest wykorzystanie kwantyfikatorów leniwych, powstałych przez dodanie do zwykłych kwantyfikatorów

Slajd
35

dodatkowego pytanika.



B2E
BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Grupy

(abc)
 (abc (123)?)+
 aaa (bbb|ccc|ddd)
 ;([^;]+);\1;
 0testabc1testxyz
 \d[a-z]{4}(?=abc)
 \d[a-z]{4}(?!=abc)


DAILY
GROUP

Wcześniejsze metody pokazywały jak określić liczbę powtórzeń określonego znaku. Często sytuacją jest jednak powtarzanie się we wzorcu grup znaków. Aby oznaczyć grupę należy wpisać ją w nawiasy okrągłe. Pierwszy przykład pokazuje grupę składającą się z sekwencji abc. Drugi grupę składającą się z sekwencji abc, po której może wystąpić sekwencja cyfr 123. Cała grupa może pojawić się raz lub więcej.

Możemy także zdefiniować kilka różnych wersji ciągu. W przykładzie trzecim pokazano sekwencję aaa po której może wystąpić albo sekwencja bbb, ccc lub ddd. Takie warianty należy oddzielić od siebie znakiem pionowej linii.

Grupowanie daje nam dodatkowo możliwość odwołania się do ciągu odnalezionego w tekście. Każda grupa wyodrębniona za pomocą nawiasu ma swój kolejny numer. Możemy się do takiej grupy odwołać za pomocą wyrażenia \numer. W przykładzie sprawdzamy czy w ciągu wartości oddzielonych średnikiem nie ma dwóch takich samych umieszczonych obok siebie

Inną dodatkową możliwością jest spojrzenie w przód – sprawdzenie czy w dalszej części wyrażenia występuje lub nie występuje określony ciąg.

Stosujemy do tego operatory ?= i ?!=

W powyższym przykładzie w tekście 0testabc1testxyz w 1 przypadku zostanie zwrócony tekst 0test (cyfra, 4 znaki a-z a po nich występuje ciąg abc) a w drugim przypadku 1test – pierwszy fragment jest pominięty, ponieważ chcemy w wyniku otrzymać wyrażenie po którym nie następuje ciąg abc.

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Slajd
36



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript


Ćwiczenia

Napisz skrypt sprawdzający poprawność wpisanych danych (np. numer telefonu, kod pocztowy, imię i nazwisko wpisane z dużych liter)


DAILY
 GROUP

Slajd
37


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```

<html><head>
<title> JavaScript</title>
<script type="text/javascript" src="19cw5.js" >
function sprawdz() {
    var regImie = new RegExp("[A-Z][a-z]+");
    var regNrTel = new RegExp("\\+\\d{2}\\(0?\\d{2}\\)\\d{7}");
    var imie = document.getElementById("imie");
    var telefon = document.getElementById("telefon");
    if (imie.value == regImie.match() && telefon.value == regNrTel.match()
    )
        alert("OK");
    }
</script>
</head>
    
```


DAILY
 GROUP

Człowiek - najlepsza inwestycja

Slajd
 38




JavaScript

Przykładowe rozwiązanie

```

<body>
Proszę podać swoje dane: <br />
<form name="form_nr_1" action="form.php" method="get">
<table border="0">
<tr><td><b>Dane personalne:</b></td><td></td></tr>
<tr><td>imię:</td><td><input type="text" name="imie"></td></tr>
<tr><td>nazwisko:</td><td><input type="text" name="nazwisko"></td></tr>
<tr><td>nr telefonu:</td><td><input type="text" name="telefon"> format
+48 (022) 0000000</td></tr>
<tr><td>kod pocztowy:</td><td><input type="text" name="kod"> format 00-
000</td></tr>
<tr><td align="center" colspan="2"><input type="button" name="wyslij"
value=" Wyślij! " onclick="check();"></td></tr>
</table>
</form></body></html>
    
