Technologie internetowe

CZĘŚĆ 7

JavaScript

Wykład opracowany na podstawie materiałów z:
PHP i MySQL. Tworzenie stron internetowych. Helion
http://pl.wikipedia.org/
Materiały wykładowe - dr Tomasz Bajorek

Dr inż. Arkadiusz Rzucidło

Co to jest JavaScript ?

- Język programowania:
  - interpretowany
  - zorientowany obiektowo
  - Skryptowy
- JavaScript jest oddzielnym językiem, nie jest uproszczoną wersją języka Java
- Osadzany w postaci skryptów w innych strukturach, na przykład w plikach HTML
- Interpreter zaimplementowany we wszystkich przeglądarkach
- Ma dostęp do obiektów otoczenia (np. modelu obiektowego przeglądarki internetowej – DOM – Dynamic Object Model), można sprawować nad tym otoczeniem kontrolę

Porównanie JavaScript i Java

JavaScript	Java
Interpretowany przez klienta	Wykonywany przez klienta
Tworzy skrypty, które mogą być bezpośrednio wstawiane do stron HTML	Tworzy "aplety", które są wywoływane ze strony HTML
Łatwy	Wymaga doświadczenia programistycznego

Kod JavaScript w HTMLu

- Skrypty JavaScript są zagnieżdżane w dokumentach HTML. Skrypt JavaScript umieszczane są między znacznikami <SCRIPT> i </SCRIPT>

```
<SCRIPT LANGUAGE="JavaScript">
 treść skryptu
</SCRIPT>
```

- Przykład

```
<HTML><HEAD></HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
// obiekt document i jego metoda write - wypisanie
//tekstu
document.write ("To jest zwykły tekst<BR>");
document.write ("<BR>"); // wysyłamy znacznik html
 //przypisujemy wartość zmiennej
x=5;
// ... i wyświetlamy jej wartość
document.write("Wartość zmiennej <I>x</I> : "+x);
document.write ("<BR>To jest liczba PI:"+Math.PI);
</SCRIPT>
</BODY></HTML>
```

Komentowanie skryptu

- Komentarz o kilku wierszach,

```
/* …
treść komentarza
 … */
```

- Jednowierszowy komentarz

```
// tekst
```

Operatory

- Arytmetyczne:
  - + - * / %(reszta z dzielenia)
- Przypisania:
  - =
  - += x+=5 odpowiada x=x+5,
  - –= x–=5 odpowiada x=x–5,
  - *= x*=5 odpowiada x=x*5,
  - /= x/=5 odpowiada x=x/5,
  - %= x%=5 odpowiada x=x%5 (reszta z dzielenia)
- Przykładowo:

```
x = 7; x += 4; x %= 10;
```

- Konkatencja:

```
"Mateusz"  +  "Kowalski"      (" lub `)
```

Operatory

- Porównania:
  - == !=(nierówne) <= < > >=
- Przykład:
  - 4==5 3>=x 3!="3" (x==2)<(3*y+50);
- Porównania używane są w instrukcji *if* i pętlach warunkowych
- Można też przypisywać ich wartość wartość do zmiennej typu Boolean
- Logiczne:
  - koniunkcja (i) &&
  - alternatywa (lub) ||
  - negacja !

```
true && false daje false
!false daje true
```

Instrukcje języka

- Instrukcje języka oddzielane są średnikami (jeśli zapisywane są w tym samym wierszu).
- Blok instrukcji otaczany jest klamrami { } – zazwyczaj w ciele funkcji, instrukcjach warunkowych i iteracyjnych (np. *if, for, while*) – tak samo co: begin end w Pascalu.
- Ważne są duże i małe litery w identyfikatorach zmiennych, obiektów, metod, funkcji!!!!

Instrukcja warunkowa – if … else

- Instrukcja *if* powoduje wykonanie kodu źródłowego instrukcja1 tylko wtedy, gdy warunek logiczny jest spełniony.

