

## Ćwiczenie 9

# JavaScript – skrypty klient-side – część 2

Czas trwania: 90 min.

Materiały opracowane na podstawie ćwiczeń dr. Tomasza Bajorka

Zadania do wykonania:

**1. W oparciu o instrukcję przećwiczyć kilka przykładów zastosowania języka JavaScript.**

(Czas trwania: ok. 45 min.)

W instrukcji zamieszczono przykłady kilku skryptów napisanych w technologii JavaScript. Należy je przećwiczyć sprawdzając ich działanie jako elementów dodanych do strony WWW.

**2. Na podstawie przeprowadzonych ćwiczeń wykonać zadania przedstawione na końcu instrukcji**

(Czas trwania: ok. 45 min.)

### Objaśnienia

JavaScript – jest językiem skryptowym typu klient-side, czyli uruchamianym przez przeglądarkę internetową po stronie klienta. Język doskonale współdziała w zakresie współpracy z kodem PHP oraz HTML. Za jego pomocą można wykonywać czynności pomocnicze, komplementarne dla budowanego serwisu. Jest składnikiem technologii dynamicznych.

### Ćwiczenia praktyczne

1. 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.

Składnia:

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

Właściwości: brak

Metody:

getDate()	pobiera dzień miesiąca dla ustalonej daty (liczba całkowita z zakresu 1-31)
getDay()	pobiera dzień tygodnia dla ustalonej daty (liczba całkowita, od 0=Niedziela do 6=Sobota)
getHours()	pobiera godzinę dla ustalonej daty (liczba całkowita z zakresu 0-

	23)
getMinutes()	pobiera minuty dla ustalonej daty (liczba całkowita z zakresu 0-59)
getMonth()	pobiera miesiąc dla ustalonej daty (liczba całkowita z zakresu 1..12)
getSeconds()	pobiera sekundy dla bieżącego czasu (liczba całkowita z zakresu 0-59)
getTime()	oblicza liczbę milisekund od 1 stycznia 1970 00:00:00
getFullYear()	pobiera rok dla ustalonej daty (liczba 4-cyfrowa)
getMilliseconds()	wyświetla liczbę milisekund od 1/1/70 00:00:00 do podanej daty
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
setFullYear()	ustawia rok dla aktualnego obiektu Date (rok jest liczbą całkowitą większą od 1900)

Wykonać poniższy przykład

```
<html>
<head>
</head>
<body>
<script LANGUAGE="JavaScript">
var t1 = new Date(2011,3,6,1,1,59) ;//nowy obiekt
document.write("Aktualny czas to:
",t1.getHours(),":",t1.getMinutes(),":",
t1.getSeconds());
document.write("<BR>Aktualna data to: "+
t1.getDate()+"/",t1.getMonth()+1+"/", t1.getFullYear());
document.write(" -: "+t1.getDay());
var inna = new Date() ;//nowy obiekt Date
inna.setFullYear(2001);
inna.setDate(23);
inna.setMonth(11);
document.write("<BR>Inna data to: "+ inna.getDate()+"/"+
inna.getMonth()+1+"/"+ inna.getFullYear());
</script>
</body></html>
```

#### Zadanie do wykonania:

Utworzyć skrypt obliczający ile dni żyjemy. (Wskazówka: stworzyć dwa obiekty Date, dla obu ustawić daty, znaleźć czasy milisekundowe funkcją getTime, odjąć, przeliczyć na dni).

## 2. 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, którą chcesz otworzyć, NAME jest nazwą okna, a dodatkowe własności określa wielkość okna, położenie itp.

Przykład – okno prompt, opóźnienie wykonania:

```
<HTML><HEAD></HEAD>  
<SCRIPT LANGUAGE = "JavaScript">  
function x()  
{  
var imie = window.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>
```

Przykład - otwarcie nowego okna

```
<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 funkcji 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>
```

### 3. Obiekt document

Obiekt document jest ważnym obiektem JavaScript. Zawiera on 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.

#### 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>

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

#### Prosty przykład 1:

```
<HTML><BODY>  
<SCRIPT language=javascript>  
document.bgColor='black';  
document.linkColor='red';  
document.vlinkColor='white';  
document.write("<A  
HREF='http://www.onet.pl'>ONET</A>");</SCRIPT>  
</BODY></HTML>
```

#### Przykład 2 – przeglądarka obrazków

```
<<HTML><HEAD>  
<SCRIPT language=javascript>  
function otworzObrazek(x)  
{
```

```
var
fotka=window.open(x, 'fotka', 'width=800,height=600,left=8
0,top=20');
}
</SCRIPT>
</HEAD><BODY>
<IMG src="1.JPG" width=10% height=10%
style="cursor:hand" onclick="otworzObrazek('1.JPG');" >
</BODY></HTML>
```