```



 Slajd
 39




JavaScript

Dziękuję za uwagę



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



11.9.4 Ćwiczenia

Na slajdach zaprezentowano kilka podstawowych ćwiczeń wykorzystujących zaprezentowane możliwości języka. Pierwszym jest ćwiczenie zmieniające kolor przycisku (slajd 6) obrazujący wykorzystanie JS do zmiany wyglądu witryny www.

Człowiek - najlepsza inwestycja

Drugim ćwiczeniem jest ćwiczenie wymagające napisania funkcji walidującej wartość wpisana w pole formularza (slajd 13)

11.9.5 Opis założonych osiągnięć ucznia

Uczestnik po zakończeniu lekcji zna podstawowe sposoby wykorzystania języka JavaScript przy tworzeniu stron www. Potrafi za pomocą wyrażeń regularnych przeszukiwać treść strony.

11.10 Lekcja 10 – Biblioteka JQuery

11.10.1 Cel lekcji

Celem lekcji jest wprowadzenie do zaawansowanego, ale bardzo prostego w użyciu frameworka JavaScript – jQuery. Jest to biblioteka funkcji napisana w języku JavaScript rozszerzająca i upraszczająca część rozwiązań zaprezentowanych w poprzedniej lekcji.

11.10.2 Sposoby osiągania celów kształcenia



Czynności nauczyciela	Czynności ucznia
wprowadza nowe pojęcia związane z tematem zajęć (materiał ze slajdów) przygotowuje ćwiczenia z zaprezentowanego materiału	Wykonuje zadane przez nauczyciela ćwiczenia na komputerze
Prowadzi z uczniami dyskusję na temat wybranych przez nich rozwiązań ćwiczeń	Prezentuje rozwiązania ćwiczeń nauczycielowi i pozostałym uczniom, tłumaczy wybrane przez siebie rozwiązanie
Podsumowuje lekcję, ocenia pracę uczniów na zajęciach.	Dokonuje samooceny swojej pracy, omawia z nauczycielem rozwiązania ćwiczeń


11.10.3 Treść – slajdy z opisem

Slajd 1		W lekcji przedstawimy krótki opis frameworka jQuery, który jest jednym z najczęściej wykorzystywanym zbiorem funkcji w JavaScript. Zaprezentujemy podstawowe mechanizmy jQuery, jego składnię a także kilka praktycznych przykładów wykorzystania w witrynach WWW.
------------	--	--

Człowiek - najlepsza inwestycja


Slajd
2

 **B2E**
BUSINESS TO EDUCATION
  **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY



jQuery

- lekka, szybka biblioteka JavaScript
- "pisz mniej, zrób więcej"
- umożliwia przeszukiwanie drzewa DOM, obsługę zdarzeń, animację, komunikację Ajax.
- wykorzystywane przez wielkie firmy (Microsoft, IBM, Google)
- proste API działające w wielu przeglądarkach (nawet w Internet Explorer 6!)

 **DAILY**
GROUP

JavaScript

jQuery to lekka i szybka biblioteka JavaScript.

"Write less, do more"

Zawiera następujące funkcjonalności:

- manipulację dokumentem html/ drzewem DOM
- manipulację CSS
- obsługę zdarzeń html
- efekty i animacje,
- AJAX,
- narzędzia
- system wtyczek pozwalający na rozszerzenie biblioteki

Zawiera proste do wykorzystania API, które działa w wielu przeglądarkach (obsługą różnic w interpretacji JavaScript zajęli się twórcy biblioteki)

jQuery jest wykorzystywane m.in. przez Wordpress (system do blogów, zarządzania treścią) oraz przez Wikipedię, Google, Microsoft itp.

Człowiek - najlepsza inwestycja


B2E
 BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Dodawanie jQuery

Aby wykorzystywać jQuery musimy załączyć do naszej strony plik z kodem biblioteki.

Możemy pobrać go ze strony www.jquery.com, lub wykorzystać mechanizm CDN – content delivery network.

```
<script src="jquery-1.9.1.min.js"></script>
<script
src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.
min.js"></script>
```



Aby wykorzystywać jQuery musimy pobrać i załączyć do naszej strony plik z kodem biblioteki. Jest on dostępny na stronie www.jquery.com. Aktualna wersja to 1.9.1. Możemy pobrać plik w 2 wersjach zminimalizowanej lub normalnej. Wersja zminimalizowana ma usunięte wszystkie zbędne białe znaki. Jej celem jest maksymalne zmniejszenie rozmiaru pliku, co pozwala na szybsze załadowanie go do przeglądarki. Jeśli chcemy natomiast poznać budowę poszczególnych funkcji warto ściągnąć wersję normalną, bardziej czytelną dla człowieka.

Zamiast pobierać plik i podłączać go do naszej strony możemy wskazać odwołanie do skryptu wykorzystując mechanizm CDN – content delivery network. Są to specjalne serwisy przechowujące i udostępniające treści w internecie. Na slajdzie przedstawiłem przykład pobrania skryptu ze stron Google.


Zaletą podłączenia skryptu z CDN jest fakt, że w trakcie przeglądania internetu nasza przeglądarka już mogła pobrać jQuery (przy odwiedzeniu innej strony korzystającej z tej biblioteki) i sam plik został już umieszczony w pamięci podręcznej przeglądarki.


Człowiek - najlepsza inwestycja


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Slajd
4


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Składnia jQuery

Podstawowa składnia to:

```
$(selector).action()
```

gdzie:


- znak \$ - wskazanie że wykorzystujemy jQuery,
- (selector) – wyrażenie do wyszukania elementu
- action() - akcja do wykonania

Przykłady:

```
$(this).hide() – ukrywa bieżący element
```

```
$("p").hide() – ukrywa wszystkie paragrafy
```

```
$(".test").hide() – ukrywa wszystkie elementy o klasie test
```


DAILY
 GROUP

W jQuery wyszukujemy element html i wykonujemy na nim akcje. Podstawowy zapis to `$(selector).action()`, gdzie znak \$ oznacza, że wykorzystujemy jQuery, (selector) – wyrażenie służące do wyszukania elementu, action() - akcja do wykonania

Przykłady:

```
$(this).hide() – ukrywa bieżący element
```

```
$("p").hide() – ukrywa wszystkie paragrafy
```

```
$(".test").hide() – ukrywa wszystkie elementy o klasie test
```

Istnieje możliwość, że zmienna \$ została już wykorzystana przez inną bibliotekę. Wtedy należy wykorzystać opcję jQuery – `noConflict()`. Pozwoli to na wykorzystywanie innych bibliotek js, a także na wykorzystanie zmiennej \$ przy dostępie do funkcji jQuery.

Standardowe wywołanie to `$.noConflict()`. Należy umieścić je przed każdym skryptem wykorzystującym jQuery.




Człowiek - najlepsza inwestycja



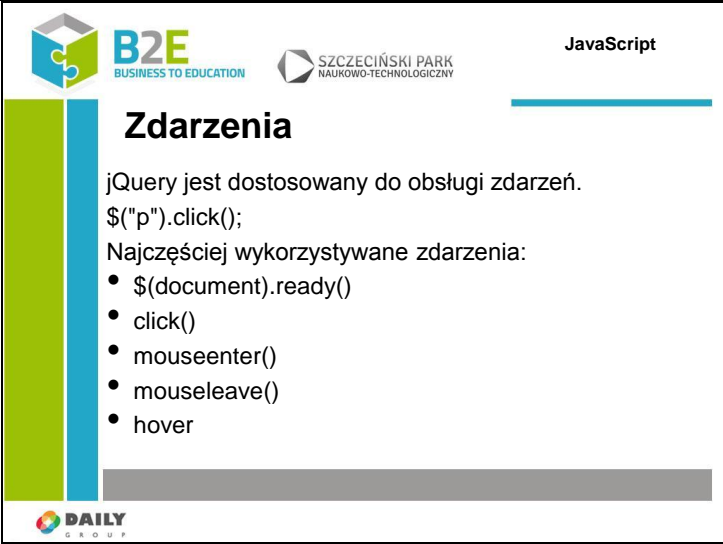
KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY






Slajd 5	<div>   <div>JavaScript</div> </div> <h2>zdarzenie Document Ready</h2> <p>Zdarzenie generowane po załadowaniu dokumentu. Wykorzystywane aby opóźnić wywołanie funkcji jQuery aż do momentu aż cała strona będzie dostępna</p> <p>W skrypcie należy umieścić wywołanie:</p> <pre>\$(document).ready(function() { //tutaj nasze instrukcje });</pre> <p>lub wersję skróconą:</p> <pre>\$(function() { // tutaj instrukcje });</pre> <div>  </div>
------------	--




Człowiek - najlepsza inwestycja

		<p>równym przekazanej wartości.</p> <p>Selektor <code>this</code> zwraca bieżący element.</p> <p>Selektor atrybutu <code>[href]</code> zwróci wszystkie elementy, które zawierają atrybut <code>href</code></p>
Slajd 7		<p>jQuery jest dostosowany do obsługi zdarzeń. O zdarzeniach mówiliśmy na jednej z wcześniejszych lekcji. Jako przypomnienie zdarzenia pozwalają nam na reakcję na działania użytkowników na stronie. Tworząc odpowiednie metody obsługi zdarzeń możemy rozszerzać statyczne strony o interaktywne elementy. Aby przypisać zdarzenie do elementu należy zastosować następującą składnię. Po selektorze elementu stawiamy <code>.</code> a po niej nazwę zdarzenia. Większość zdarzeń DOM ma swój odpowiednik w jQuery.</p> <p>Najczęściej wykorzystywane zdarzenia:</p> <ul style="list-style-type: none"> <code>\$(document).ready()</code> – odpalane po załadowaniu dokumentu <code>click()</code> – odpalane po kliknięciu na element <code>mouseenter()</code> – odpalane po najechaniu kursorem na element <code>mouseleave()</code> – odpalane po opuszczeniu elementu przez kursor <code>hover()</code> – ma 2 argumenty, jest połączeniem <code>mouseenter</code> i <code>mouseleave</code>

Człowiek - najlepsza inwestycja


Slajd 8	<div>  B2E BUSINESS TO EDUCATION </div> <div>  SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY </div> <div>JavaScript</div> <hr/> <h2>Efekty jQuery</h2> <p>Podstawowe efekty w jQuery:</p> <ul style="list-style-type: none"> • ukrywanie/pokazywanie • zanikanie/rozświetlanie • zwijanie/rozwijanie • animacje <div>  DAILY GROUP </div>
------------	---


Człowiek - najlepsza inwestycja

Slajd 10	<div>  B2E BUSINESS TO EDUCATION </div> <div>  SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY </div> <div>JavaScript</div> <hr/> <h2>Efekty - znikanie</h2> <p>W jQuery można wykonać animację znikania i pojawiania się elementu. Służą do tego akcje:</p> <ul style="list-style-type: none"> <code>fadeIn()</code> <code>fadeOut()</code> <code>fadeToggle()</code> <code>fadeTo()</code> <div>  DAILY GROUP </div>
-------------	---

Człowiek - najlepsza inwestycja

Slajd
 12


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Animacje

Metoda animate() pozwala na utworzenie własnych animacji

```

$(selektor).animate(
    {parametry},
    predkosc,
    funkcja_zwrotna);

$("button").click(function() {
    $("div").animate({left:'250px'});
    });
    