- Jeżeli zostanie użyty poszerzony wariant instrukcji if, to po spełnieniu warunku zostanie wykonany kod instrukcja1 lecz w przeciwnym wypadku zostanie wykonany kod instrukcja2

```
if (warunek) {

kod wykonywany jeżeli warunek spełniony

}

else {

kod wykonywany jeżeli warunek nie spełniony

}
```

Zagnieżdżanie if … else

```
<HTML><HEAD></HEAD><BODY>
<SCRIPT LANGUAGE="JavaScript">
document.write ("To jest zwykły
tekst<BR>");
x=5; //przypisujemy wartośc
zmiennej
     // wyświetlamy jej wartość
document.write("Wartość zmiennej
<I>x</I> : "+x+"<BR />");

if (x>0)
document.write("tak"); //
warunkowo
else document.write("nie");

</SCRIPT>
</BODY></HTML>
```

```
if (x>0)
document.write("dodatnie"); //
  warunkowo
else if (x==0)
document.write("zero");
else
document.write("ujemne");

</SCRIPT>
</BODY></HTML>
```

Pętla - for

- Pętla *for* powtarza instrukcje wnętrza pętli aż do momentu, kiedy testowany warunek staje się fałszywy
- W JavaScript pętla *for* jest podobna do pętli w Java i C
- Składnia:

```
for ( inicjalizacja licznika; test_logiczny;
  inkrementacja )
    instrukcja;
```

- lub

```
for ( inicjalizacja licznika; test_logiczny;
  inkrementacja )
{
    instrukcja1;
    instrukcja2;
    ...
}
```

Pętla – for

- Przykład:

```
<HTML>
<HEAD></HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
  for (i=0; i<11; i++)
  {
  document.write(i+"<BR>");
  }
</SCRIPT>
</BODY></HTML>
```

Pętla – while

- Składnia:

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

- Wykonuje iterację (powtarza instrukcję) gdy warunek spełniony, gdy nie, wtedy przechodzi dalej
- Sprawdzenie warunku na początku
- Przykład:

```
n = 0;
x = 0;
  while( n < 10 )
{
      n ++;
      x += n;
      document.write(x+"<BR>");
}
```

Pętla – do while

- Składnia:

```
do
   instrukcja
while (warunek);
```

- Wykonuje iteracje (powtarza instrukcję) gdy warunek spełniony, gdy nie, wtedy przechodzi dalej
- Sprawdzenie warunku na końcu
- Przykład:

```
n = 0;
x = 0;
  do
{
      n ++;
      x += n;
      document.write(x+"<BR>");
} while ( n < 10 )
```

while czy do while ?

- Pętle while i do.. while różnią się tym,
  - że w pętli *while* warunek może być od razu fałszywy i instrukcja nigdy się nie wykona
  - zaś w pętli *do.. while* instrukcja zawsze jest co najmniej raz wykonana.

Przełącznik - switch

- Instrukcja *switch* pozwala programowi na sprawdzenie ciągu warunków i próbuje wykonać wartość wyrażenia przypisaną do odpowiedniej etykiety *case*
- Jeśli wartość wyrażenia jest znaleziona, program wykonuje instrukcję
- **Składnia:**

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

Funkcje

- Definiowane w bloku HEAD
- Możliwe też umieszczenie w osobnym pliku *.js i wtedy w bloku HEAD należy o tym poinformować stosownym kodem:

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

- Wykorzystywane w skryptach w bloku BODY

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function dodaj(s1,s2) {
    s3=Number(s1)+Number(s2);
    alert(s3);
    }
</SCRIPT>
</HEAD>

<BODY  style="font-size:28px">
…
<SCRIPT LANGUAGE="JavaScript">
dodaj(3,5);
</SCRIPT>
…
</BODY>
```

Obiekty Math

- Obiekt *Math* przechowuje wartości matematyczne, jako właściwości i metody
- Przechowywane są pewne stałe i funkcje matematyczne.
- Składnia:

`Math.property`

lub

`Math.method`

gdzie *property* lub *method* jest jednym z podanych obok elementów

- Przykład:

```
<SCRIPT LANGUAGE="JavaScript">
for (i=0; i<91; i++)
   {
   document.write(i+"
   "+Math.sin(i*Math.PI/180)+"<br>");
   }
</SCRIPT>
```

- zauważmy, że metoda write jest wieloargumentowa (argumenty oddzielane przecinkami)

E	**e - stała Eulera, która wynosi ok. 2.718**
PI	**wartość liczby π, czyli ok. 3.14159**
LOG10E	**logarytm o podstawie 10 z liczby Eulera, czyli ok. 0.434**
LOG2E	**logarytm o podstawie 2 z liczby Eulera, czyli ok. 1.442**
LN10	**logarytm naturalny z dziesięciu, tj. ok. 2.302**
LN2	**logarytm naturalny z 2, tj. ok. 0.693**
SQRT1_2	**pierwiastek kwadratowy z 0.5, czyli ok. 0.707**
SQRT2	**pierwiastek kwadratowy z 2, czyli ok. 1.414**

abs(wyrażenie)	**wartość bezwzględna *liczby***
cos(wyrażenie) sin(wyrażenie) tan(liczba)	**funkcje trygonometryczne (argument w radianach)**
ceil(liczba)	**zaokrąglenie do całkowitej w górę**
floor(liczba)	**zaokrąglenie do całkowitej w dół**
round(liczba)	**zaokrąglenie do najbliższej całkowitej**
exp(liczba)	**e^x**
log(liczba)	**logarytm naturalny *liczby* !**
pow(liczba1,liczba2)	**wartość *liczby1* podniesionej do potęgi *liczby2***
random()	**wartość pseudolosowa z przedziału (0,1)**
sqrt(liczba)	**pierwiastek kwadratowy *liczby***

Alert

- Metoda dla obiektu *window*, tworząca okienko dialogowe z napisem informacyjnym

```
<script language="JavaScript">
        alert("Witaj!");
</script>
```

  - *dokładnie window.alert ale domyślny obiekt to window*
- **Można wykorzystać atrybuty elementu HTML do wykonania akcji**
- **Kod Javascript może być tworzony bezpośrednio jako wartość atrybutu:**
  - *onblur* - atrybut definiuje reakcję na opuszczenie aktywnego elementu (np. tabulatorem),
  - *ondblclick* - atrybut definiuje reakcję na podwójne kliknięcie myszy,
  - *onfocus* - atrybut definiuje reakcję na uaktywnienie elementu,
  - *onkeydown* - atrybut definiuje reakcję na wciśnięcie klawisza myszki

```
<input type="text"
name="Button1" onfocus='alert("Edycja")' onblur='alert("Kliknales
  poza polem")' />
```

Obiekt Date

- Obiekt *Date* może operować z datą i czasem - latami, dniami, godzinami, minutami, sekundami etc.
- JavaScript przechowuje daty jako **liczby milisekund (!),** które upłynęły od 1 stycznia 1970, godz. 00:00:00.
- Składnie:

```
zmienna = new Date()
// jest zwracana bieżąca data (new – nowy obiekt)
  zmienna = new Date(hours, minutes, seconds)
//bieżący czas
  zmienna = new Date(year, month, day)
// bieżący rok, miesiąc i dzień
```

- Właściwości: brak

Obiekt Date

getDate()	wyświetla dzień miesiąca dla ustalonej daty (liczba całkowita z zakresu 1-31)
getDay()	wyświetla dzień tygodnia dla ustalonej daty (liczba całkowita, od 0=Niedziela do 6=Sobota)
getHours()	wyświetla godzinę dla ustalonej daty (liczba całkowita z zakresu 0-23)
getMinutes()	wyświetla minuty dla ustalonej daty (liczba całkowita z zakresu 0-59)
getMonth()	wyświetla miesiąc dla ustalonej daty (liczba całkowita z zakresu 0=Styczeń 11=Grudzień)
getSeconds()	wyświetla sekundy dla bieżącego czasu (liczba całkowita z zakresu 0-59)
getTime()	pokazuje ustaloną datę z użyciem liczb (liczba milisekund od 1 stycznia 1970 00:00:00)
getYear()	wyświetla rok dla ustalonej daty (liczba dwucyfrowa)
setDate()	ustawia dzień miesiąca dla aktualnego obiektu Date
setHours()	ustawia godzinę dla aktualnego obiektu Date
setMinutes()	ustawia minuty dla aktualnego obiektu Date
setMonth()	ustawia miesiąc dla aktualnego obiektu Date
setSeconds()	ustawia liczbę sekund dla aktualnego obiektu Date
setTime()	ustawia datę i godzinę dla aktualnego obiektu Date, w milisekundach od 1/1/70 00:00:00
setYear()	ustawia rok dla aktualnego obiektu Date (rok jest liczbą całkowitą większą od 1900)

Metoda Parse

- Zwraca liczbę milisekund dla daty podanej ciągiem tekstowym w ustalonym formacie

```
document.wtite(Date.parse("Aug 9, 1995"));
```

- Przykład:

```
<html>
<head>
</head>
<body>
<script LANGUAGE="JavaScript">
   today = new Date() ;//nowy obiekt Date
document.write(today.getTime()/1000/60/60/24);
//pokaże ile lat godzin minęło od 1.01.1997

document.write
    ("Aktualny czas to:",today.getHours(),":",today.getMinutes());
   document.write
 ("<br />Aktualna data to:",today.getDate()+1,"-", today.getMonth(),"-
   ",today.getYear());
</script>
</body>
</html>
```

Obiekt Window

- Obiekt window jest najwyższym w hierarchii obiektem dla każdego obiektu *location*, *history* lub *document*.
- Składnia:

```
window = window.open("URL", "NAME","dodatkowe własności")
```

  - window jest nazwą okna,
  - URL jest adresem URL strony
  - NAME jest nazwą okna,
  - dodatkowe własności określają wielkość okna, położenie itp.
- Wiele sposobów dostępu do właściwości i metod okna, najczęściej używa się:
  - window.właściwość
  - window.metoda(parametry)

Obiekt Window

- Właściwości

document	Dotyczy bieżącego dokumentu
frames	Tablica zawierająca listę wszystkich ramek w oknie
frame	Przewijalne okno używane dla tablicy
length	Liczba ramek w bieżącym oknie
location	Pełny adres URL bieżącego dokumentu
name	Nazwa okna
self	Dotyczy bieżącego okna
top	Dotyczy okna położonego najwyżej w hierarchii (głównego)

- Metody

alert	Tworzy okienko dialogowe z alertem
close	Zamyka dokument
confirm	Tworzy okienko dialogowe z potwierdzeniem z przyciskami OK i Cancel
open	Otwiera nowe okno
prompt	Tworzy okienko dialogowe zachęcające do wprowadzenia jakichś danych
setTimeout	Wykonuje skrypt JavaScript po upłynięciu podanej liczby milisekund
clearTimeout	Przerywa polecenie setTimeout

Obiekt Document

- Obiekt *document* jest ważnym obiektem JavaScript.
- Zawiera informacje o bieżącej stronie i dostarcza sposobów wyświetlania strony HTML
- Składnia:
  - document.property

lub

  - document.method

gdzie *property* i *method* są jednym z elementów poniższej listy

- Metody:
  - clear, close, open, write i writeln

Obiekt Document

- Wybrane właściwości

alinkColor	określa kolor aktywnego odsyłacza (active link color)
linkColor	określa kolor odsyłaczy
vlinkColor	określa kolor odsyłacza do odwiedzonej już wcześniej strony (visited link color)
bgColor	określa kolor tła
fgColor	określa kolor tekstu
lastModified	informuje, kiedy dokument był ostatnio modyfikowany
location	wyświetla bieżący adres URL dokumentu
title	wyświetla zawartość znacznika <TITLE>

Przykłady

- Przykład – okno *prompt*, opóźnienie wykonania – metoda dla obiektu window

```
<HTML>
<HEAD>
</HEAD>
<SCRIPT LANGUAGE = "JavaScript">
   function x()
   {
   var imie = prompt ("Podaj swoje imie:", "");
   if (imie == null)
   {
       document.write ("<H3>Anulowales</H3>");
   }
   else
   {document.write ("Czesc " + imie + "!");
   }
   }
</SCRIPT>
<BODY>
   <SCRIPT LANGUAGE = "JavaScript">
   setTimeout("x()",2000); //wykonanie po 2 sekundach
</SCRIPT>
</BODY>
</HTML>
```

Przykłady

- Przykład – otwarcie nowego okna w przeglądarce

```
<html>
<head>
<SCRIPT language=javascript>
   function podaj(x){
var nokno=window.open('','okno','width=500,height=400,left=180,top=80');
   a=x.t1.value;
   b=x.t2.value;
   c=x.t3.value;
   delta=b*b-4*a*c;
   nokno.document.write("delta="+delta+"<BR>");
   if (delta<0) nokno.document.write("Nie ma pierwiastkow")
          else
             if(delta==0)
                    {nokno.document.write("x1="+(-b/2/a))                    }
             else
            {nokno.document.write("x1="+(-b-Math.sqrt(delta)/2/a)+"<BR>"+"x2="+(-
   b+Math.sqrt(delta)/2/a))
             }}
</SCRIPT>
</head>
<body>
   <form onsubmit="podaj(this)"><!—argumentem jest aktualny obiekt FORM-->
   a<input type="text" name="t1">
   b<input type="text" name="t2">
   c<input type="text" name="t3">
   <input type=submit value=Kliknij>
   </FORM>
</body>
</html>
```

Przykłady

- Przeglądarka obrazków

