

JavaScript

Typowe błędy

```
<html>
```

```
<head>      text
```

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

function

```
funcion mnozenie(a,b) ;
```

```
{
```

```
  return a*b;
```

```
}
```

```
</script>
```

```
</head>
```

Zjadając ważne znaki
generujesz sobie braki!

```
<body>
```

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

```
document.write("12*5=" + mnozenie(12,5));
```

```
</script>
```

```
</body>
```

```
</html>
```

Cechy języka

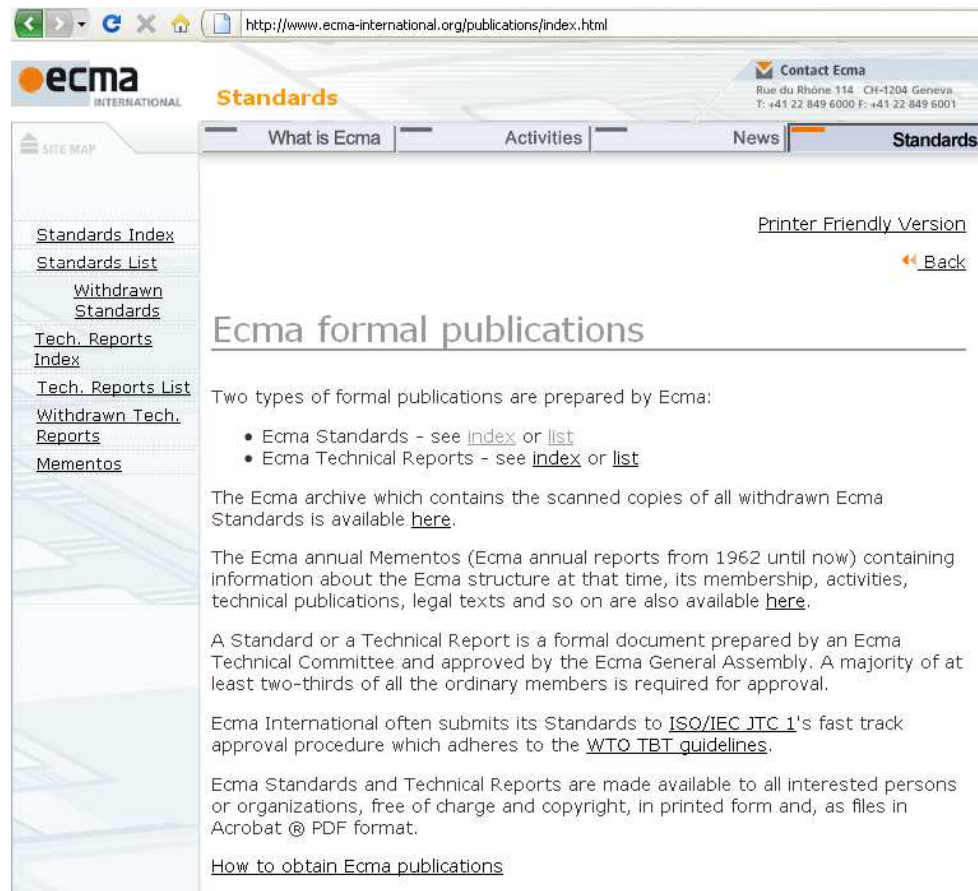
- obiektowy język skryptowy
- składnia i struktura języka podobna do C++ !!
- język interpretowany przez przeglądarkę
(instrukcje są wykonywane bez wcześniejszej kompilacji
- brak pliku wykonywalnego *.exe)
- skrypty są najczęściej umieszczane bezpośrednio na stronie internetowej (można je również łąadować z innych komputerów, plików zewnętrznych)
- dodaje interaktywność do stron internetowych
- każdy może używać języka za darmo
(nie trzeba kupować licencji)
- Java i JavaScript to nie to samo! Java jest językiem programowania o potężnych możliwościach (tak jak np. C++), JavaScript wygląda przy Javie skromnie.

Możliwości

- JavaScript może umieszczać na stronie tekst zmieniający się dynamicznie
- JavaScript obsługuje zdarzenia – np. można napisać kod reagujący na kliknięcie myszą w dany element
- JavaScript może zmieniać treść elementów HTML na stronie
- JavaScript można wykorzystać do walidacji danych, czyli sprawdzenia ich poprawności przed wysłaniem na serwer
- JavaScript może rozpoznać typ przeglądarki, co można wykorzystać do załadowania odpowiedniej strony (w zależności od przeglądarki)
- JavaScript ma dostęp do plików cookies

ECMAScript

Oficjalna nazwa języka to "ECMAScript".
Standard jest utrzymywany i rozwijany przez organizację
ECMA



ECMAScript



Ecma Standards

An index
is also av

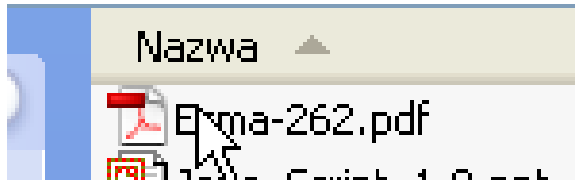
ECMA-261	Broadband Private Integrated Services Network (B-PISN) - Service Description - Broadband Connection Oriented Bearer Services (B-BCSD) (June 1997)	ISO/IEC 15899
ECMA-262	ECMAScript Language Specification, 3 rd edition (December 1999)	ISO/IEC 16262 *
ECMA-263	Private Integrated Services Network (PISN) Specification, Functional Model and Information Call Priority Interruption and Call Priority Protection Supplementary Services (CPPI)	

The screenshot shows the detailed page for ECMA-262. The left sidebar is identical to the previous image. The main content area has the heading 'Standard ECMA-262' and 'ECMAScript Language Specification' in orange. Below this, it says '3rd edition (December 1999)'. A paragraph states: 'This Standard defines the ECMAScript scripting language.' Below this, it says 'Copy these file(s), free of charge:'. A table lists the available files:

File name	Size (Bytes)	Content
ECMA-262.pdf	720 951	Acrobat (r) PDF file

At the bottom, it states: 'This Ecma publication is also approved as ISO/IEC 16262'.

ECMAScript



Standard ECMA-262

3rd Edition - December 1999

ECMA

Standardizing Information and Communication Systems

ECMAScript Language Specification

Ciekawostki

ECMA-262 jest oficjalnym standardem języka JavaScript. Standard jest oparty na językach JavaScript (Netscape) oraz JScript (Microsoft). Głównym twórcą języka jest amerykański programista

Brendan Eich



Ciekawostki

- Język po raz pierwszy pojawił się w przeglądarce Netscape Navigator 2.0 i od roku 1996 stał się dostępny we wszystkich przeglądarkach napisanych przez firmy Netscape i Microsoft
- W roku 1998 standard ECMA został zatwierdzony przez Międzynarodową Organizację Standaryzacyjną ISO (ISO/IEC 16262).
- Język ciągle się rozwija.

Wersje specyfikacji ECMAScript

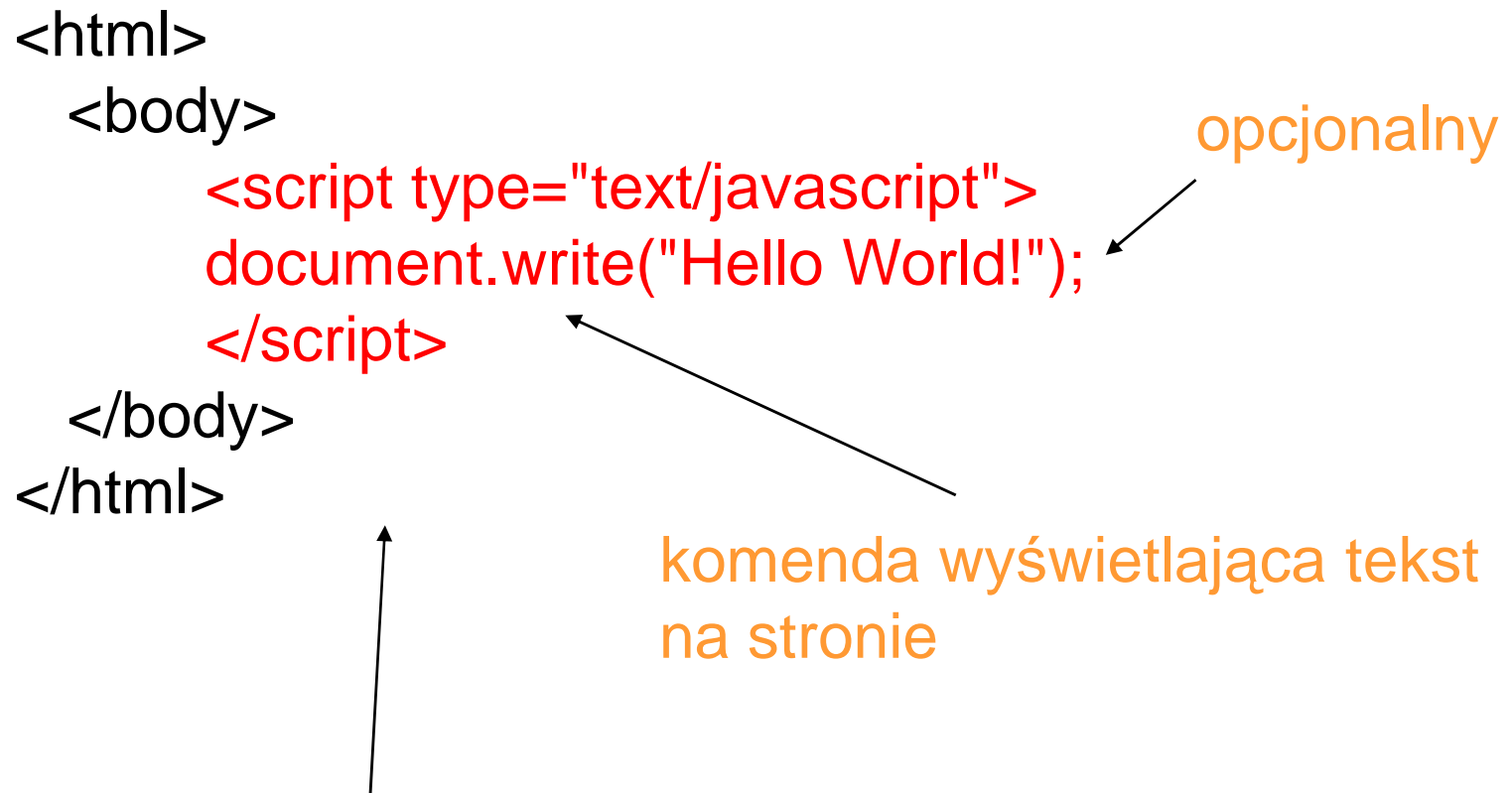
Wersja	Wydana	Zmiany ^[1]
1	czerwiec 1997	
2	czerwiec 1998	Głównie korekta redaktorska ^[2]
3	grudzień 1999	Wyrażenia regularne, obsługa wyjątków, nowe rodzaje pętli, literały tablic i obiektów
4	produkcja	Klasy, jawna kontrola dostępu, alternatywne dla DOM wbudowane wsparcie dla XML ^[3]

Umieszczanie skryptu w kodzie strony

```
<html>  
  <body>  
    <script type="text/javascript">  
      document.write("Hello World!");  
    </script>  
  </body>  
</html>
```

opcjonalny

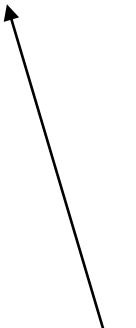
komenda wyświetlająca tekst na stronie



Kod wykonuje się w trakcie ładowania strony
(jest umieszczony w sekcji <body>)

Umieszczanie skryptu w kodzie strony w sekcji <head>

```
<html>
  <head>
    <script type="text/javascript">
      .....
    </script>
  </head>
  <body>
  </body>
</html>
```



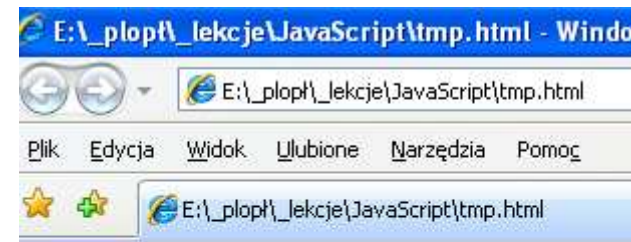
Tutaj umieszczamy kod reagujący na zdarzenia
na stronie

Umieszczanie skryptu w kodzie strony dla przeglądarek nie obsługujących tego języka

Slashe
zabezpieczają
przed wykonaniem
przez JavęScript
znacznika -->

```
<html>
  <body>
    <script type="text/javascript">
      <!--
      document.write("Hello World!");
      //-->
    </script>
  </body>
</html>
```

gdyby nie było komentarzy
przeglądarka wyświetliłaby
skrypt jako treść strony



document.write("Hello World!");

Umieszczanie skryptu w zewnętrznym pliku

```
<html>  
  <head>  
    <script type="text/javascript" src="skrypty.js"> </script>  
  </head>  
  <body>  
  </body>  
</html>
```

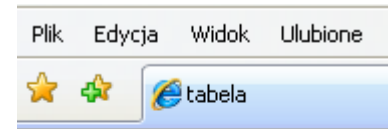
↑

w zewnętrznym pliku
brak znaczników <script>

```
<html>  
  <head>  
    <script type="text/javascript" src="http://www.plopl.com/skrypty.js"> </script>  
  </head>  
  <body>  
  </body>  
</html>
```

Ćwiczenia

- 1.1 Umieść w kodzie strony w sekcji `<body>` skrypt wyświetlający tabelę 2x2. W komórkach tabeli umieść słowo *JavaScript*.