```

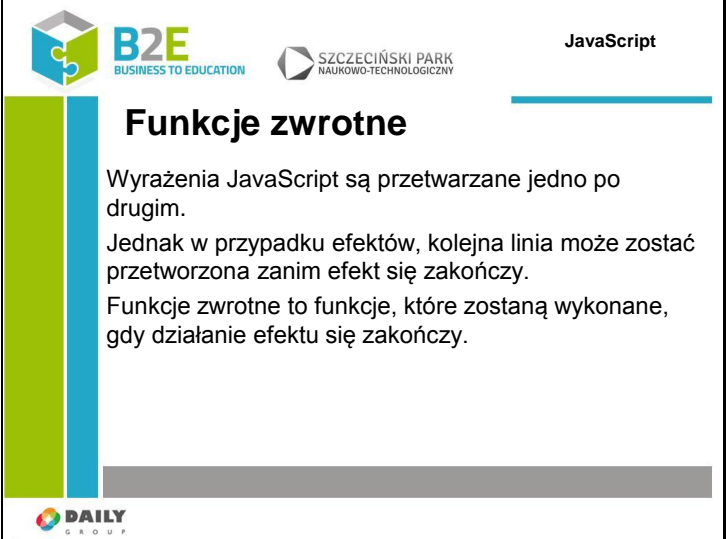
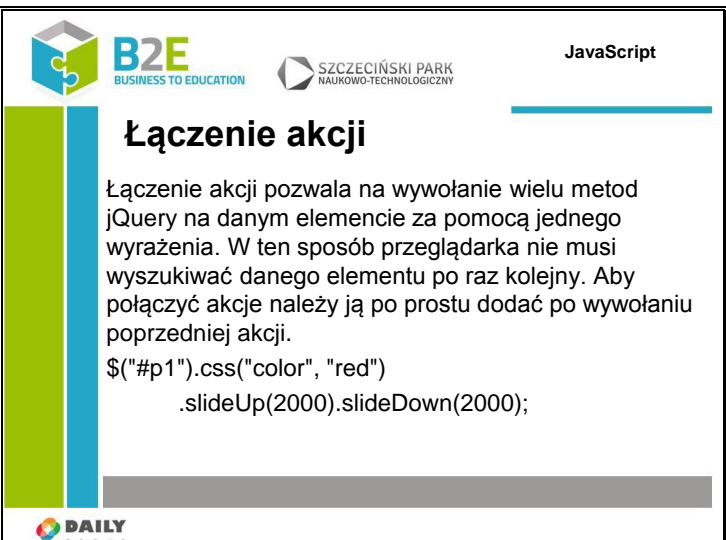

DAILY
 GROUP

Metoda animate pozwala na utworzenie własnych animacji. Wymagany argument "parametry" definiuje właściwości css, które będą animowane. Powyższy przykład po naciśnięciu przycisku przesunie element w lewo aż jego właściwość left osiągnie 250 pikseli. Domyślnie elementy HTML mają właściwość static – nie można ich przesuwac. Przed zdefiniowaniem animacji trzeba ustawić właściwość position elementu na relative, fixed lub absolute.







Można w parametrach przekazać więcej niż 1 właściwość. Możemy stworzyć animację, która przesunie element i zwiększy jego rozmiar. Jak powinna wyglądać taka funkcja?

Za pomocą metody animate można zmieniać większość właściwości styli css – trzeba tylko pamiętać o zmianie trybu pisania na camel case. zamiast pisać padding-left napiszemy paddingLeft itd. Podstawowa biblioteka jQuery nie obsługuje animacji atrybutu color. Aby animować zmiany kolorów trzeba pobrać odpowiednią wtyczkę z witryny jquery. jQuery automatycznie tworzy kolejkę dla animacji. Możemy zdefiniować kilka animacji a jQuery automatycznie wyświetli je jedna po drugiej.


Człowiek - najlepsza inwestycja


		<p>Animacje można zatrzymać wykorzystując metodę <code>stop</code> z opcjonalnymi argumentami <code>stopAll</code> (powoduje zatrzymanie całej kolejki animacji) i <code>goToEnd</code> – przechodzi na koniec animacji. Oba argumenty mają domyślną wartość <code>false</code>, co oznacza że zatrzymana zostanie tylko bieżąca animacja.</p>
Slajd 13		<p>Wyrażenia JavaScript są przetwarzane jedno po drugim. Jednak w przypadku efektów, kolejna linia może zostać przetworzona zanim efekt się zakończy. Funkcje zwrotne to funkcje, które zostaną wykonane, gdy działanie efektu się zakończy.</p>
Slajd 14		<p>Łączenie akcji pozwala na wywołanie wielu metod jQuery na danym elemencie za pomocą jednego wyrażenia. W ten sposób przeglądarka nie musi wyszukiwać danego elementu po raz kolejny. Aby połączyć akcje należy ją po prostu dodać po wywołaniu poprzedniej akcji. Przykład obrazuje działanie na obiekcie o id <code>#p1</code> – zmieniamy jego kolor na czerwony, po czym zwiżamy obiekt a następnie rozwijamy. Operacje zwiżania i</p>

Człowiek - najlepsza inwestycja

		rozwijania trwają po 2 sekundy
Slajd 15	  <div style="float: right;">JavaScript</div> <hr style="width: 20%; margin-left: auto;"/> <h2 style="text-align: center;">Manipulacja DOM</h2> <p>jQuery zawiera metody do manipulacji elementami html i ich atrybutami. Wykorzystuje do tego metody działające na modelu dokumentu (DOM)</p> <p>DOM – interfejs, który pozwala programom i skryptom na dynamiczny dostęp i modyfikację treści, struktury i stylu dokumentu</p> 	jQuery zawiera metody do manipulacji elementami html i ich atrybutami. Wykorzystuje do tego metody działające na modelu dokumentu (DOM) Jak wspominaliśmy wcześniej DOM to interfejs, który pozwala programom i skryptom na dynamiczny dostęp i modyfikację treści, struktury i stylu dokumentu.
Slajd 16	  <div style="float: right;">JavaScript</div> <hr style="width: 20%; margin-left: auto;"/> <h2 style="text-align: center;">Pobieranie treści text() html()</h2> <ul style="list-style-type: none"> • text() <pre>\$("#btn1").click(function(){ alert("Text: " + \$("#test").text()); });</pre> • html() <pre>\$("#btn2").click(function(){ alert("HTML: " + \$("#test").html()); });</pre> 	Podstawowymi metodami do pobierania treści dokumentu są metody: text() – pobiera lub ustawia wartość tekstową wybranego elementu. html() – pobiera lub ustawia treść elementu (razem ze znacznikami html)

Człowiek - najlepsza inwestycja


B2E
 BUSINESS TO EDUCATION


SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript

Metody val() i attr()

- **val()**

