JAVAScript – tablice, przekazanie danych do funkcji, obiekty **Date** i **window**

Tablice w JavaScript

Przykład zastosowania tablicy dla przekazania rezultatów funkcji do miejsca wywołania funkcji (omówiony na wykładzie):

```
<HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
function tab2(a,b){
        var x=new Array( )
        x[1]=(a+b)/2;
        x[2]=(a-b)/2;
        return x;
        }
</SCRIPT>
</HEAD><BODY>
<SCRIPT LANGUAGE="JavaScript">
//teraz wykonanie funkcji
a=tab2(10, 8); //wykonanie zdefiniowanej funkcji, wynik przechowaj w tablicy a
//wypisz wyniki – elementy tablicy
document.write(a[1],"<BR>");
document.write(a[2]);
</SCRIPT>
</BODY></HTML>
```

Przekazanie danych do funkcji z pola tekstowego INPUT

```
<HTML><HEAD>
<SCRIPT language="JavaScript">
  function nacisnij(a)
    { document.write(a); }
</SCRIPT>
</HEAD><BODY>
 <INPUT type="text" id="dane">
<INPUT type="submit" value="OK"   onclick="nacisnij(dane.value);">
</BODY> </HTML>
```

Zadania

1. Wykonać i zrozumieć działanie powyższych przykładów.
2. Utworzyć skrypt, którego zadaniem będzie wykorzystanie własnej funkcji obliczającej objętość i pole powierzchni kuli. Jako argument przekazujemy promień kuli, wynikowe wartości (pole i objętość) są zwracane są w postaci tablicy dwuelementowej (jeśli nie pamiętamy wzorów, znaleźć w internecie).
3. Powyższe zadanie wykonać również w wersji z interaktywnym wykonaniem funkcji na kliknięcie przycisku.
4. Powyższe zadanie wykonać w wersji z przekazaniem wartości promienia kuli z pola tekstowego.
5. Sprawdzić przykład z wykładu dotyczący dodawania dwóch tekstów z pól INPUT i – jeśli te dane zawierają interpretację liczbową – wykorzystanie funkcji **Number**(x) w celu konwersji tekstu na liczbę.

Obiekt Date

Wbudowany typ obiektowy ***Date*** (podobnie jak i *Math*) posiada pewne umiejętności - 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.

Wykorzystując obiekt ***Date*** można tworzyć wiele dat (instancje obiektu) używając operatora **new** dla utworzenia nowego obiektu (dokładniej INSTANCJI obiektu):

Składnie:

var *zmienna* = **new** Date( ) // jest zwracana bieżąca data

var *zmienna* = **new** Date(**rok**, **miesiąc**, **dzień** ,**godzina**, **minuty**, **sekundy**, **milisekundy**) // 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 0..11)
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)
getMiliseconds()	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)

Przykład:

```
<HTML><HEAD></HEAD><BODY>
<SCRIPT language="JavaScript">
var t1=new Date( ) //utworzenie nowego obiektu o nazwie t1 typu daty
document.writeln("Czas aktualny: ",t1.getHours(),":",t1.getMinutes(),":", t1.getSeconds());

var t2 = new Date(1981,11,13,1,1,59) ;//nowy obiekt – uwaga! miesiąc grudzień - 11
document.write("<BR>Stara data to: ", t2.getDate(),"-",t2.getMonth()+1,"-", t2.getFullYear());
document.write("<BR>Aktualny czas to: ",t2.getHours(),":",t2.getMinutes(),":", t2.getSeconds());
document.write(" Dzień: "+t2.getDay());

var inna = new Date( ) ;//nowy obiekt Date
inna.setFullYear(2001);
inna.setDate(23);
inna.setMonth(0);//styczeń
document.write("<BR>Inna data to: ", inna.getDate(),"-", inna.getMonth()+1,"-", inna.getFullYear());
</SCRIPT></BODY></HTML>
```

Zadania:

1. Utworzyć skrypt obliczający ile dni żyjemy.
   *Wskazówka: stworzyć dwa nowe obiekty Date(), dla obu ustawić daty, znaleźć czasy milisekundowe funkcją **getTime**, odjąć, przeliczyć na dni.*
2. Wyświetlić swoją datę urodzenia - obliczyć ile żyjemy minut, godzin, dni – pokazać wyniki na stronie.

3. Spowodować, by w zależności od czasu (przed południem, po południu) pojawiał się na stronie akapit z odpowiednią informacją.
4. Wykonać dokument HTML obliczający ile dni upłynęło od początku roku do dziś i ile zostało do końca roku.

Zadanie domowe: Utworzyć i wykorzystać w dokumencie własną funkcję, której działanie polega na przekazaniu do funkcji dnia tygodnia w postaci liczby (rezultat funkcji **getDay**()). Funkcja zamienia tę liczbę na odpowiednik tekstowy (np. "wtorek"), wyświetla rezultat na stronie.