- 1.2 Umieść w kodzie strony w sekcji `<body>` skrypt wyświetlający zdjęcie z wieżą Eiffla (zdjęcie może być inne)



Odpowiedzi do ćwiczeń

1.1

Nie wstawiaj znaków nowej linii (enter)!

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />

<title>tabela</title>
</head>
<body>

<script type="text/javascript">
document.write("<table border='1'"><tr><td>JavaScript</td><td>JavaScript</td></tr><tr><td>JavaScript</td><td>JavaScript</td></tr></table>");
</script>

</body>
</html>
```

aby przeglądarka nie myliła znaku " z końcem stringu
trzeba wstawić znak specjalny \" (czyli \"1\")
(można również .." .. '1' .. " ..)

1.2

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />

<title>Wieża Eiffla</title>
</head>
<body>

<script type="text/javascript">
document.write("<img src='\\wieza_eiffla.jpg\\' alt='\\Wieża Eiffla\\' />");
</script>

</body>
</html>
```

Łamanie linii

```
document.write("Ten tekst napisała funkcja wypisz());
```

Gdy chcesz złamać linie wstaw w stringu \ (backslash)

```
document.write("Ten tekst napisała \  
funkcja wypisz());
```


Funkcje

```
<html>
<head>
<meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=ISO-8859-2">
<script type="text/javascript">
```

funkcja
(deklaracja i definicja)

```
function wypisz()
{
    document.write("Ten tekst napisała funkcja wypisz()");
}
```

ciało funkcji

```
</script>
</head>
<body>
```

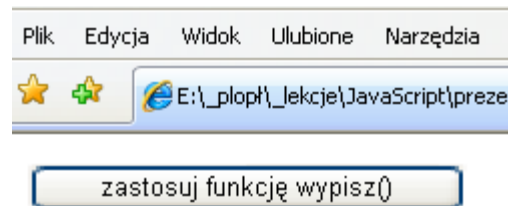
zdarzenie

wywołanie funkcji

```
<input type="button" onclick="wypisz()" value="zastosuj funkcję wypisz()" />
```

```
</body>
</html>
```

przycisk



napis na przycisku

Funkcja przyjmująca argumenty i zwracająca wartość

```
<html>
<head>
<meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=ISO-8859-2">
```

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

```
function mnozenie(a,b)
```

```
{
  return a*b;
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
12*5=
```

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

```
document.write(mnozenie(12,5));
```

```
</script>
```

```
</body>
```

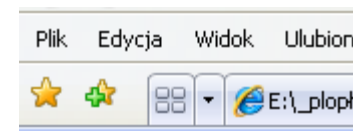
```
</html>
```

argumenty formalne

wartość zwracana

argumenty aktualne

wywołanie funkcji



wygenerowane
przez funkcję

Alert box

```
<html>
```

```
<head>
```

```
<meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=ISO-8859-2">
```

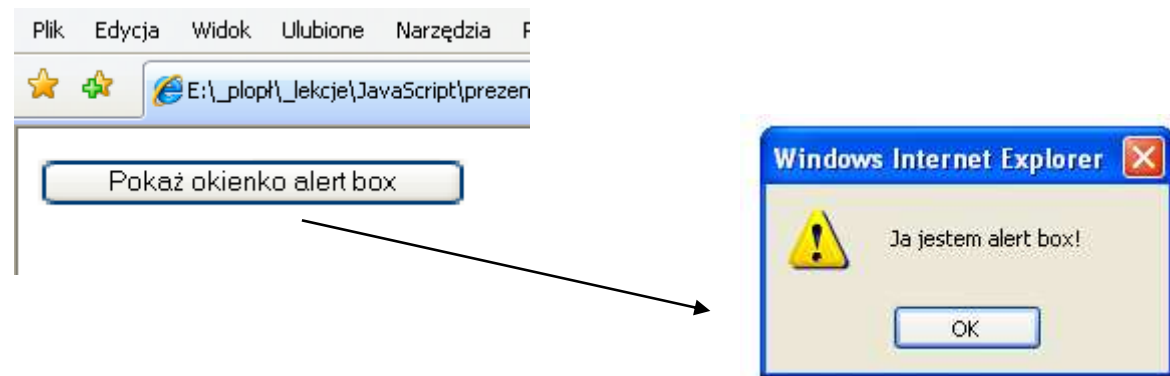
```
</head>
```

```
<body>
```

```
<input type="button" onclick="alert('Ja jestem alert box!'); "  
value="Pokaż okienko alert box" />
```

```
</body>
```

```
</html>
```

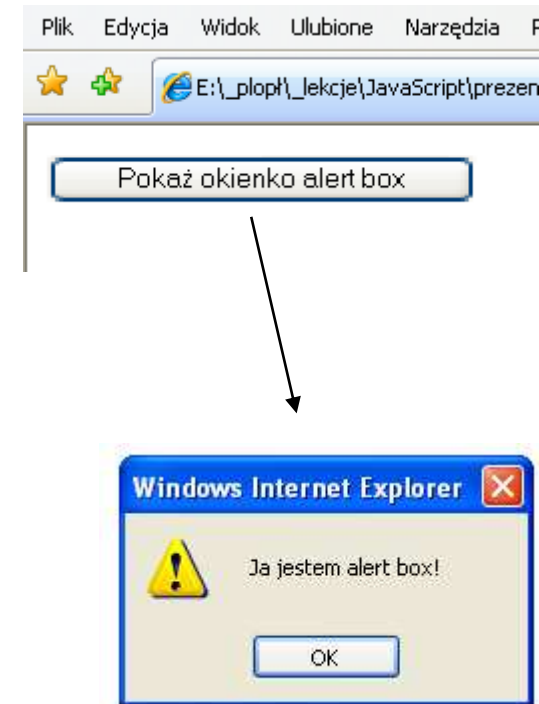


Alert box

```
<html>
<head>
<meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=ISO-8859-2">
<script type="text/javascript">
function show_alert()
{
    alert("Ja jestem alert box!");
}
</script>
</head>
<body>

<input type="button" onclick="show_alert()"
value="Pokaż okienko alert box" />

</body>
</html>
```

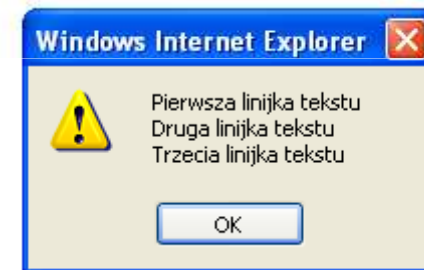
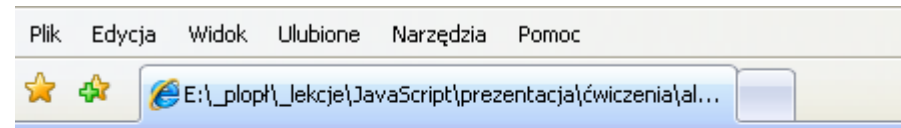


Alert box

```
<html>
<head>
<meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=ISO-8859-2">
<script type="text/javascript">
function pokaz_alert()
{
alert("Pierwsza linijka tekstu" + '\n' + "Druga linijka
tekstu" + '\n' + "Trzecia linijka tekstu");
}
</script>
</head>
<body>

<input type="button" onclick="pokaz_alert()"
value="Pokaż okienko alert box z kilkoma liniami
tekstu" />

</body>
</html>
```

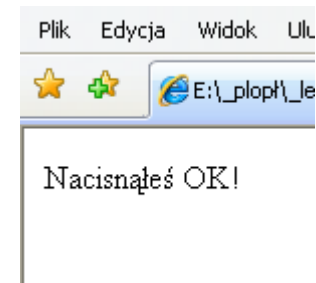
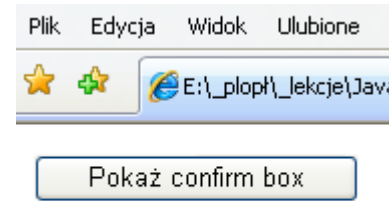


Confirm box

```
<html>
<head>
<meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=ISO-8859-2">
<script type="text/javascript">
function show_confirm()
{
var odp=confirm("Naciśnij przycisk");
if (odp==true)
{
document.write("Nacisnąłeś OK!");
}
else
{
document.write("Nacisnąłeś Anuluj!");
}
}
</script>
</head>
<body>

<input type="button" onclick="show_confirm()" value="Pokaż confirm box" />

</body>
</html>
```

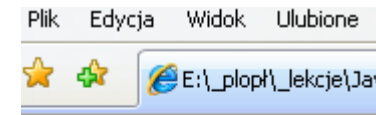


Prompt box

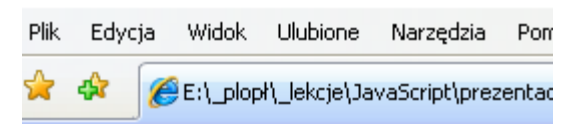
```
<html>
<head>
<meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=ISO-8859-2" />
<script type="text/javascript">
function show_prompt()
{
var imie=prompt("Wpisz swoje imię","Twoje imię");
if (imie!=null && imie!="")
{
document.write("Cześć " + imie + "! Już niedługo koniec lekcji!");
}
}
</script>
</head>
<body>

<input type="button" onclick="show_prompt()" value="Pokaż prompt box" />

</body>
</html>
```



Pokaż prompt box



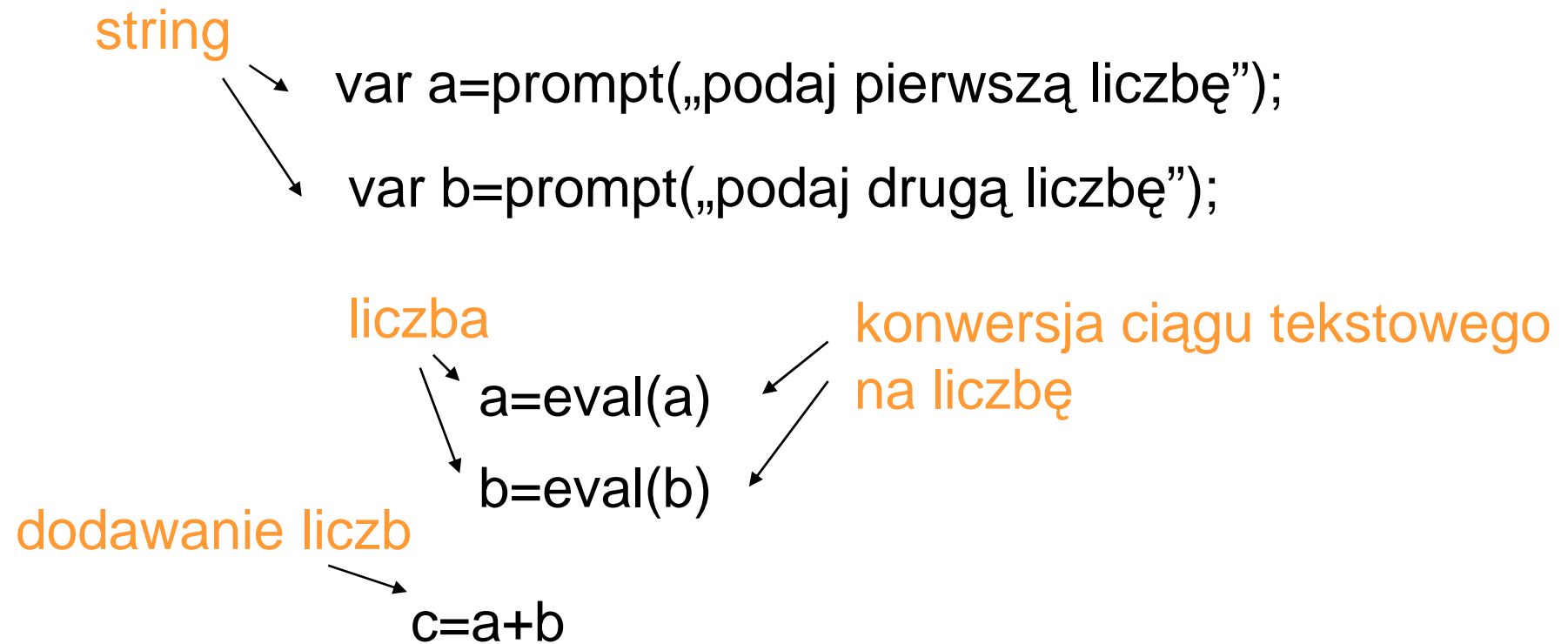
Cześć Marek! Już niedługo koniec lekcji!

Ćwiczenia

1. Napisz sumator: podajemy dwie liczby i otrzymujemy ich sumę
2. Napisz pełny kalkulator z możliwością dodawania, odejmowania, mnożenia i dzielenia dwóch pobranych od użytkownika liczb.