```
$("#btn1").click(function(){
    alert("Value: " + $("#test").val());
});
```
- **attr()**

```
$("#button").click(function(){
    alert($("#link").attr("href"));
});
```




DAILY
 GROUP

val() – pobiera lub ustawia wartość pola formularza. W przypadku ustawiania pól typu select i checkbox do pobrania wartości można użyć selektorów :selected i :checked. Jeśli select nie ma ustawionej wartości wynikiem działania funkcji val będzie null – jeśli wartość jest ustawiona to w wyniku otrzymamy tablicę z zaznaczonymi elementami. Dzieje się tak dlatego, że w znaczniku select możemy wybrać więcej niż jedną wartość (ustawiając atrybut multiselect).

Metoda attr() pobiera wartość danego atrybutu


Wszystkich wymienionych teraz funkcji można do ustawienia wartości danego elementu. Wartość tę należy umieścić w nawiasie jako parametr metody. Zamiast wartości możemy podać funkcję zwrótną, która jako argumenty przyjmuje indeks bieżącego elementu w zaznaczeniu, oraz wartość oryginalną elementu. Jako wartość wynikową ustawiamy tekst, który ma być nową wartością elementu.

Slajd
18


B2E
 BUSINESS TO EDUCATION
 
SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY
 JavaScript

Dodawanie elementów

- `append()`
`$("p").append("dodano z jQuery.");`
- `prepend()`
`$(p).prepend("dodano z jQuery.");`
- `after()`
- `before()`



Za pomocą jQuery można bardzo łatwo dodawać elementy i treść do strony. Służą do tego następujące metody:

- `append()` – dodaje treść na końcu danego elementu.



Założmy, że mamy document zawierający jakiś paragraf o treści Witaj świecie!. Wywołanie metody

`$("p").append("dodano z jQuery.");` spowoduje, że paragraf ten będzie miał teraz treść: Witaj świecie! dodano z jQuery.;

- `prepend()` – dodaje treść na początku danego elementu. W powyższym przykładzie wynikiem będzie tekst: "dodano z jQuery.Witaj świecie!"

Metody `after` i `before` wstawiają treść odpowiednio przed i po elemencie – nie tak jak wcześniejsze na początku lub końcu wartości elementu.


 Slajd
19


B2E
 BUSINESS TO EDUCATION
 
SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY
 JavaScript

Usuwanie elementów

Do usuwania elementów służą metody:

- `remove()`
`$("div").remove()`
`$("p").remove(".italic");`
- `empty()`
`$("div").empty();`



Do usuwania elementów służą metody:


- `remove()`
- `empty()`


Metoda `remove` usuwa wszystkie węzły potomne elementu oraz sam element, jako parametr możemy podać selektor elementów do usunięcia. W przykładzie usuwamy wszystkie elementy `p`, które mają klasę `italic`.

Metoda `empty` usuwa wszystkie węzły potomne danego elementu.

Człowiek - najlepsza inwestycja

Slajd
20


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Manipulacja CSS

Do zmiany stylu css elementów służą następujące metody:

- `addClass()`
- `removeClass()`
- `toggleClass()`
- `css()`


DAILY
 GROUP

Do zmiany stylu css elementów służą następujące metody:

- `addClass()` – metoda pozwala na dodanie klasy do wybranego elementu na stronie. Jako parametr podajemy nazwę klasy css zdefiniowaną w pliku stylu. Możemy dodać od razu kilka klas
- `removeClass()` – metoda usuwa podaną klasę z elementu.
- `toggleClass()` – metoda dodaje lub usuwa klasę z elementu
- `css()` – ustawia lub zwraca jeden lub wiele właściwości stylu dla wybranego elementu; aby pobrać wartość elementu należy wywołać metodę z nazwą właściwości, aby ustawić – oprócz nazwy podajemy nową wartość. W przypadku pobierania wartości zostanie zwrócona wartość pierwszego dopasowanego obiektu do selektora. Ustawienie wartości spowoduje zmiany we wszystkich elementach wyszukianych przez selektor.







Człowiek - najlepsza inwestycja



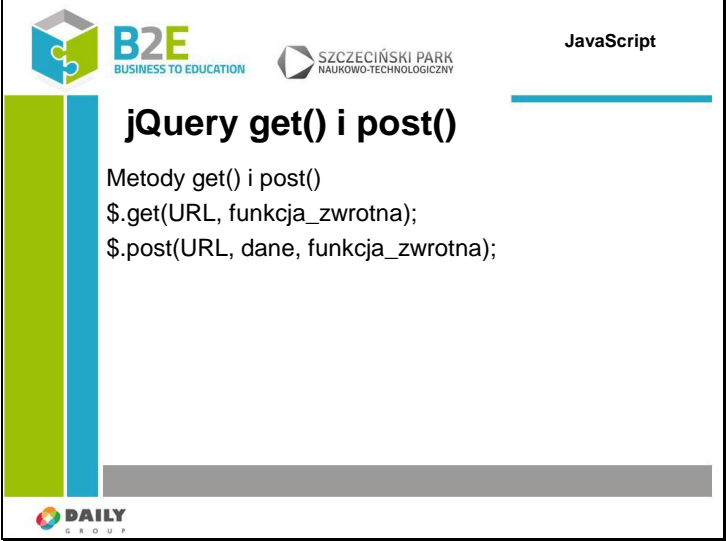
KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY




<p>Slajd 21</p>	  <div>JavaScript</div> <hr/> <h2>jQuery i AJAX</h2> <ul style="list-style-type: none"> co to jest AJAX? Asynchronous JavaScript and XML <p>w skrócie: służy do pobierania danych w tle i wyświetlania ich na stronie bez konieczności przeładowania strony</p> 	<p>Co to jest AJAX? Asynchronous JavaScript and XML. Służy do pobierania danych w tle i wyświetlania ich na stronie bez konieczności przeładowania strony. Pisanie wywołań AJAX jest utrudnione, różne przeglądarki inaczej implementują AJAX. Oznacza to konieczność napisania dodatkowego kodu dla różnych przeglądarek. W jQuery funkcjonalność ta już została zaimplementowana i można jej używać używając pojedynczej linii kodu.</p>
<p>Slajd 22</p>	  <div>JavaScript</div> <hr/> <h2>metoda load()</h2> <p>Metoda load() pobiera dane z serwera i umieszcza je w wybranym elemencie</p> <pre>\$(selektor).load(url, dane, funkcja_zwrotna) \$("#div1").load("dane.txt");</pre> 	<p>Metoda load() pobiera dane z serwera i umieszcza je w wybranym elemencie. Polecenie może mieć 3 parametry:</p> <ul style="list-style-type: none"> - obowiązkowy URL, który wskazuje adres serwera (pliku lub usługi), - opcjonalne dane: zawierający informacje które chcemy wysłać razem z żądaniem (np. identyfikatory, nagłówki autoryzacji itp.) - opcjonalne funkcje_zwrotną, która zostanie wywołana po odebraniu danych. <p>Funkcja zwrotna ma 3 parametry: responseTxt – tekst odpowiedzi, statusTXT – zawiera status żądania, xhr – zawiera obiekt XMLHttpRequest</p>


Człowiek - najlepsza inwestycja

		<p>Założmy że mamy na dysku plik dane.txt zawierający dane, które chcemy wyświetlić w elemencie div1. Użyjemy do tego polecenia</p> <pre>\$("#div1").load("dane.txt");</pre>
Slajd 23		<p>Metody get i post wysyłają żądania do serwera za pomocą żądań HTTP GET i HTTP POST.</p> <p>GET – pobiera dane z określonego adresu POST – wysyła dane na określony adres</p> <p>Funkcja get() przyjmuje 2 parametry:</p> <ul style="list-style-type: none"> - obowiązkowy URL – adres zasobu, który chcemy pobrać - i opcjonalny funkcja_zwrotna (callback) – funkcja wywołana po odebraniu danych <p>Funkcja post() przyjmuje 3 argumenty:</p> <ul style="list-style-type: none"> - obowiązkowy URL – adres na który wysyłamy żądanie. - opcjonalny dane - dane które chcemy wysłać. Dane muszą być w formacie JSON - opcjonalny funkcja_zwrotna – funkcja, która zostanie uruchomiona po wysłaniu danych

Człowiek - najlepsza inwestycja

Slajd
24



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Ćwiczenia


Napisz skrypt w którym po naciśnięciu przycisku pojawi się komunikat z treścią zawartą w znaczniku p


DAILY
 GROUP

Napisz skrypt w którym po naciśnięciu przycisku pojawi się komunikat z treścią zawartą w znaczniku p.

Slajd
25


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```
$(document).ready(function() {
    $("#btn1").click(function() {
        alert($("#p").text());
    });
});
(...)
```

```
<body><p>To jest nasza strona</p>
<button id="btn1"> Wyświetl </button>
</body>
```