## Ćwiczenia

1. Przeanalizować i dołączyć przykłady z ćwiczenia do własnej strony.
2. Dołączyć do własnej strony wyświetlenie daty, nazwy dnia tygodnia (stworzyć nazwę przy pomocy instrukcji switch dla metody getDay) i czasu na stronie czołowej.
3. Wyświetlić swoją datę urodzenia - obliczyć ile żyjemy minut, godzin, dni – pokazać wyniki na stronie
4. Spowodować, by w zależności od czasu (przed południem, po południu) strona czołowa miała dwa różne kolory tła. Wykorzystać instrukcję if oraz metodę:  
document.bgColor="red"  
albo  
document.bgColor="#77ff66"
5. Wykonać dokument HTML obliczający ile dni upłynęło od początku roku do dziś i ile zostało do końca roku.
6. Wykonać przeglądarkę w nowych oknach kilku dowolnych obrazów jpg (np. ściągniętych z internetu).
7. Animacja obiektu z wykorzystaniem stylu
8. Prostą animację utworzyć można stosując znaną nam już metodę setTimeout dla obiektu window – opóźnione wykonanie funkcji. Aby uzyskać ruch, funkcja musi być wywoływana wielokrotnie ze zmieniającymi argumentami. Zastosowano tu rekurencję – wywołanie funkcji odbywać się będzie z wnętrza tej funkcji ze zmieniającymi argumentami.
9. Animację obiektu z wykorzystaniem stylu (cecha left) pokazuje poniższy przykład. Rekurencyjnie wywołana funkcja ruch powoduje narastanie cechy left co 0.01 sekundy. Uwaga: styl wymaga wartości fixed dla cechy position (pozycjonowanie nie zmienia się gdy strona jest przewijana), a także konwersji zmiennej liczbowej x na łańcuch tekstowy (String).

```
<HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
function ruch(element,x){
x++;//powiększenie o 1
element.style.left=x;
setTimeout("ruch(element, "+x+")",100); //rekurencja
funkcji z opóźnieniem 0,1 s
}
</SCRIPT>
<BODY>
<form name=Formularz>
<Input type=button id=przycisk name=Przycisk value=Napis
style="position:absolute">
</form>
```

```
<SCRIPT LANGUAGE="JavaScript">
element=document.getElementById('przycisk');//element
przekazujemy do funkcji
ruch(element,1);//wykonanie funkcji
</SCRIPT>
</BODY></HTML>
```

### Zadanie 1.

Przeanalizować i dołączyć przykład z ćwiczenia do własnej strony.

Wykonać modyfikację przykładu z ćwiczenia, gdzie przycisk „wędruje”:

- o pionowo w dół
- o po skosie - w prawo w dół.
- o ruchem falującym (np. według sinusoidy)

Animacja tekstu dokumentu

Przykład:

```
<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;">
</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).

### Zadanie 2.

Ograniczyć rekurencyjne wykonanie funkcji do 20 razy (np. przy pomocy instrukcji if). Zmianę napisów (np. tekstu na przycisku) w reakcji na kliknięcie można łatwo wykonać stosując poniższą metodę:

```
<HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
function zmien() {
if (document.Formularz.Przycisk.value=="Stop")
  document.Formularz.Przycisk.value="Start"
else
  document.Formularz.Przycisk.value="Stop";
}
</SCRIPT>
<BODY>
<FORM name=Formularz>
<INPUT type="buton" name="Przycisk" value="Stop"
onClick="zmien()" ">
</FORM>
</BODY></HTML>
```

### Zadanie 3.

Zmodyfikować przykład, aby po kliknięciu przycisku napis na nim zmieniał się co sekundę: 0,1,2,3...(już prawie będziemy mieli zegar).

Można spowodować automatyczne wykonanie funkcji bez jakiegokolwiek zdarzenia:

```
<HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
function zmien(element)
{
setTimeout('zmien(element)',1000)
if (element.value=="Stop")
  element.value="Start"
else
  element.value="Stop";
}
</SCRIPT>
<BODY>
<FORM name=Formularz>
<INPUT type="button" value="Stop" id="prz">
</FORM>
<SCRIPT LANGUAGE="JavaScript">
element=document.getElementById("prz");
zmien(element);
</SCRIPT>
</BODY></HTML>
lub inicjując kliknięciem innego przycisku:
<HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
function inicjacja()
{element=document.getElementById('prz');
zmien(element);
}
function zmien(element)
{setTimeout('zmien(element)',1000)
```

```
if (element.value=="Stop")
  element.value="Start"
else element.value="Stop";
}
</SCRIPT>
<BODY>
<FORM name=Formularz>
<INPUT type=button name=Przycisk value=Stop id=prz >
<INPUT type=button value="Naciśnij"
onclick="inicjacja()" >
</FORM>
</BODY></HTML>
```

**Zadania dodatkowe:**

1. Przeanalizować i dołączyć przykład z ćwiczenia do własnej strony.
2. Spróbować:
  - a. Czy da się, aby przycisk po kliknięciu sam inicjował cykliczną zmianę swojej value?
  - b. Czy da się, aby dwa przyciski wzajemnie inicjowały cykliczną zmianę swoich value?
  - c. Czy da się, przycisk po kliknięciu sam inicjował cykliczną zmianę swojej value, a następne kliknięcie zatrzymywało cykl zmian?