Prompt box

Zamiana ciągu tekstowego na liczbę



Zdarzenia

Events

Zdarzenie	Zdarzenie zachodzi kiedy...
<u>onabort</u>	anulowano pobieranie obrazka (przycisk stop)
<u>onblur</u>	element (obiekt) stracił focus (aktywność)
<u>onchange</u>	użytkownik zmienił zawartość pola formularza wprowadzania danych
<u>onclick</u>	użytkownik kliknął na obiekt – link lub element formularza
<u>ondblclick</u>	użytkownik dwa razy kliknął na obiekt
<u>onerror</u>	pojawił się błąd podczas ładowania dokumentu, obrazka
<u>onfocus</u>	element (obiekt) uzyskał focus (aktywność)
<u>onkeydown</u>	wciśnięto klawisz na klawiaturze
<u>onkeypress</u>	wciśnięto i zwolniono klawisz na klawiaturze
<u>onkeyup</u>	zwolniono klawisz na klawiaturze
<u>onload</u>	strona lub obrazek załadowały się

Zdarzenia

Events

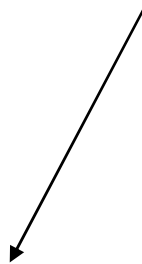
<u>onmousedown</u>	wciśnięto przycisk myszy
<u>onmousemove</u>	wskaźnik myszy przesunął się (poruszenie myszy)
<u>onmouseout</u>	wskaźnik myszy zjechał z obiektu
<u>onmouseover</u>	wskaźnik myszy najechał na obiekt
<u>onmouseup</u>	zwolniono przycisk myszy
<u>onreset</u>	naciśnięto przycisk reset
<u>onresize</u>	użytkownik (lub skrypt) zmienia rozmiar okna lub ramki
<u>onselect</u>	zaznaczono tekst w polu formularza
<u>onsubmit</u>	naciśnięto przycisk submit
<u>onunload</u>	użytkownik opuszcza stronę (wyjście ze strony, zamknięcie strony)

Komentarze

```
<script type="text/javascript">  
//komentarz pierwszy  
document.write("<h1>Oto moja strona</h1>");  
//komentarz drugi  
document.write("<p>Chodzę do Liceum PŁ</p>");  
document.write("<p>Tutaj jest super!</p>");  
/*komentarz blokowy  
document.write("<p>Chodzę do Liceum PŁ</p>");  
document.write("<p>Tutaj jest super!</p>");  
*/  
</script>
```

Ważne!

JavaScript is Case Sensitive

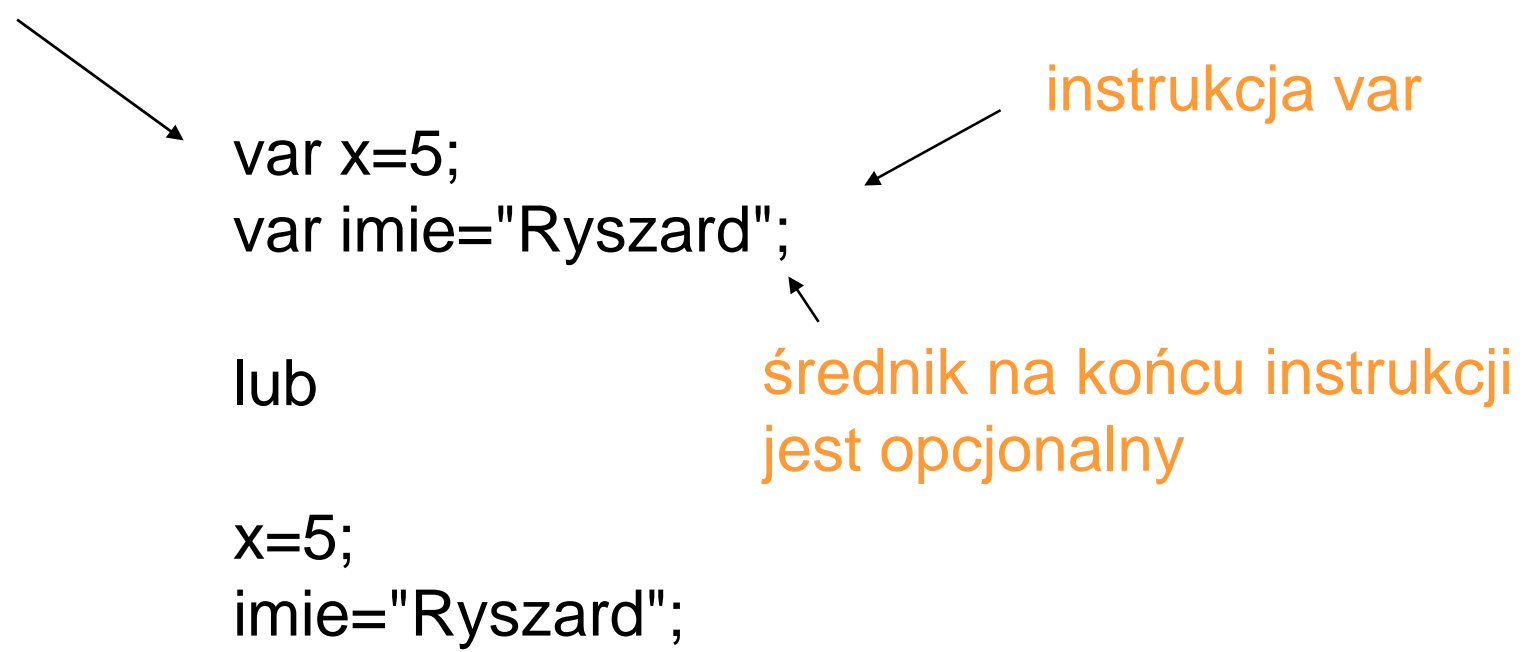


wielkość liter ma znaczenie!

zmienna *liczba* to nie to samo co zmienna *Liczba*

Deklaracja i inicjalizacja zmiennych

brak określenia typu zmiennej (variable – zmienna)



```
var x=5;  
var imie="Ryszard";
```

instrukcja var

lub

```
x=5;  
imie="Ryszard";
```

średnik na końcu instrukcji
jest opcjonalny

Uwaga! W ciele funkcji należy deklarować zmienne!

Operatory arytmetyczne

+	dodawanie	<i>addition</i>
-	odejmowanie	<i>subtraction</i>
*	mnożenie	<i>multiplication</i>
/	dzielenie	<i>division</i>
%	modulo – reszta z dzielenie liczb całkowitych	<i>modulo</i>

Operatory inkrementacji oraz dekrementacji

przykłady dla x=4;

++x	pre-inkrementacja	y=++x;	y=5; x=5;
--x	pre-dekrementacja	y=--x;	y=3; x=3;
x++	post-inkrementacja	y=x++;	y=4; x=5;
x--	post-dekrementacja	y=x--;	y=4; x=3;

```
<script type="text/javascript">  
  x=4;  
  y=x--;  
  document.write(y + " " + x);  
</script>
```

← skrypt sprawdzający
operatory

Operatory przypisania

przykłady dla $y=10$

operator	przypisanie	inny zapis tego samego	rezultat
=	$y=5$		$y=5$
+=	$y+=5$	$y=y+5$	$y=15$
-=	$y-=5$	$y=y-5$	$y=5$
/=	$y/=5$	$y=y/5$	$y=2$
=	$y=5$	$y=y*5$	$y=50$
%=	$y\%=5$	$y=y\%5$	$y=0$

Łańcuchy tekstowe

łańcuch tekstowy (string)

tekst_1="Informatyka ";

tekst_2 ="jest OK";

tekst=tekst_1 + tekst_2;

document.write(tekst);

Informatyka jest OK

stringi można dodawać

Dodawanie stringów i liczb

<code>x=5+5;</code>	10
<code>document.write(x);</code>	

<code>x="5"+"5";</code>	55
<code>document.write(x);</code>	

<code>x=5+"5";</code>	55
<code>document.write(x);</code>	

<code>x="5"+5;</code>	55
<code>document.write(x);</code>	

Jeśli dodajemy łańcuch tekstowy i liczbę
rezultatem będzie łańcuch

Try it Yourself

Operatory relacji

==	równy	<i>Equal to</i>
===	dokładnie równy (wartość i typ)	<i>Exactly equal to</i>
!=	różny	<i>Not equal to</i>
>	Większy	<i>Greater than</i>
<	Mniejszy	<i>Less than</i>
>=	większy lub równy	<i>Greater than or equal to</i>
<=	mniejszy lub równy	<i>Less than or equal to</i>

Operatory relacji

x=5

==	x==4	<i>fałsz</i>
	x==5	<i>prawda</i>
	x=="5"	<i>prawda</i>
===	x===5	<i>prawda</i>
	x=== "5"	<i>fałsz</i>
!=	x!=4	<i>prawda</i>
>	x>4	<i>prawda</i>
<	x<4	<i>fałsz</i>
>=	x>=4	<i>prawda</i>
<=	x<=4	<i>fałsz</i>

Operatory logiczne

Logical operators

&&	koniunkcja (i)	<i>and</i>
	alternatywa (lub)	<i>or</i>
!	negacja (nie)	<i>not</i>

Operatory logiczne

Logical operators

x=6 i y=3

&&	(x < 8 && y > 4)	<i>fałsz</i>
	(x==8 y==3)	<i>prawda</i>
!	!(x==y)	<i>prawda</i>

Znaki specjalne

escape sequences

znak	znaczenie	nazwa angielska	przykład
\n	nowy wiersz	<i>new-line NL (LF)</i>	
\t	tabulacja pozioma	<i>horizontal tab HT</i>	
\b	cofnięcie kursora o jeden znak	<i>backspace BS</i>	
\r	powrót karetki	<i>carriage return CR</i>	
\f	nowa strona	<i>form feed FF</i>	
\\	backslash (lewy ukośnik)	<i>backslash \</i>	
\'	apostrof	<i>single quote '</i>	
\"	cudzysłów	<i>double quote "</i>	

INSTRUKCJE STERUJĄCE

Wyrażenie

operatory arytmetyczne

operandy stałe

$a + b - 5 > 10$

operandy zmienne

operator relacji

Instrukcja warunkowa if

Wyrażenie może przyjąć jedną z dwóch wartości logicznych:
true - wyrażenie jest prawdziwe
false – wyrażenie jest fałszywe

Podobnie jak w C++
przyjmuje się, że:
0 – false
liczba różna od zera – true

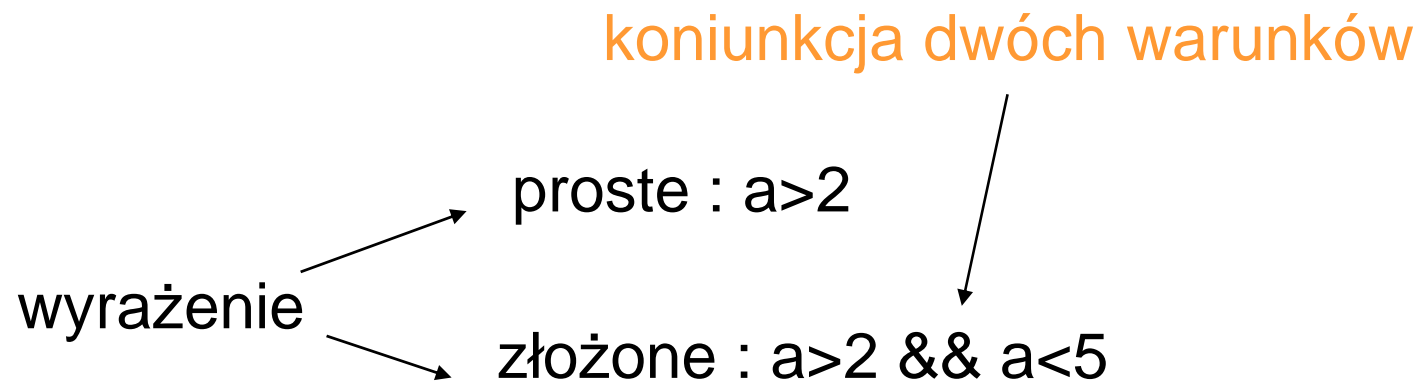


```
if (wyrażenie)  
    instrukcja1
```

Przykłady:

(a==2)	true dla zmiennej a równej 2 false dla zmiennej a równej 3
(a)	true dla zmiennej a różnej od 0 (np.: 1,2,4, 10 itp.) false dla zmiennej a równej 0
a+2*b>c	true jeśli wartość wyrażenia a+2*b jest większa od c

Instrukcja warunkowa if



Instrukcja warunkowa if

if (wyrażenie)
 instrukcja1

Pojedyncza instrukcja
nie musi mieć nawiasów
klamrowych

if (wyrażenie)
 instrukcja1
else
 instrukcja2

if (wyrażenie) {
 instrukcja1
 instrukcja2
 instrukcja3
}

blok instrukcji
(musi być ujęty
w nawiasy klamrowe)

if (wyrażenie 1)
 instrukcja1
else if (wyrażenie 2)
 instrukcja2
else if (wyrażenie 3)
 instrukcja3

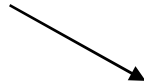
if (wyrażenie 1)
 instrukcja1
else if (wyrażenie 2)
 instrukcja2
else if (wyrażenie 3)
 instrukcja3
else
 instrukcja4

wybór wielowariantowy

Instrukcja warunkowa if

```
if (wyrażenie)  
    instrukcja1  
else  
    instrukcja2
```

inny zapis
Instrukcji
warunkowej



```
zmienna=(wyrażenie)? instrukcja1 : instrukcja2
```

Instrukcja switch

```
switch(wyrażenie)
{
    case wartość1:
        instrukcja1;
        break;
    case wartość2:
        instrukcja2;
        break;
    default:
        instrukcja3;
        break;
}
```

Pętla for

Przed pierwszym uruchomieniem pętli wykonywana jest ta instrukcja

Gdy jest tylko jedna instrukcja nawiasów może nie być

Warunek pętli – wyrażenie sprawdzane przed każdorazowym obiegiem pętli. Jeśli jest prawdziwe wykonywana jest treść pętli.