DAILY
 GROUP

Człowiek - najlepsza inwestycja




KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI


UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
26



B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Ćwiczenia


Napisz skrypt w którym po naciśnięciu przycisku do paragrafu został dodany tekst "dodany tekst"


DAILY
 GROUP

Napisz skrypt, w którym po naciśnięciu przycisku do paragrafu został dodany tekst "dodany tekst".

Slajd
27


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie

```
$(document).ready(function() {
    $("#btn1").click(function() {
        $("p").append("dodany tekst");
    });
});
(...)
```

```
<body><p>To jest nasza strona</p>
<button id="btn1"> Dodaj tekst</button>
</body>
```


DAILY
 GROUP

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY




Slajd
28

 **B2E**
BUSINESS TO EDUCATION
  **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY

JavaScript



Ćwiczenia

Napisz skrypt w którym po naciśnięciu przycisku kolor tekstu w paragrafie zmieni się na niebieski.

 **DAILY**
GROUP

Napisz skrypt w którym po naciśnięciu przycisku kolor tekstu w paragrafie zmieni się na niebieski.

Slajd
29


 **B2E**
BUSINESS TO EDUCATION
  **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY

JavaScript

Przykładowe rozwiązanie


```
$(document).ready(function() {
    $("#btn1").click(function() {
        $("p").css(color: blue;);
    });
});
(...)
```

```
<body><p>To jest nasza strona</p>
<button id="btn1"> Wyświetl </button>
</body>
```


 **DAILY**
GROUP

Człowiek - najlepsza inwestycja

Slajd
30



B2E
BUSINESS TO EDUCATION




SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

JavaScript

Ćwiczenia


Napisz skrypt w którym po naciśnięciu przycisku tekst paragrafu zostanie pogrubiony




DAILY
GROUP

Napisz skrypt w którym po naciśnięciu przycisku tekst paragrafu zostanie pogrubiony.