Obiekt window

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

Niektóre właściwości:

Właściwość	Opis
document	zwraca obiekt **document**
location	zwraca obiekt **Location** (ma m.in. właściwość **pathname**)
name	zwraca nazwę okna
navigator	zwraca obiekt **Navigator** (ma m.in. właściwość **appName**)
screenLeft	zwraca współrzędną x okienka w stosunku do lewego górnego rogu ekranu
screenTop	zwraca współrzędną y okienka w stosunku do lewego górnego rogu ekranu

Wybrane metody

Metoda	Opis
alert( )	wyświetla okienko alert z tekstem i przyciskiem OK
prompt( )	wyświetla okienko dialogowe do wprowadzenia tekstu
open( )	otwiera nowe okno
close( )	zamyka bieżące okno
moveTo(x, y)	przesuwa okno do pozycji x y w stosunku do lewego górnego rogu ekranu
setTimeout( )	wywołuje funkcję lub przelicza wyrażenie po wyspecyfikowanej liczbie milisekund

Przykłady:

```
window.open("", "okienko" "height=200,width=200,fullscreen=no")
window.alert(tekst);
x=window.prompt("Podaj x:")
```

Uwaga: window jest obiektem domyślnym, a zatem można pomijać odniesienie do obiektu:

```
open("1.jpg","Obrazek", "height=200,width=200,fullscreen=no");
alert("tekst");
```

Zadania

1. Wykonać i przeanalizować poniższe skrypty:
2. Utworzyć plik o nazwie **okienko.html** z kodem:

```
<HTML> <HEAD> </HEAD><BODY>
<PRE>
<SCRIPT language="JavaScript">
document.writeln(window.name);
document.writeln(window.length);
document.writeln(window.location.pathname);
document.writeln(window.document.location);
document.writeln(window.screenTop);
document.writeln(window.screenLeft);
document.writeln(window.navigator.appName);
</SCRIPT>
<INPUT type="button" value="Kliknij" onClick="window.moveTo(0,0);" />
</PRE>
</BODY></HTML>
```

UWAGA: Gdy otoczymy skrypt znacznikiem <PRE> można stosować metodę **writeln**, która nie wymaga umieszczania znacznika
 w celu przenoszenia kolejnego tekstu do następnego wiersza.

3. Następnie utworzyć główny plik HTML ze skryptem wykorzystującym okienko:

```
<HTML> <HEAD> </HEAD><BODY style="font-size:20px">
<SCRIPT language="JavaScript">
window.open("okienko.html","Moje_okno", "left=100,top=100,height=400,width=400,fullscreen=no")
</SCRIPT>
</BODY></HTML>
```

4. Wypróbować działanie, zinterpretować informacje ukazujące się w okienku.
5. Zmodyfikować przykład tak, aby pojawienie się okienka było inicjowane kliknięciem w przycisk <INPUT>
6. Okno **prompt i** opóźnienie wykonania **setTimeout**:

```
<HTML><HEAD></HEAD><BODY>
<SCRIPT LANGUAGE = "JavaScript">
function x( ){
imie = prompt ("Podaj swoje imie:", "");
if (imie == null)
        document.write ("<H3>Anulowaleś</H3>");
 else
        document.write ("Czesc " + imie + "!");
}//koniec definicji funkcji
//teraz wykonujemy funkcję x( ) po 2 sekundach
window.setTimeout("x( )",2000);
</SCRIPT> </BODY></HTML>
```

7. Przeglądarka obrazków w okienkach – wykorzystanie atrybutu **onclick** dla znacznika (wcześniej w folderze z plikiem HTML umieścić obrazek o nazwie 1.jpg):

```
<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" style="cursor:hand;width:2cm" onclick="otworzObrazek('1.JPG');" >
<P> Zmniejszony obrazek</P>
</BODY></HTML>
```

Dobrać wymiary okienka do rozmiarów obrazka.

8. Wykonać przeglądarkę obrazków w nowych oknach kilku dowolnych obrazów jpg (np. ściągniętych z internetu).

Zadanie domowe (trudniejsze)

W folderze umieścić dwa obrazki o nazwach 1.jpg i 2.jpg.

Napisać dwa skrypty:

obrazek.html

```
<HTML>
<HEAD> </HEAD>
<BODY>
<SCRIPT language="JavaScript">
document.write('<IMG src="', name,'.jpg','" />','<P>', name ,'.jpg','</P>' );
</SCRIPT>
</BODY>
</HTML>
```

obrazki.html

```
<HTML><HEAD></HEAD><BODY>
<SCRIPT language=javascript>
function otworzObrazek(x,y)
{
var y=y.toString();//zamiana liczby y na tekst
var fotka=window.open(x,y,'width=800,height=600,left=80,top=20');
 }
a1="<IMG src=\"";
a2=".JPG\" style=\"cursor:hand;width:2cm\" onclick=\"otworzObrazek(\'obrazek.html\',";
a3=");\" />";
for (var k=1;k<=2;k++)
        document.write(a1,k,a2,k,a3);
</SCRIPT>
<P> Zmniejszony obrazek</P>
</BODY></HTML>
```

Zrozumieć działanie przeglądarki obrazków w okienku uruchamiając w przeglądarce plik obrazki.html. Poszerzyć skrypt na działanie z kilkoma obrazkami.

Uwaga: odwrotne ukośniki \ przed znakiem " lub ' pozwalają na umieszczenie tych znaków wewnątrz stałych tekstowych.