Instrukcja wykonywana każdorazowo na zakończenie obiegu pętli

```
for (i=1; i<=10; i++)
```

```
{
```

```
    document.write("Licznik pętli: ");
```

```
    document.write (i);
```

```
    document.write("<br />");
```

```
}
```

Treść pętli (instrukcje ujęte w nawiasach klamrowych)

Ćwiczenia

1. Napisz skrypt wyświetlający na stronie tabelę o zadanej liczbie kolumn i rzędów. W polu tabeli umieść słowa „informatyka”.
2. Do poprzedniego zadania dodaj następujące funkcjonalności:
 - zadana treść tabeli
 - zadana grubość tabeli
 - zadany kolor tła

Pętla for

Deklaracja i zainicjowanie licznika pętli

Warunek

Krok pętli

```
for (i=1; i<=10; i++)  
    document.write(i);
```

Ciało pętli

Gdy jest tylko
jedna instrukcja
nawiasów klamrowych
może nie być

Pętla for

Kolejność wykonywania instrukcji

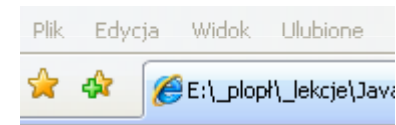
```
      1      2      4  
for (i=1; i<=10; i++)  
{  
    3  
    document.write(i);  
}
```

pierwszy obieg pętli	1234
drugi obieg pętli	234
trzeci obieg pętli	234
kolejne obiegi pętli	234

break

```
<script type="text/javascript">
for (i=0;i<=10;i++)
{
  if (i==3)
  {
    break;
  }
  document.write("i=" + i);
  document.write("<br />");
}
document.write(" wyjście z pętli przy i=" + i);
</script>
```

wyjście z pętli for przy i = 3

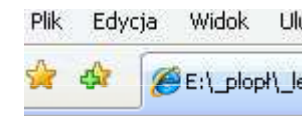


i=0
i=1
i=2
wyjście z pętli przy i=3

continue

```
<script type="text/javascript">
for (i=0;i<=10;i++)
{
  if (i==3)
  {
    continue;
  }
  document.write(„i=" + i);
  document.write("<br />");
}
</script>
```

przy i = 3 przerywa bieżącą pętlę
(pomija bieżącą kolejkę)

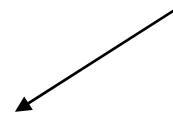


i=0
i=1
i=2
i=4
i=5
i=6
i=7
i=8
i=9
i=10

brak i=3

Tablica jednowymiarowa

deklaracja tablicy



```
var zeszyty = new Array();
```

```
zeszyty[0] = "matematyka";
```

```
zeszyty[1] = "biologia";
```

```
zeszyty[2] = "informatyka";
```



inicjalizacja tablicy

```
var zeszyty = new Array("matematyka ", "biologia ", "informatyka");
```



deklaracja z inicjalizacją

Pętla for ...in

```
<html>
```

```
<body>
```

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

```
var x;
```

```
var zeszyty = new Array();
```

```
zeszyty[0] = "matematyka";
```

```
zeszyty[1] = "biologia";
```

```
zeszyty[2] = "informatyka";
```

```
for (x in zeszyty)
```

```
{
```

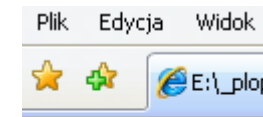
```
document.write(zeszyty[x] + "<br />");
```

```
}
```

```
</script>
```

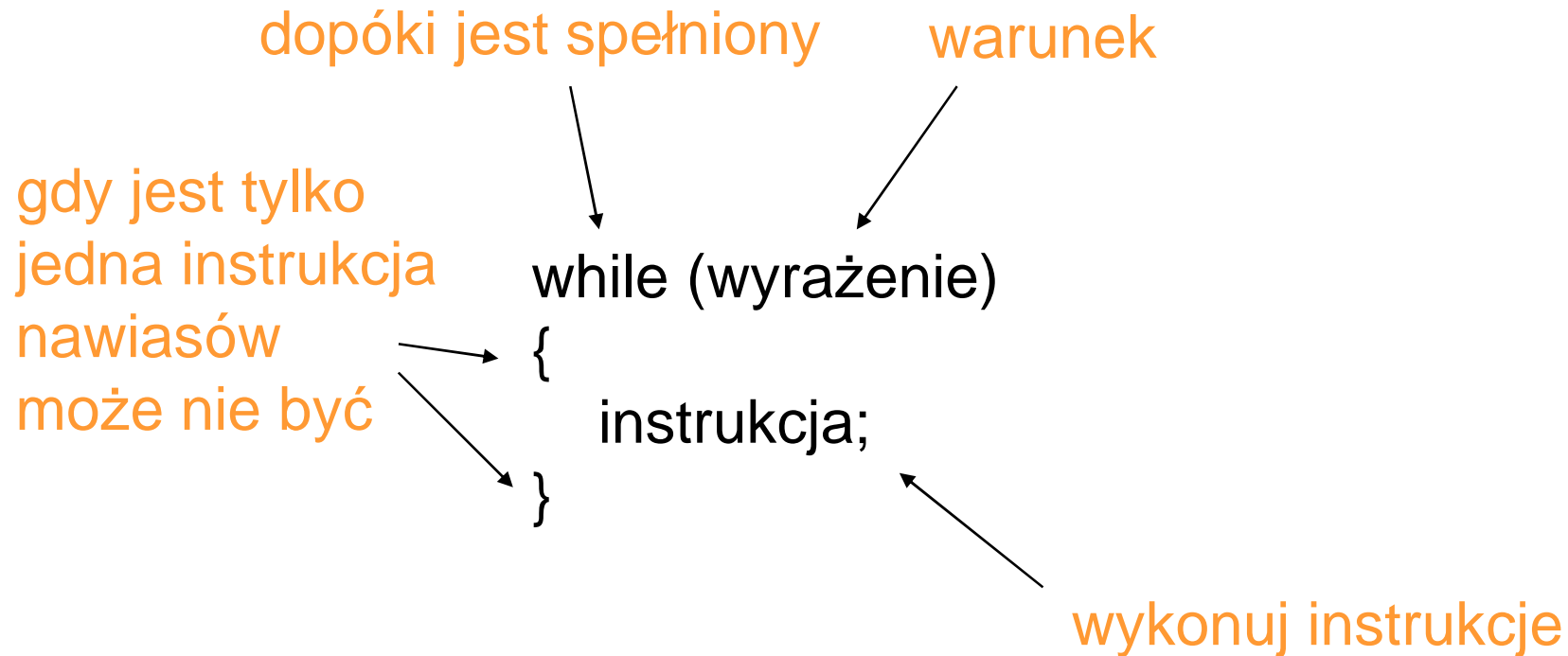
```
</body>
```

```
</html>
```



matematyka
biologia
informatyka

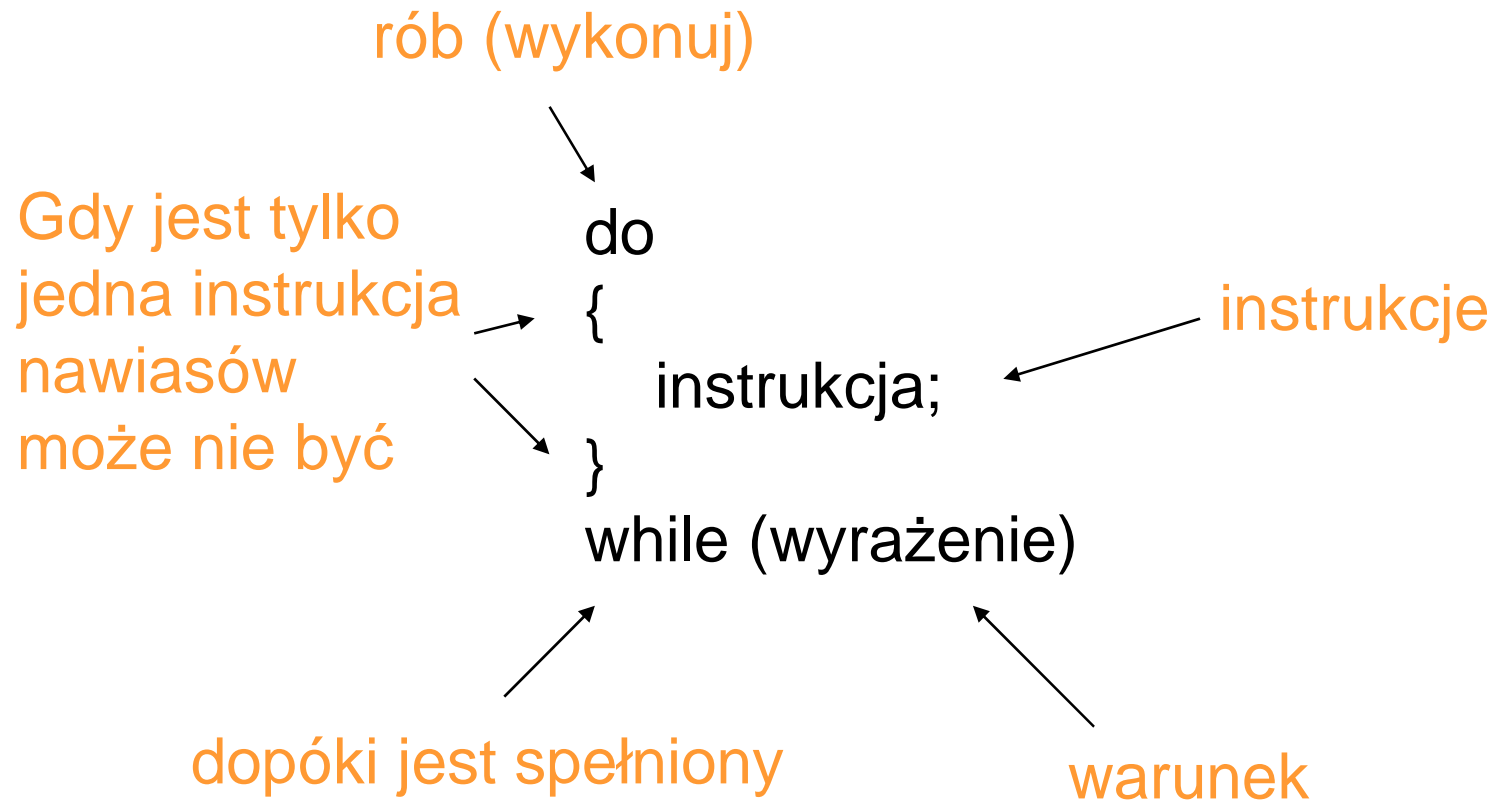
Pętla while



wyrażenie może przyjmować wartość logiczną:

- prawda
- fałsz

Pętla do while



wyrażenie może przyjmować wartość logiczną:

- prawda
- fałsz

obiektość

Object Oriented Programming (OOP) language



JavaScript



język zorientowany obiekto

obiektość

obiekt **informatyk**



Properties

właściwości obiektu:

kolor włosów: brązowy

kolor oczu: błękitny

wzrost: ok. 170cm

Methods

metody obiektu:

programowanie_w_C++()

projektowanie_stron_WWW()

fotografowanie()

Obiekt String

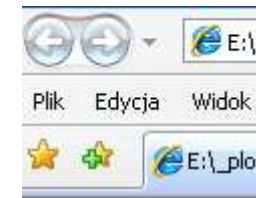
String object

operacje na łańcuchach znaków

deklaracja zmiennej *wyraz* typu string

właściwość
obiektu

```
var wyraz="Informatyka";  
document.write(wyraz.length);
```



11

wypisanie na stronie długości słowa *informatyka*

Obiekt String

String object

operacje na łańcuchach znaków

deklaracja zmiennej *wyraz* typu string

metoda
obiektu

```
var wyraz="Informatyka";  
document.write(wyraz.toUpperCase() );
```



INFORMATYKA

wypisanie na stronie słowa *INFORMATYKA*

Obiekt String

Google™

☒ Szukaj w internecie ☐ Szukaj na stronach kategorii: ję:

Sieć

Porada: [Szukaj tylko stron w języku polskim](#). Możesz określić swój język wyszukiwania:

JavaScript Reference - [[Tłumaczenie strony](#)]

This section contains a complete **reference** of all the built-in JavaScript objects, along with their methods and properties. ...

www.w3schools.com/jsref/default.asp - 26k - [Kopuj](#)

[String](#)
[Function Reference](#)
[Date Object Reference](#)
[JS Events](#)

[Array](#)
[Math Object R](#)
[RegExp Objec](#)
[JS Number](#)

[Więcej wyników z w3schools.com](#)

w3schools.com

Search W3Schools :

HOME HTML CSS XML JAVASCRIPT ASP PHP SQL MORE... References

YOUR FREE WEBSITE

Free Flash Website
Free Flash Templates
Free Web Design

★★★★★ [More reviews](#)

"A must-have for the .NET developer"
- James H.

"Easily the best WPF book available"
- C. P.

WPF Unleashed

Windows Press
Founda

JavaScript Reference

This section contains a complete reference of all the built-in JavaScript objects, along with their methods and properties.

This section also contains an event reference and a reference of the top-level properties and functions that can be used on all of the built-in JavaScript objects.

A reference to all of the HTML DOM objects are listed in the menu to the left.

JavaScript String Object

The String object is used to manipulate a stored piece of text.

For a complete reference of the properties and methods that can be used with the String object, go to our [String object reference](#).

The reference contains a brief description and examples of use for each property and method!