```
<HTML>
<HEAD>
<SCRIPT language=javascript>
   function otworzObrazek(x)
   {
     var
   fotka=window.open(x,'fotka','width=800,height=600,left=80,top=20');
    }
</SCRIPT>
</HEAD>
<BODY>
   <IMG src="1.JPG" width=10% height=10% style="cursor:hand"
   onclick="otworzObrazek('1.JPG');" >
</BODY>
</HTML>
```

Przykłady

- Animacja tekstu dokumentu

```
<html>
<head>
<SCRIPT language=JavaScript>
   function C(element,x)
   {
       x+=1;
       //nadajemy wartość właściwości innerHTML dla elementu
       element.innerHTML+=String(x)+" ";//można też zamiast += dać samo =
       setTimeout("C(element,"+x+")", 1000);
   }
</SCRIPT>
</head>
<BODY>
   <DIV id="elem" style="color:blue;font-size:50px;"> TUTAJ </DIV>
   <SCRIPT language="JavaScript">
   element=document.getElementById('elem');
   C(element,0);
</SCRIPT>
</body>
</html>
```

- Główny skrypt inicjuje wykonanie funkcji *C(element,x)*.
- Funkcja *C(element,x)* w swoim ciele zawiera wykonanie samej siebie z inkrementowaną (zwiększaną o 1) wartością x. Odbędzie się nieskończenie wiele wywołań funkcji co 1000 milisekund (1 sekundę).

Przykłady

- Wykorzystanie pól edycyjnych do akcji

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function pobierz(s1,s2) {   //dwa argumenty!!
    s3=Number(s1)+Number(s2);
    alert(String(s3));
   //dodatkowo
   //document.write(s3);
  }
</SCRIPT>
</HEAD>
<BODY  style="font-size:28px">
Wpisz dwie liczby:
<FORM>
 <input type="text" name="pole1" style="font-size:28px"><BR />
 <input type="text" name="pole2" style="font-size:28px"><BR />
 <input type="button" name="Button1" value="Naciśnij" onclick="pobierz(pole1.value,
   pole2.value)">
  </FORM>
</BODY>
</HTML>
```

Przykłady

- Wyświetlanie aktualnej daty

```
<html>
<head>
</head>
<body style="font-size:25px">
<script LANGUAGE="JavaScript">
var today = new Date() ;//nowy obiekt Date
document.write("ile dni minęło od 1.01.1970"+"<BR>");
document.write(today.getTime()/1000/60/60/24+"<BR>");
document.write ("Aktualny czas to:",today.getHours(),":",today.getMinutes());
document.write
 ("<br />Aktualna data to:"+
today.getDate()+"-"
+ today.getMonth()+1+"-"
+today.getFullYear());
</script>
</body>
</html>
```

Przykłady

- Wyświetlanie daty urodzin

```
<html>
<head>
</head>
<body style="font-size:25px">
<script LANGUAGE="JavaScript">
var ur = new Date();
ur.setDate(23);
ur.setMonth(11);
ur.setFullYear(1976);
document.write (ur);
document.write
   ("<BR>Urodzony:"+
   ur.getDate()
   +"."
   +ur.getMonth()
   +"."+ur.getFullYear());
</script>
</body>
</html>
```

Pytania sprawdzające

- Co to jest JavaScript?
- Jaka jest różnica miedzy językami JavaScript a Java?
- Jak osadza się JavaScript?
- Jak komentuje się skrypty?
- Operatory w JavaScript?
- Podstawowe instrukcje w JavaScript?
- Funkcje i ich wykorzystanie?
- Alert - co to jest?
- Przykładowe obiekty w Java?

Netografia

- PHP i MySQL. Tworzenie stron internetowych. Helion
- http://pl.docs.pld-linux.org/
- http://coogi.cba.pl
- Kurs HTML http://www.kurshtml.edu.pl
- Za zgodą - fragmenty materiałów dydaktycznych dr inż. Tomasza Bajorka
- http://www.a4.pl
- manual na stronie z instrukcjami http://pl2.php.net/FAQ.php
- PHP-owa witryna: PHP-Nuke http://www.phpnuke.org

Koniec

Temat następnego wykładu to:

Java Script