Slajd
31



B2E
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY


JavaScript

Przykładowe rozwiązanie

```
$(document).ready(function() {
  $("#btn1").click(function() {
    $("#txt").html("<b>"+"$("#txt").text()+"</b>");
  });
});
(...)
```

`<p id="txt">To jest przykładowy tekst</p>`

`<button id="btn1">Ustaw pogrubiony</button>`



DAILY
GROUP

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Slajd
 32







JavaScript


Podsumowanie

W trakcie tej lekcji poznaliśmy podstawowe informacje o frameworku jQuery. Wiemy, że pozwala on m.in. na:

- animowanie elementów na stronie (pokazywanie, ukrywanie, przesuwanie itp.)
- wyszukiwanie i modyfikację treści elementów
- dodawanie i usuwanie elementów
- modyfikację stylu css strony
- komunikację z innymi usługami z wykorzystaniem AJAX

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



W trakcie tej lekcji poznaliśmy podstawowe informacje o frameworku jQuery. Wiemy, że pozwala on m.in. na:

- animowanie elementów na stronie (pokazywanie, ukrywanie, przesuwanie itp.)
- wyszukiwanie i modyfikację treści elementów
- dodawanie i usuwanie elementów
- modyfikację stylu css strony
- komunikację z innymi usługami z wykorzystaniem AJAX

11.10.4 Ćwiczenia

Napisz skrypt dodający efekt "zebry" w tabeli (wiersze parzyste i nieparzyste mają inne kolory). Dodatkowo niech wiersz nagłówkowy zawiera test pogrubiony.

Utwórz stronę zawierającą prosty formularz (zawierający jedno pole o identyfikatorze "name") i element div. Po wprowadzeniu znaku do pola dodaj wpisany znak do utworzonej warstwy. Wykorzystaj zdarzenie keyup.

Fragment przykładowego rozwiązania (część jQuery):

```
<script type="text/javascript">

$(document).ready(function() {

    $("#name").bind('keyup', function() {

        $("#div").append('').append($(this).val());

    });

});

</script>
```

Człowiek - najlepsza inwestycja

11.10.5 Opis założonych osiągnięć ucznia






Uczestnik po zakończeniu lekcji potrafi dodać do strony bibliotekę jQuery, a także wykorzystać ją podczas tworzenia witryny www. Potrafi dynamicznie zmieniać wygląd strony a także dodać podstawowe efekty na stronie.

11.11 Lekcja 11 – Interdyscyplinarny projekt książki teleadresowej

11.11.1 Cel lekcji



Celem lekcji jest dodanie do zaprojektowanej aplikacji sprawdzenia wprowadzanych danych. W tej lekcji uczniowie będą mogli wykazać się umiejętnościami nabytymi w czasie poprzednich zajęć.

11.11.2 Treść – slajdy z opisem

Slajd 1	  <div>JavaScript</div> <div>JavaScript</div> <div>Projekt interdyscyplinarny Baza teleadresowa koleżanek i kolegów ze szkoły</div> <div><small>Człowiek - najlepsza inwestycja</small></div> <div> <small>KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI</small></div> <div> <small>UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY</small></div> <div><small>Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego</small></div> 	W ramach projektu dodamy do naszej aplikacji funkcje JavaScript sprawdzające poprawność wpisanych danych.
Slajd 2	  <div>JavaScript</div> <div>Założenia</div> <ul style="list-style-type: none"> • Przy usuwaniu rekordów z bazy system wyświetli okno dialogowe z potwierdzeniem usunięcia rekordu • Do każdej klasy będzie dodany wychowawca • Imię i nazwisko wychowawcy oraz uczniów będzie wprowadzone z dużej litery • Pole telefon będzie nieobowiązkowe, natomiast jeśli podamy numer to wymagane będzie wprowadzenie 9 cyfr • Adres e-mail będzie poprawnym adresem 	

Człowiek - najlepsza inwestycja

Slajd
3

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY


JavaScript

Potwierdzenie usuwania

Funkcja confirm()

Funkcja będzie wywoływana w następujących miejscach:

- Usuwanie klas
- Usuwanie uczniów



Do uzyskania potwierdzenia wykorzystamy standardową funkcję JavaScript – confirm(). Zwraca ona wartość true jeśli użytkownik wybierze przycisk "OK" i false jeśli zamknie okno, lub wybierze przycisk "Anuluj".

Wywołanie tej funkcji dodamy w widokach listy klas i listy uczniów.

Przykładową realizacją będzie dodanie klasy do każdego odnośnika, uruchamiającego akcję usuń (w pliku GUI.php) i dodanie skryptu wyświetlającego okno z pytaniem.

```
$(document).ready(function() {
    $(".deleteBtn").click(function(e) {
        if (!confirm("Czy na pewno chcesz usunąć ten rekord?")) {
            e.preventDefault();
        }
    });
});
```

Klasę deleteBtn musimy dodać w metodach:

- displayClasses
- displayClass(id)

Człowiek - najlepsza inwestycja




KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI


UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd
 4


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


DAILY
 GROUP

Przy wysyłaniu formularza tworzenia klasy sprawdzimy czy oba pola są wprowadzone oraz czy w polu wychowawca wprowadzono imię i nazwisko z dużych liter (czy 1 znak to duża litera oraz czy dalej w ciągu jest spacja a po niej kolejna duża litera).