JS Reference

JS Overview

- JS Array
- JS Boolean
- JS Date
- JS Math
- JS Number
- JS String
- JS RegExp
- JS Functions
- JS Events

DOM Objects

- DOM Window
- DOM Navigator
- DOM Screen
- DOM History
- DOM Location
- DOM Document
- DOM Anchor
- DOM Area
- DOM Base
- DOM Body
- DOM Button
- DOM Event
- DOM Form
- DOM Frame
- DOM Frameset

Obiekt String

String object

właściwości

String Object Properties

Właściwość	opis
<u>constructor</u>	Referencja do konstruktora
<u>length</u>	Zwraca liczbę znaków w stringu
<u>prototype</u>	Pozwala dodawać właściwości i metody do obiektu

Obiekt String

String object

Metody

String Object Methods

Metoda	Opis
<u>anchor()</u>	Creates an HTML anchor
<u>big()</u>	Displays a string in a big font
<u>blink()</u>	Displays a blinking string
<u>bold()</u>	Displays a string in bold
<u>charAt()</u>	Returns the character at a specified position
<u>charCodeAt()</u>	Returns the Unicode of the character at a specified position
<u>concat()</u>	Joins two or more strings
<u>fixed()</u>	Displays a string as teletype text
<u>fontcolor()</u>	Displays a string in a specified color
<u>fontsize()</u>	Displays a string in a specified size
<u>fromCharCode()</u>	Takes the specified Unicode values and returns a string
<u>indexOf()</u>	Returns the position of the first occurrence of a specified string value in a string

Obiekt String

String object

<u>italics()</u>	Displays a string in italic
<u>lastIndexOf()</u>	Returns the position of the last occurrence of a specified string value, searching backwards from the specified position in a string
<u>link()</u>	Displays a string as a hyperlink
<u>match()</u>	Searches for a specified value in a string
<u>replace()</u>	Replaces some characters with some other characters in a string
<u>search()</u>	Searches a string for a specified value
<u>slice()</u>	Extracts a part of a string and returns the extracted part in a new string
<u>small()</u>	Displays a string in a small font
<u>split()</u>	Splits a string into an array of strings
<u>strike()</u>	Displays a string with a strikethrough
<u>sub()</u>	Displays a string as subscript

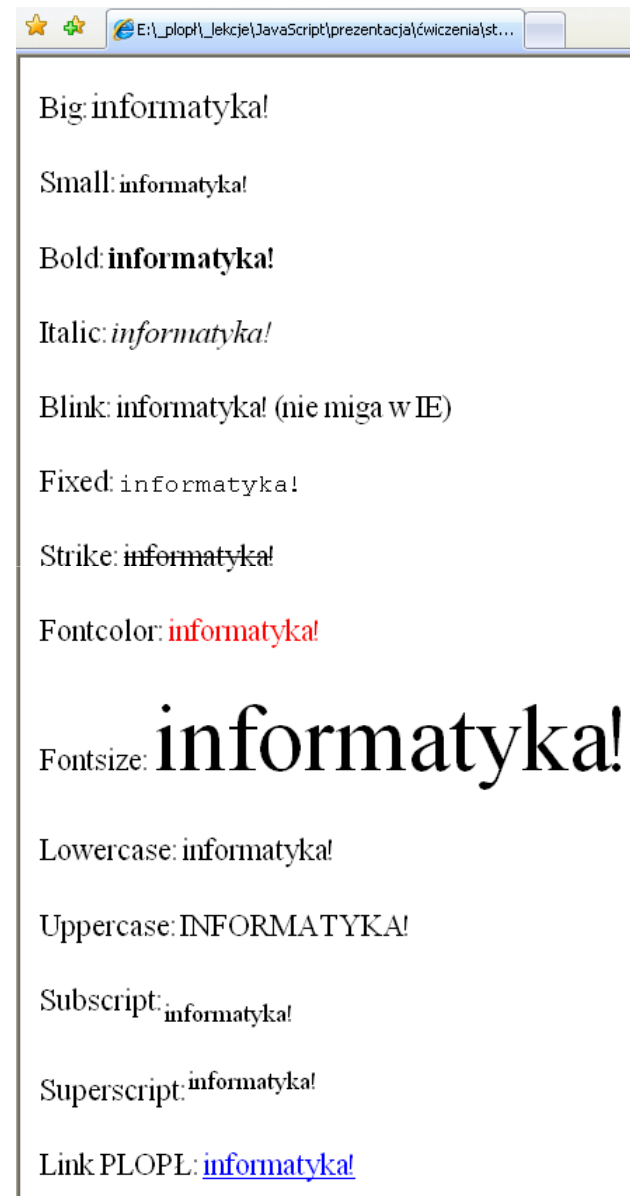
Obiekt String

String object

<u>substr()</u>	Extracts a specified number of characters in a string, from a start index
<u>substring()</u>	Extracts the characters in a string between two specified indices
<u>sup()</u>	Displays a string as superscript
<u>toLowerCase()</u>	Displays a string in lowercase letters
<u>toUpperCase()</u>	Displays a string in uppercase letters
<u>toSource()</u>	Represents the source code of an object
<u>valueOf()</u>	Returns the primitive value of a String object

Ćwiczenie

Napisać stronę
wyświetlającą w różny sposób
napis *informatyka*.



Ćwiczenia

1. Wypisz słowo *informatyka* za pomocą metody `charAt()`
2. Pobierz imię od użytkownika i wypisz je na stronie od końca.

Obiekt Math

Property	Description
<u>E</u>	Returns Euler's constant (approx. 2.718)
<u>LN2</u>	Returns the natural logarithm of 2 (approx. 0.693)
<u>LN10</u>	Returns the natural logarithm of 10 (approx. 2.302)
<u>LOG2E</u>	Returns the base-2 logarithm of E (approx. 1.442)
<u>LOG10E</u>	Returns the base-10 logarithm of E (approx. 0.434)
<u>PI</u>	Returns PI (approx. 3.14159)
<u>SQRT1_2</u>	Returns the square root of 1/2 (approx. 0.707)
<u>SQRT2</u>	Returns the square root of 2 (approx. 1.414)

Obiekt Math

Method	Description
<u>abs(x)</u>	Returns the absolute value of a number
<u>acos(x)</u>	Returns the arccosine of a number
<u>asin(x)</u>	Returns the arcsine of a number
<u>atan(x)</u>	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
<u>atan2(y,x)</u>	Returns the angle theta of an (x,y) point as a numeric value between -PI and PI radians
<u>ceil(x)</u>	Returns the value of a number rounded upwards to the nearest integer
<u>cos(x)</u>	Returns the cosine of a number
<u>exp(x)</u>	Returns the value of E^x
<u>floor(x)</u>	Returns the value of a number rounded downwards to the nearest integer
<u>log(x)</u>	Returns the natural logarithm (base E) of a number
<u>max(x,y)</u>	Returns the number with the highest value of x and y
<u>min(x,y)</u>	Returns the number with the lowest value of x and y
<u>pow(x,y)</u>	Returns the value of x to the power of y
<u>random()</u>	Returns a random number between 0 and 1
<u>round(x)</u>	Rounds a number to the nearest integer
<u>sin(x)</u>	Returns the sine of a number
<u>sqrt(x)</u>	Returns the square root of a number
<u>tan(x)</u>	Returns the tangent of an angle
<u>toSource()</u>	Represents the source code of an object
<u>valueOf()</u>	Returns the primitive value of a Math object

Ćwiczenia

pi.html 1. Wypisz na stronie liczbę π .

2. Sprawdź równanie $\sqrt{2} = 1.414213562 \dots$

abs.html 2. Co oznacza skrót *abs*? Napisz skrypt wykorzystujący metodę *abs()*.

3. Wypisz na stronie 10 liczb losowych jedna pod drugą.

random.html 4. Wypisz na stronie zadaną przez użytkownika ilość liczb losowych.

5. Wypisz na stronie 10 liczb losowych w z zakresu $\langle 0, 100 \rangle$

random1.html 6. Pobierz od użytkownika :

- liczbę liczb losowych do wypisania
- szukaną liczbę

Wypisz na stronie zadaną przez użytkownika ilość liczb losowych oraz podaj, ile razy pojawiła się zadana przez użytkownika szukana liczba.

Ćwiczenia

7. Napisz skrypt realizujący losowanie:

- jednej liczby
- dwóch liczb
- trzech liczb

w zakresie $<1,49>$

losowanie_jednej_liczby.html

losowanie_dwoch_liczb.html

losowanie_trzech_liczb.html

Obiekt Array

Property	Description
<u>constructor</u>	Returns a reference to the array function that created the object
index	
input	
<u>length</u>	Sets or returns the number of elements in an array
<u>prototype</u>	Allows you to add properties and methods to the object

Obiekt Array

Method	Description
<u>concat()</u>	Joins two or more arrays and returns the result
<u>join()</u>	Puts all the elements of an array into a string. The elements are separated by a specified delimiter
<u>pop()</u>	Removes and returns the last element of an array
<u>push()</u>	Adds one or more elements to the end of an array and returns the new length
<u>reverse()</u>	Reverses the order of the elements in an array
<u>shift()</u>	Removes and returns the first element of an array
<u>slice()</u>	Returns selected elements from an existing array
<u>sort()</u>	Sorts the elements of an array
<u>splice()</u>	Removes and adds new elements to an array
<u>toSource()</u>	Represents the source code of an object
<u>toString()</u>	Converts an array to a string and returns the result
<u>unshift()</u>	Adds one or more elements to the beginning of an array and returns the new length
<u>valueOf()</u>	Returns the primitive value of an Array object

Array - metoda pop()

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

```
var imiona = new Array(3);
```

```
imiona[0] = „Ania”;
```

```
imiona[1] = „Waldek”;
```

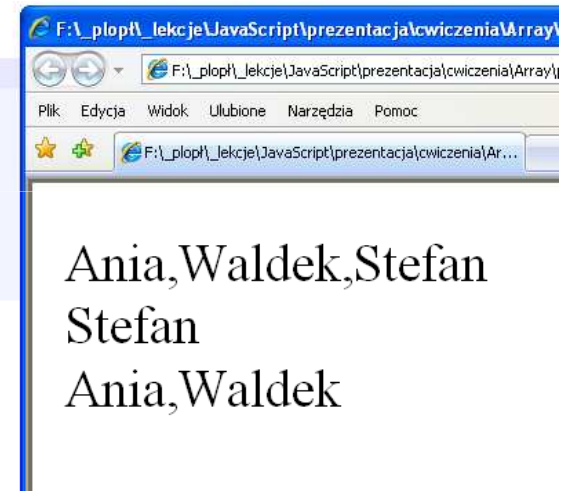
```
imiona[2] = „Stefan”;
```

```
document.write(imiona + "<br />");
```

```
document.write(imiona.pop() + "<br />");
```

```
document.write(imiona);
```

```
</script>
```

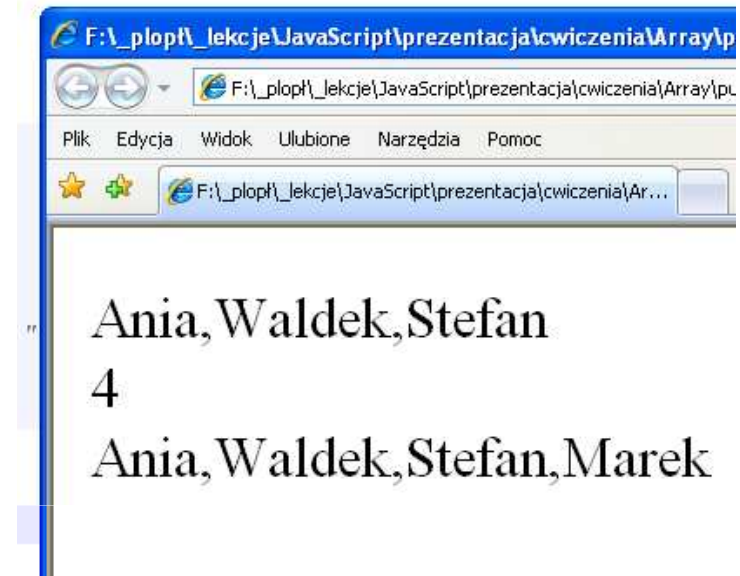


Array - metoda push()

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

```
var imiona = new Array(3);  
imiona[0] = "Ania";  
imiona[1] = "Waldek";  
imiona[2] = "Stefan";  
document.write(imiona + "<br />");  
document.write(imiona.push("Marek") + "<br />");  
document.write(imiona);
```

```
</script>
```

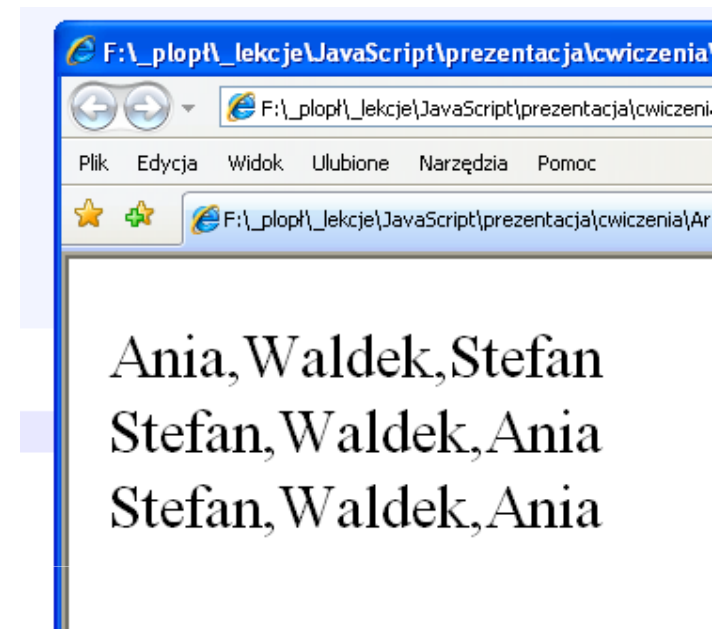


Array - metoda reverse()

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

```
var imiona = new Array(3);  
imiona[0] = "Ania";  
imiona[1] = "Waldek";  
imiona[2] = "Stefan";  
document.write(imiona + "<br />");  
document.write(imiona.reverse() + "<br />");  
document.write(imiona);
```

```
</script>
```



Obiekt Date

Property	Description
<u>constructor</u>	Returns a reference to the Date function that created the object
<u>prototype</u>	Allows you to add properties and methods to the object

Obiekt Date

Method	Description
Date()	Returns today's date and time
getDate()	Returns the day of the month from a Date object (from 1-31)
getDay()	Returns the day of the week from a Date object (from 0-6)
getFullYear()	Returns the year, as a four-digit number, from a Date object
getHours()	Returns the hour of a Date object (from 0-23)
getMilliseconds()	Returns the milliseconds of a Date object (from 0-999)
getMinutes()	Returns the minutes of a Date object (from 0-59)
getMonth()	Returns the month from a Date object (from 0-11)
getSeconds()	Returns the seconds of a Date object (from 0-59)
getTime()	Returns the number of milliseconds since midnight Jan 1, 1970
getTimezoneOffset()	Returns the difference in minutes between local time and Greenwich Mean Time (GMT)
getUTCDate()	Returns the day of the month from a Date object according to universal time (from 1-31)
getUTCDay()	Returns the day of the week from a Date object according to universal time (from 0-6)
getUTCMonth()	Returns the month from a Date object according to universal time (from 0-11)
getUTCFullYear()	Returns the four-digit year from a Date object according to universal time
getUTCHours()	Returns the hour of a Date object according to universal time (from 0-23)
getUTCMinutes()	Returns the minutes of a Date object according to universal time (from 0-59)
getUTCSeconds()	Returns the seconds of a Date object according to universal time (from 0-59)
getUTCMilliseconds()	Returns the milliseconds of a Date object according to universal time (from 0-999)
getYear()	Returns the year, as a two-digit or a three/four-digit number, depending on the browser. Use <code>getFullYear()</code> instead !!

Obiekt Date

<u>parse()</u>	Takes a date string and returns the number of milliseconds since midnight of January 1, 1970
<u>setDate()</u>	Sets the day of the month in a Date object (from 1-31)
<u>setFullYear()</u>	Sets the year in a Date object (four digits)
<u>setHours()</u>	Sets the hour in a Date object (from 0-23)
<u>setMilliseconds()</u>	Sets the milliseconds in a Date object (from 0-999)
<u>setMinutes()</u>	Set the minutes in a Date object (from 0-59)
<u>setMonth()</u>	Sets the month in a Date object (from 0-11)
<u>setSeconds()</u>	Sets the seconds in a Date object (from 0-59)
<u>setTime()</u>	Calculates a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970
<u>setUTCDate()</u>	Sets the day of the month in a Date object according to universal time (from 1-31)
<u>setUTCMonth()</u>	Sets the month in a Date object according to universal time (from 0-11)
<u>setUTCFullYear()</u>	Sets the year in a Date object according to universal time (four digits)
<u>setUTCHours()</u>	Sets the hour in a Date object according to universal time (from 0-23)
<u>setUTCMinutes()</u>	Set the minutes in a Date object according to universal time (from 0-59)
<u>setUTCSeconds()</u>	Set the seconds in a Date object according to universal time (from 0-59)
<u>setUTCMilliseconds()</u>	Sets the milliseconds in a Date object according to universal time (from 0-999)
<u>setYear()</u>	Sets the year in the Date object (two or four digits). Use setFullYear() instead !!
<u>toDate()</u>	Returns the date portion of a Date object in readable form
<u>toGMTString()</u>	Converts a Date object, according to Greenwich time, to a string. Use toUTCString() instead !!
<u>toLocaleDateString()</u>	Converts a Date object, according to local time, to a string and returns the date portion
<u>toLocaleTimeString()</u>	Converts a Date object, according to local time, to a string and returns the time portion

Obiekt Date

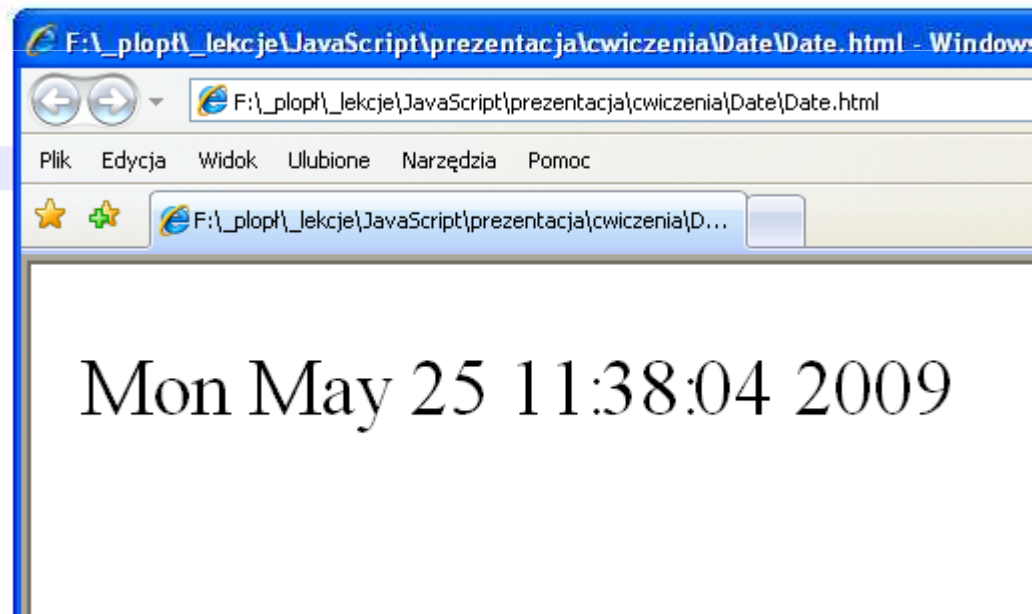
<u>toLocaleString()</u>	Converts a Date object, according to local time, to a string
<u>toSource()</u>	Represents the source code of an object
<u>toString()</u>	Converts a Date object to a string
<u>getTimeString()</u>	Returns the time portion of a Date object in readable form
<u>toUTCString()</u>	Converts a Date object, according to universal time, to a string
<u>UTC()</u>	Takes a date and returns the number of milliseconds since midnight of January 1, 1970 according to universal time
<u>valueOf()</u>	Returns the primitive value of a Date object

Date – metoda Date()

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

```
document.write(Date());
```

```
</script>
```

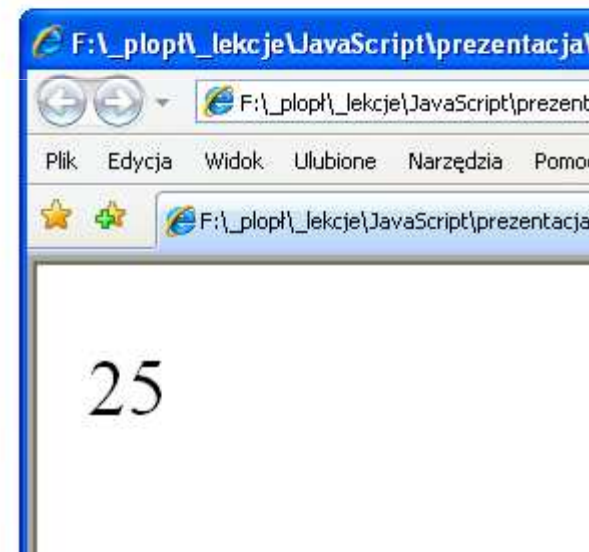


Date – metoda getDate()

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

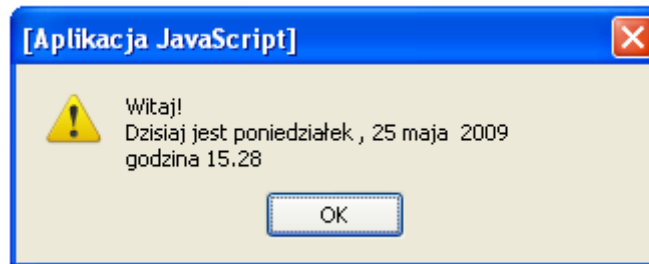
```
var d = new Date();  
document.write(d.getDate());
```

```
</script>
```



Ćwiczenia

1. Napisz skrypt wyświetlający bieżącą datę w formacie:



polska_data.html

DOM

Document Object Model



DOM



Standard organizacji W3C

DOM HTML

definiuje właściwości i metody wszystkich obiektów HTML na stronie



DOM HTML

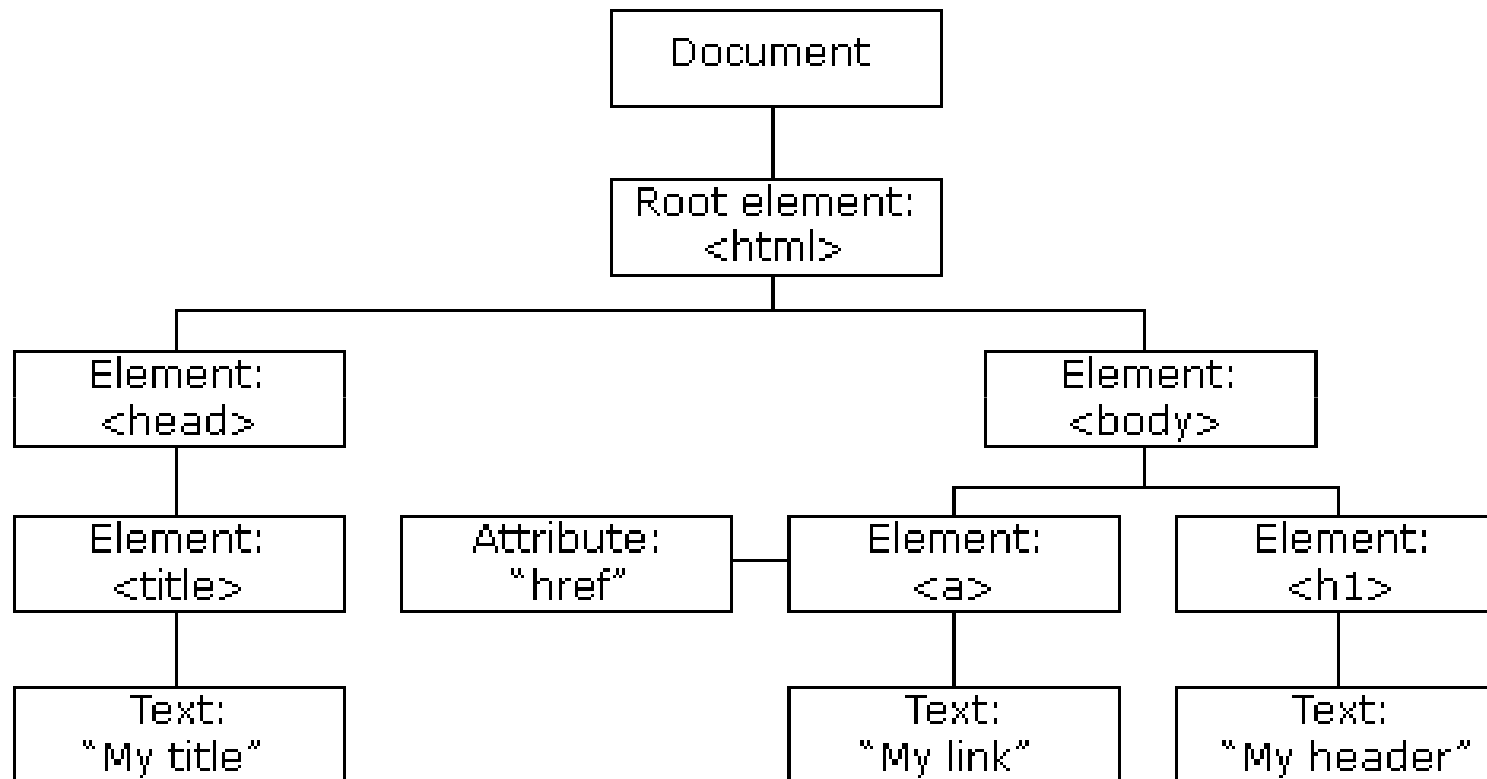


definiuje jak dostać się do elementu HTML na stronie od strony programistycznej

Drzewo dokumentu

The HTML DOM Node Tree

węzeł



Dostęp do obiektów strony

```
<p id="tekst">Informatyka jest OK</p>
```

getElementById.html




JavaScript is Case Sensitive !

```
<script type="text/javascript">  
    t=document.getElementById("tekst");  
    document.write("Tekst w akapicie " + t.innerHTML);  
</script>
```



The easiest way to get or modify the content of an element is by using the innerHTML property.
innerHTML is not a part of the W3C DOM specification. However, it is supported by all major browsers.
The innerHTML property is useful for returning or replacing the content of HTML elements (including <html> and <body>), it can also be used to view the source of a page that has been dynamically modified.