Wykorzystamy do tego wyrażenie regularne

Przykładowy skrypt:

```
var classForm = document.forms["classForm"];
var regWychowawca = new RegExp("[A-Z][a-z]*\s[A-Z][a-z]*");
if (classForm.className == "" || classForm.educator == "" || !classForm.wychowawca
    == regWychowawca.match() ) {
    alert("Nieprawidłowo wprowadzone dane");
    e.preventDefault();
}
```

Człowiek - najlepsza inwestycja





KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY



Slajd
5


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript


Walidacja tworzenia uczniów


Wymagania:

- Imię i nazwisko z dużej litery
- E-mail jeśli jest podany to występuje w nim znak @ oraz .
- Telefon jeśli jest podany składa się z 9 cyfr


DAILY
 GROUP

Slajd
6


B2E
 BUSINESS TO EDUCATION


 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY


JavaScript

Walidacja tworzenia uczniów

- Imię i nazwisko z dużej litery

```

var nameReg = new RegExp("[A-Z][a-z]+");
if(studentForm.imie != nameReg.match() ||
studentForm.nazwisko != nameReg.match()) {
    alert("Imię i nazwisko musi być wpisane z dużej litery");
    e.preventDefault();
}
    
```


DAILY
 GROUP

Pobranie formularza:

```
var studentForm = document.forms["studentForm"];
```

Wyrażenie regularne sprawdzające poprawność wprowadzonego imienia i nazwiska

```
var nameReg = new RegExp("[A-Z][a-z]+");
```

Jeśli dane nie są wprowadzone poprawnie przerywamy wysyłanie skryptu. Do bazy

Człowiek - najlepsza inwestycja

danych nie zostanie dodany nowy student.

```
$( "#studentForm" ).submit(function(e) {
    var result = true;
    console.log(document.forms);
    var sf = document.forms["studentForm"];
    if (sf.imie == "" || sf.nazwisko == "") {
        alert("Imię i nazwisko muszą być wypełnione");
        result = false;
    }
    var nameReg = new RegExp("[A-Z][a-z]+");
    if (sf.imie != nameReg.match() || sf.nazwisko != nameReg.match()) {
        alert("Imię i nazwisko musi być wprowadzone z dużych liter");
        result = false;
    }
    if (!result) {
        e.preventDefault();
    }
});
```

Slajd
7


B2E
 BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript

Walidacja tworzenia uczniów

Wprowadzenie adresu e-mail

Wyrażenie regularne:

```
var emailReg = new RegExp("[A-Za-z0-9]+@[A-Za-z0-9]+.[A-Za-z]{2,}");
```



Dodajemy sprawdzenie adresu email – jeśli jest wprowadzony to musi spełniać następujące warunki:

- Zaczyna się dowolnym ciągiem znaków, po którym występuje znak @,
- Następnie występuje kolejny dowolny ciąg znaków a po nim znak .
- Po ostatniej kropce występują 2 lub więcej znaków (domeny głównego poziomu)


Człowiek - najlepsza inwestycja



KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY

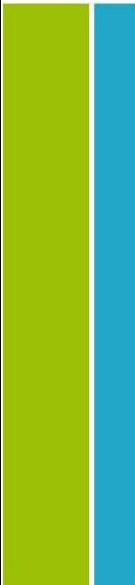

```
var sf = document.forms["studentForm"];
var emailReg = new RegExp("[A-Za-z0-9]+@[A-Za-z0-9]+.[A-Za-z]{2,}");
If (sf.email!= "" && sf.email != emailReg.match()) {
    alert("Nieprawidłowy adres email");
    e.preventDefault();
}
```

Slajd
8


B2E
 BUSINESS TO EDUCATION



 SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

JavaScript



Walidacja tworzenia uczniów

Wprowadzenie numeru telefonu:
 Wyrażenie regularne:
 var telReg= new RegExp("[0-9]{9}");




DAILY
 GROUP

Dodajemy sprawdzenie numeru telefonu – jeśli jest wprowadzony to musi składać się z 9 cyfr.

```
var sf = document.forms["studentForm"];
var telReg = new RegExp("[0-9]{9}");
If (sf.telefon != "" && telefon != telReg.match()) {
    alert("Numer telefonu powinien mieć 9 cyfr");
    e.preventDefault();
}
```



Człowiek - najlepsza inwestycja

Slajd
9


 **B2E**
BUSINESS TO EDUCATION
  **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY
 JavaScript

Podsumowanie

- Dodawanie klas możliwe jest tylko gdy wprowadzimy poprawne dane wychowawcy
- Dodanie uczniów możliwe jest gdy wprowadzimy poprawne imię i nazwisko.
- Jeśli podamy adres email to musi on zawierać znak @ i kropkę.
- Numer telefonu składa się z 9 cyfr.

 **KAPITAŁ LUDZKI**
NARODOWA STRATEGIA SPÓJNOŚCI
  **UNIA EUROPEJSKA**
EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

 **DAILY**
GROUP

Podsumowanie:

- Dodawanie klas możliwe jest tylko gdy wprowadzimy poprawne dane wychowawcy
- Dodanie uczniów możliwe jest gdy wprowadzimy poprawne imię i nazwisko.
- Jeśli podamy adres email to musi on zawierać znak @ i kropkę.
- Numer telefonu składa się z 9 cyfr.

11.11.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie uzupełnią utworzoną przez siebie aplikację o dodatkową walidację wprowadzanych danych. Dane dostępne w książce teleadresowej kolegów i koleżanek z klasy i ze szkoły będą pozbawione błędów.

Człowiek - najlepsza inwestycja