Walidacja danych

Plik	Edycja	Widok	Ulubione	Narzędzia	Pomoc
 		 walidacja danych			

Twoje dane:

imię: *

nazwisko: *

klasa:

Uwaga!

Pola oznaczone * są obowiązkowe!

Walidacja danych - formularz

walidacja.html

```
<form name="formularz" action="http://217.113.238.59/~bulkas/formularze/walidacja/walidacja.php"
method="post">
```

Twoje dane:

```
<hr>
```

```
<table>
```

```
  <tr>
```

```
    <td>imię:</td>
```

```
    <td><input type="text" name="imie" id="imie">*</td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>nazwisko:</td>
```

```
    <td><input type="text" name="nazwisko" id="nazwisko">*</td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>klasa:</td>
```

```
    <td><input type="text" name="klasa"></td>
```

```
  </tr>
```

```
</table>
```

```
<input type="button" name="wyslij" value="wyślij" onClick="walidacja_danych()">
```

```
<hr>
```

Uwaga!

Pola oznaczone * są obowiązkowe!

```
</form>
```

Walidacja danych - skrypt

```
<script type="text/javascript">
function walidacja_danych (){

    //flaga prawidłowych danych
    var wpis=1;

    //zmienna wypisująca pola, w które nic nie wpisano
    var braki="";

    //Jeśli nie wpisano nic do pola "imie"
    if (document.getElementById("imie").value == ""){

        //wpisujemy pole "imie" do braków
        braki += "imie\n"

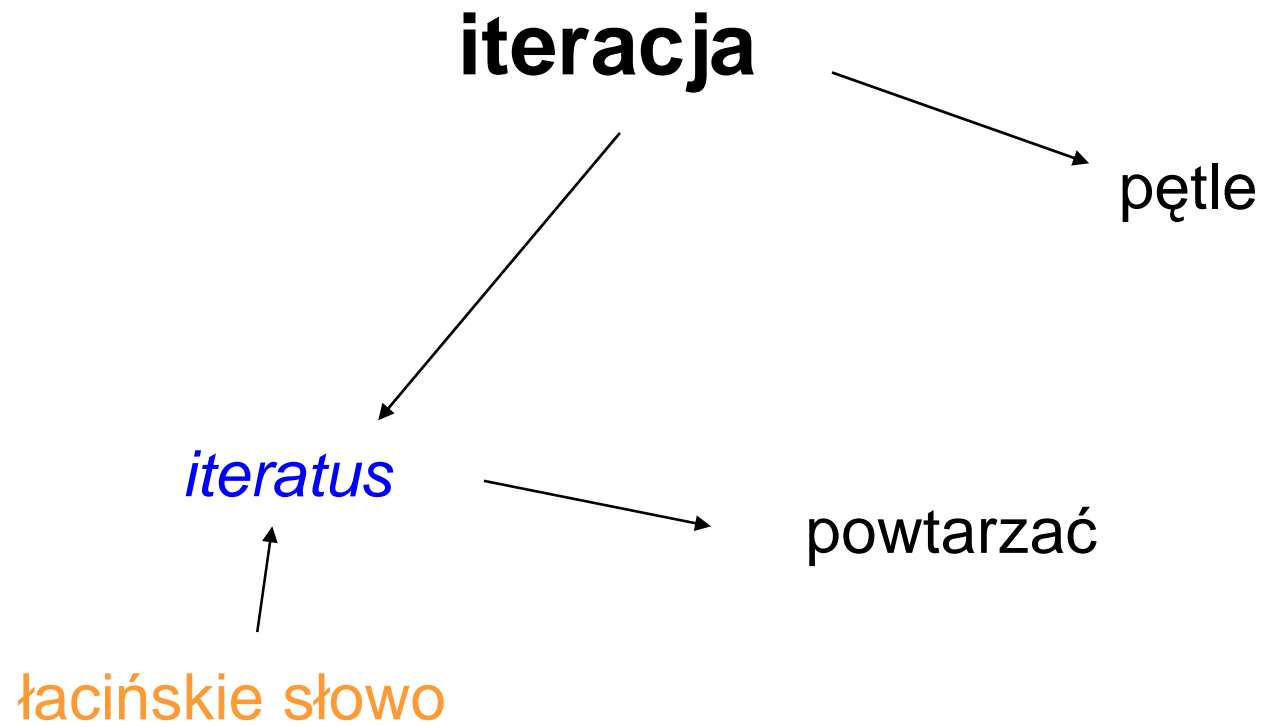
        //zerujemy flagę prawidłowych danych
        wpis=0;
    }
    if (document.getElementById("nazwisko").value == ""){
        braki += "nazwisko\n"
        wpis=0;
    }
    }

    //jeśli flaga prawidłowych danych jest prawdą (czyli równa 1)
    if (wpis)
        //wysyłamy formularz
        formularz.submit();
    //jeśli flaga prawidłowych danych jest fałszem (czyli równa 0)
    else
        //wyświetlamy stosowny alert
        alert ("Nie wypełniłeś obowiązkowych pól:\n" + braki);
    }
</script>
```

REKURENCJA

ITERACJA

Iteracja



Zadanie

Wartość silni określa wzór znany z lekcji matematyki:

$$n! = 1 * 2 * 3 * 4 * (n - 1) * n$$

Napisz funkcję obliczającą silnię zadanej liczby metodą iteracyjną.

Silnia iteracyjnie

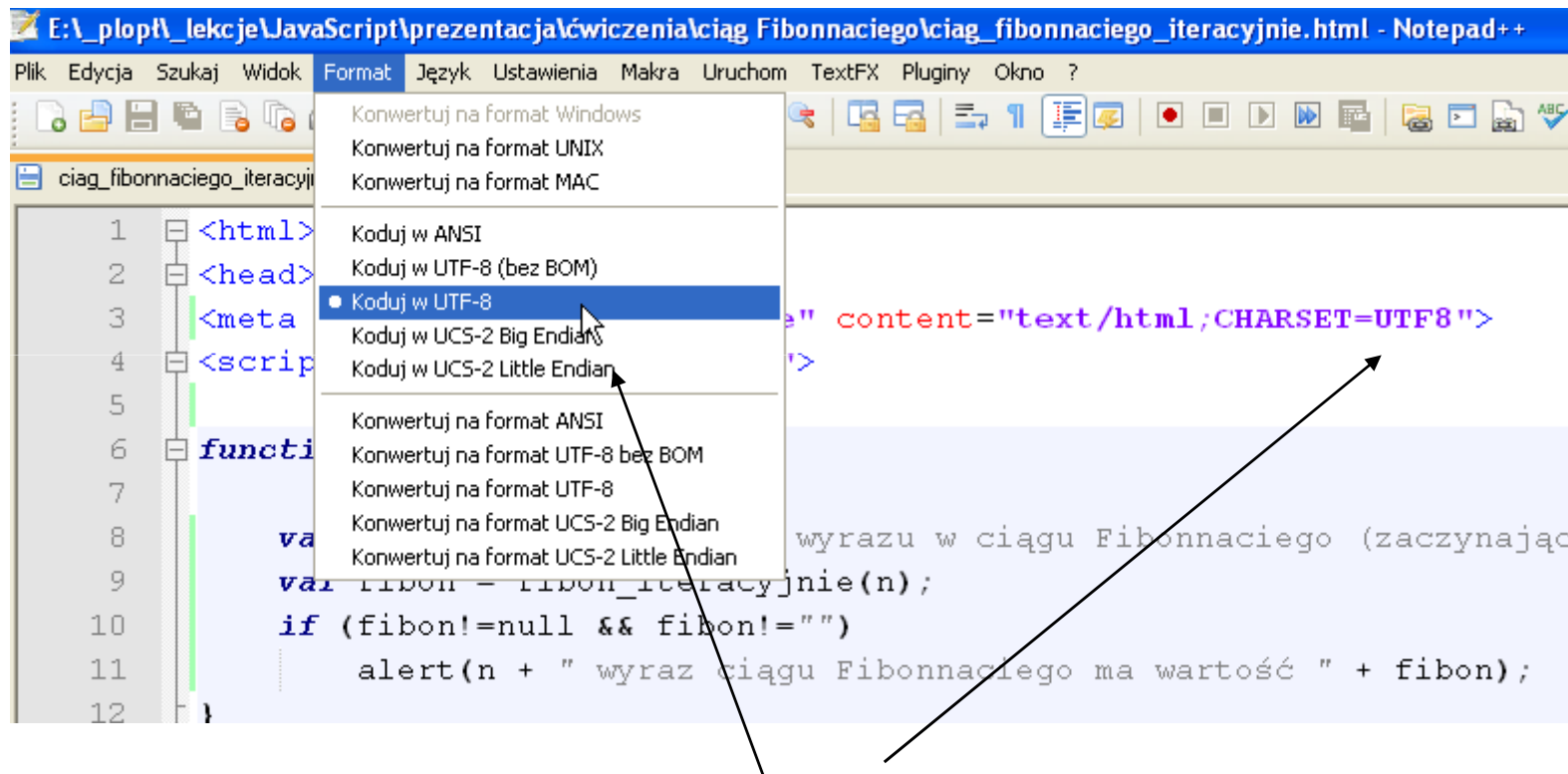
$$n! = 1 * 2 * 3 * 4 * (n - 1) * n$$

funkcja obliczająca iteracyjnie wartość silni



```
function silnia_iteracyjnie (n){  
  
    var silnia=1;  
    for (i=1; i<=n; i++)  
        silnia=silnia * i;  
    return silnia;  
  
}
```

Polskie znaki w Notepad++



gwarancja polskich znaków na stronie www

Silnia iteracyjnie

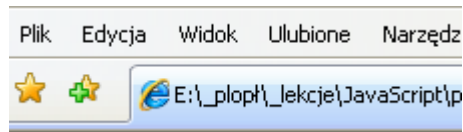
```
1 <html>
2 <head>
3   <meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=UTF8">
4
5   <script type="text/javascript">
6
7
8   function licz(){
9
10      var n=prompt("podaj liczbę całkowitą większą od zera","");
11      var silnia = silnia_iteracyjnie(n);
12      if (silnia!=null && silnia!="")
13         alert("silnia liczby " + n + " wynosi " + silnia);
14   }
15
16   function silnia_iteracyjnie(n){
17
18
19      var silnia=1;
20      for (i=1; i<=n; i++)
21         silnia=silnia * i;
22      return silnia;
23   }
24
25 </script>
26 </head>
27
28 <body>
29   <input type="button" onclick="licz()" value="silnia liczona iteracyjnie"><br />
30 </body>
31 </html>
```

wywołanie funkcji
silnia_iteracyjnie(n)

po naciśnięciu przycisku
wywoływana jest funkcja licz()

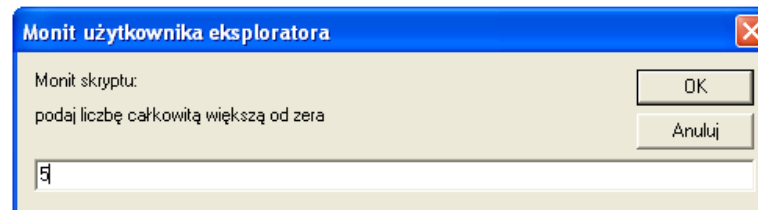
przycisk

Silnia iteracyjnie



silnia liczona iteracyjnie

przycisk

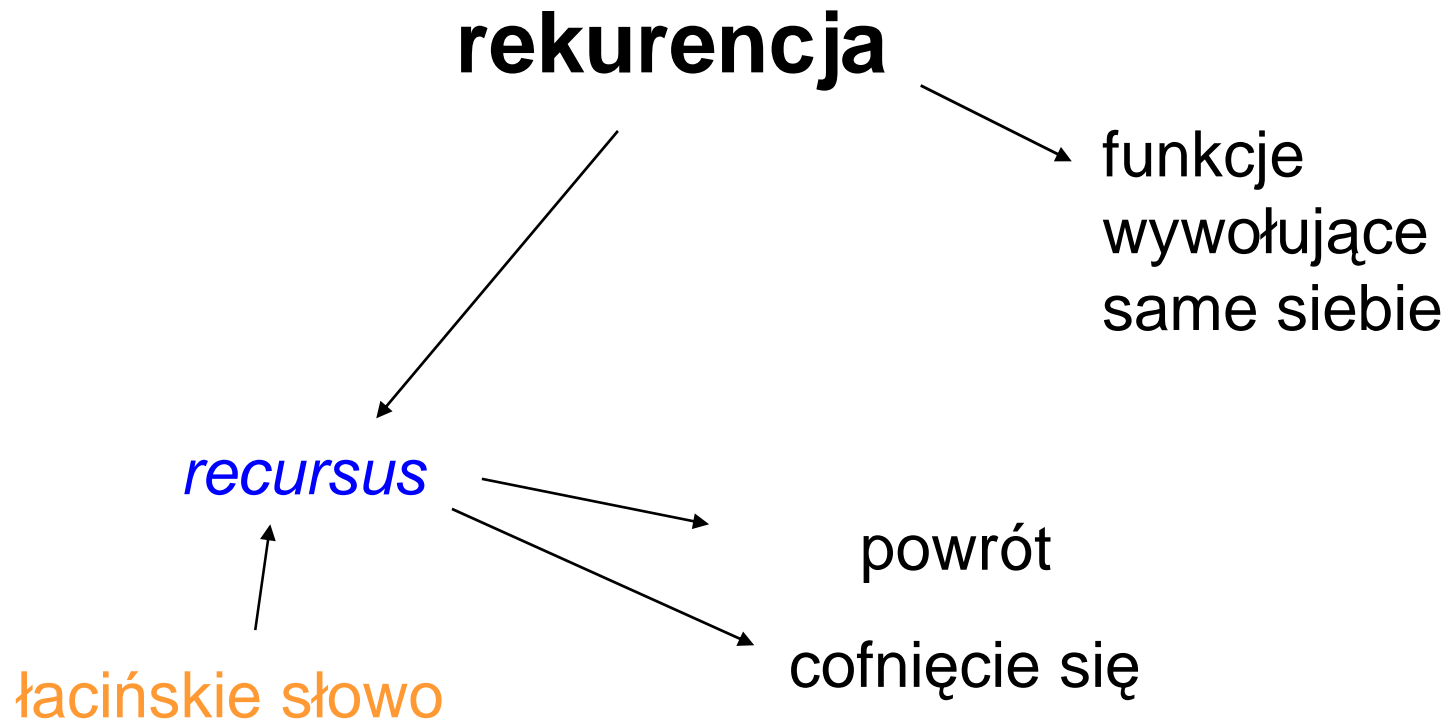


okienko prompt



okienko alert

Rekurencja



Zadanie

Napisz funkcję obliczającą silnię zadanej liczby za pomocą rekurencji.

Silnia rekurencyjnie

$$3! = 1 * 2 * 3$$

$$4! = 1 * 2 * 3 * 4$$



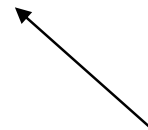
3!

$$4! = 3! * 4$$

$$n! = (n-1)! * n$$



silnia_rekurencyjnie (n)



silnia_rekurencyjnie (n - 1)

Silnia rekurencyjnie

n należy do liczb naturalnych

$$\text{silnia_rekurencyjnie}(n) = \begin{cases} 1, & \text{gdy } n=1, \\ \text{silnia_rekurencyjnie}(n-1) * n, & \text{gdy } n>1 \end{cases}$$

Uwaga!

Przyjąłem, że zero nie należy do liczb naturalnych

W matematyce przyjmuje się $0!=1$

Silnia rekurencyjnie

funkcja obliczająca rekurencyjnie wartość silni



```
function silnia_rekurencyjnie (n){  
    if (n==1)  
        return 1;  
    else  
        return silnia_rekurencyjnie(n-1) * n  
}
```

Silnia rekurencyjnie

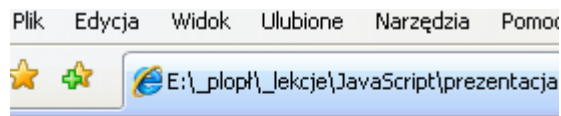
```
1 <html>
2 <head>
3   <meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=UTF8">
4
5   <script type="text/javascript">
6
7
8   function licz(){
9
10      var n=prompt("podaj liczbę całkowitą większą od zera","");
11      var silnia = silnia_rekurencyjnie(n);
12      if (silnia!=null && silnia!="")
13         alert("silnia liczby " + n + " wynosi " + silnia);
14   }
15
16   function silnia_rekurencyjnie (n){
17
18      if (n==1)
19         return 1;
20      else
21         return silnia_rekurencyjnie(n-1) * n
22
23   }
24
25
26 </script>
27 </head>
28 <body>
29   <input type="button" onclick="licz()" value="silnia liczona rekurencyjnie"><br />
30 </body>
31 </html>
```

wywołanie funkcji
silnia_rekurencyjnie(n)

przycisk

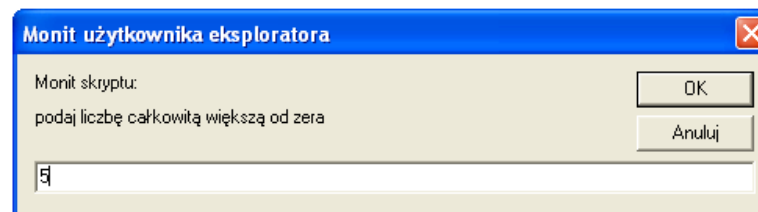
po naciśnięciu przycisku
wywoływana jest funkcja licz()

Silnia rekurencyjnie



silnia liczona rekurencyjnie

przycisk



okienko prompt



okienko alert

Ciąg Fibonacciego

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987...



niektórzy matematycy
zaczynają od zera

Czy możesz napisać wzór na ciąg?

Ciąg Fibonacciego

numer wyrazu w ciągu



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987...

pierwszy wyraz ciągu : 1

drugi wyraz ciągu : 1

następny wyraz ciągu : suma dwóch poprzednich

Zadanie

Napisz funkcję obliczającą zadany wyraz ciągu Fibonacciego iteracyjnie.

Ciąg Fibonacciego iteracyjnie

```
function fibon_iteracyjnie (n){
```

```
    if (n <= 2)
```

```
        return 1;
```

```
    else {
```

```
        f1=1;
```

```
        f2=1;
```

```
        for (i=3; i<=n; i++){
```

```
            f=f1 + f2;
```

```
            f1=f2;
```

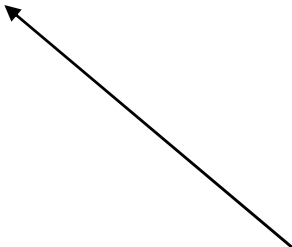
```
            f2=f;
```

```
        }
```

```
        return f;
```

```
    }
```

```
}
```



funkcja obliczająca
iteracyjnie wartość
n-tego wyrazu ciągu

Ciąg Fibonacciego iteracyjnie

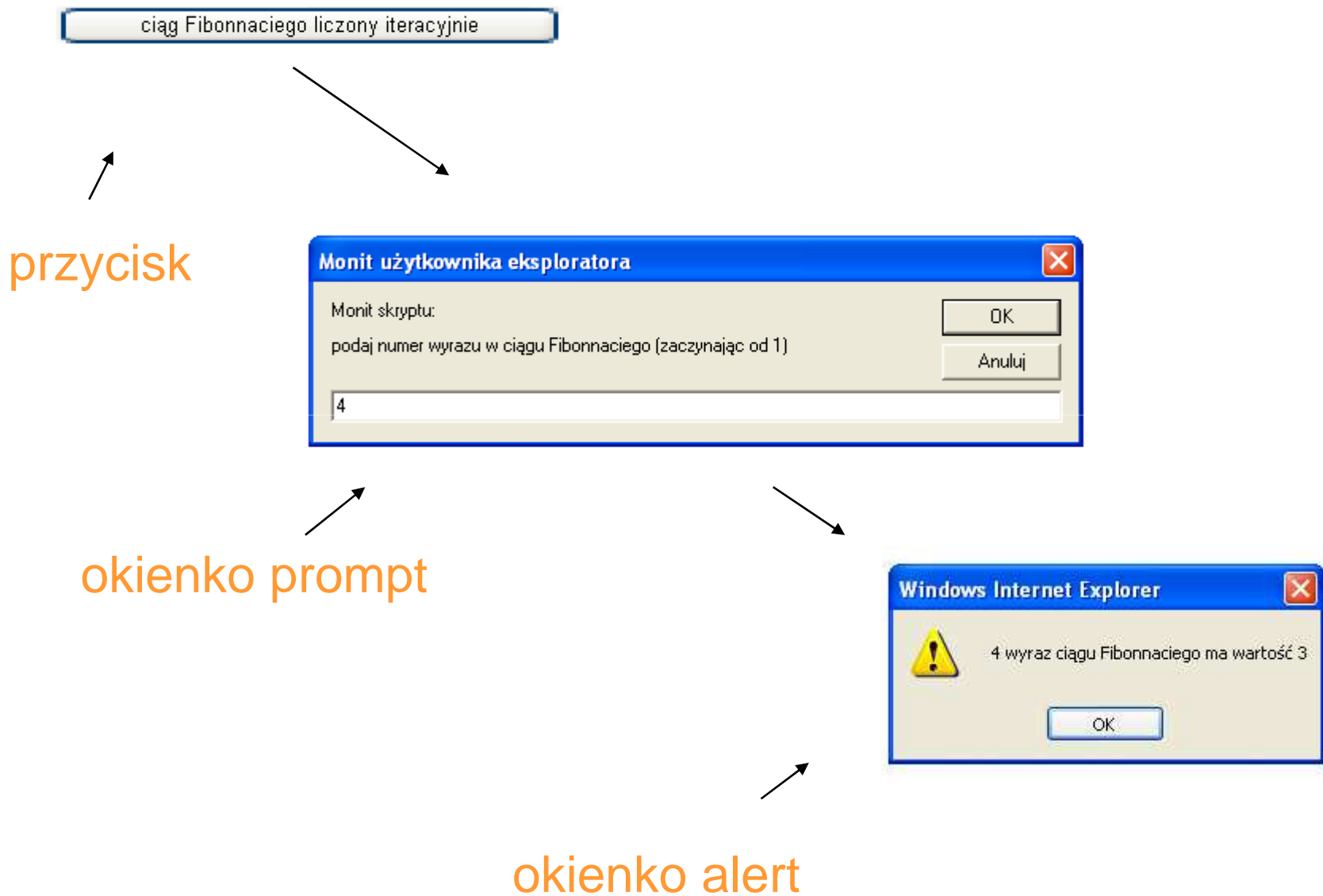
```
1 <html>
2 <head>
3   <meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=UTF8">
4   <script type="text/javascript">
5
6   function licz(){
7
8       var n=prompt("podaj numer wyrazu w ciągu Fibonnaciego (zaczynając od 1)","");
9       var fibon = fibon_iteracyjnie(n);
10      if (fibon!=null && fibon!="")
11          alert(n + " wyraz ciągu Fibonnaciego ma wartość " + fibon);
12  }
13
14  function fibon_iteracyjnie (n){
15
16      if (n <= 2)
17          return 1;
18      else {
19          f1=1;
20          f2=1;
21          for (i=3; i<=n; i++){
22              f=f1 + f2;
23              f1=f2;
24              f2=f;
25          }
26          return f;
27      }
28  }
29
30
31
32 </script>
33 </head>
34 <body>
35   <input type="button" onclick="licz()" value="ciąg Fibonnaciego liczony iteracyjnie"><br />
36 </body>
37 </html>
```

wywołanie funkcji
fibon_iteracyjnie(n)

przycisk

po naciśnięciu przycisku
wywoływana jest funkcja licz()

Ciąg Fibonacciego iteracyjnie

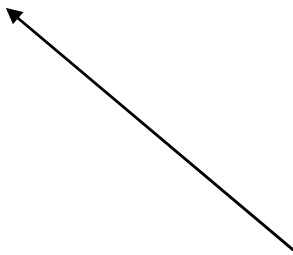


Zadanie

Napisz funkcję obliczającą zadany wyraz ciągu Fibonacciego za pomocą rekurencji.

Ciąg Fibonacciego rekurencyjnie

```
function fibon_rekurencyjnie(n){  
    if (n<=2)  
        return 1;  
    else  
        return fibon_rekurencyjnie(n-2) + fibon_rekurencyjnie(n-1)  
}
```



funkcja obliczająca
rekurencyjnie wartość
n-tego wyrazu ciągu

Ciąg Fibonacciego rekurencyjnie

```
ciag_fibonnaciego_rekurencyjnie.html
1 <html>
2 <head>
3   <meta HTTP-EQUIV="Content-Type" content="text/html;CHARSET=UTF8">
4   <script type="text/javascript">
5
6   function licz(){
7
8       var n=prompt("podaj numer wyrazu w ciągu Fibonnaciego (zaczynając od 1)","");
9       var fibon = fibon_rekurencyjnie(n);
10      if (fibon!=null && fibon!="")
11          alert(n + " wyraz ciągu Fibonnaciego ma wartość " + fibon);
12  }
13
14  function fibon_rekurencyjnie(n){
15
16      if (n<=2)
17          return 1;
18      else
19          return fibon_rekurencyjnie(n-2) + fibon_rekurencyjnie(n-1)
20
21  }
22
23
24  </script>
25  </head>
26  <body>
27    <input type="button" onclick="licz()" value="ciąg Fibonnaciego liczony rekurencyjnie"><br />
28  </body>
29  </html>
```

wywołanie funkcji
fibon_rekurencyjnie(n)

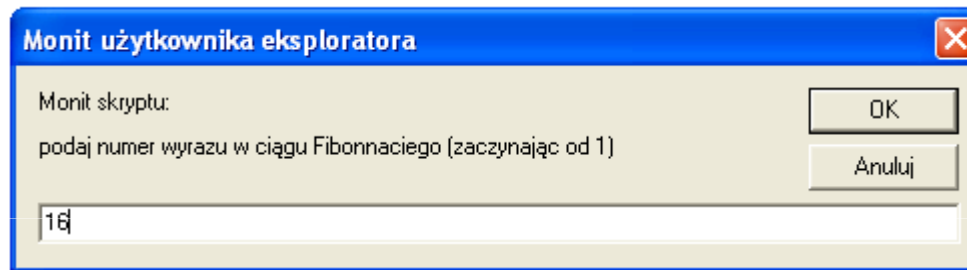
przycisk

po naciśnięciu przycisku
wywoływana jest funkcja licz()

Ciąg Fibonacciego rekurencyjnie

ciąg Fibonnaciego liczony rekurencyjnie

przycisk



Monit użytkownika eksploratora

Monit skryptu:
podaj numer wyrazu w ciągu Fibonnaciego (zaczynając od 1)

16

OK
Anuluj

okienko prompt



okienko alert