

# *przydatne strony*

<http://kursjs.pl/kurs/style.html>

<http://kursjs.pl/kurs/date.html>

<http://msdn.microsoft.com/pl-pl/library/kurs-javascript--jak-zdefiniowac-czas-odswiezania-elementu-w-javascript.aspx>

***setTimeout***

•

***setInterval-***

•

***metody obiektu window***

# setTimeout

`setTimeout('nazwaFunkcji()', milisekundy,param1,param2,...);`

Funkcja setTimeout() przyjmuje parametry: - pierwszym jest kod JavaScript do uruchomienia po upływie określonego czasu, a drugim czas w milisekundach, pozostałe-opcjonalne. Funkcja `setTimeout()` pozwala na pojedyncze uruchomienie kodu JavaScript po upływie określonego czasu.

```
<body >
  <div >po 2 sekundach pojawi się tekst</div>
  <div ></div>
<script type="text/javascript">
function setTimeoutPrzyklad1 () {
  document.getElementsByTagName ("div") [1].innerHTML="jakis tekst";
}
setTimeout ("setTimeoutPrzyklad1 () ", 2000);
</script>
</body>
```

po 2 sekundach pojawi się tekst

po 2 sekundach pojawi się tekst  
jakis tekst

# clearTimeout()

## Syntax

```
clearTimeout(id_of_settimeout)
```

Parameter	Description
<i>id_of_settimeout</i>	Required. The ID value of the timer returned by the setTimeout() method

**Note:** To be able to use the clearTimeout() method, you must use a global variable when creating the timeout method:

```
myVar = setTimeout("javascript function",milliseconds);
```

## clearTimeout(myVar)

```
<body >
  <div id="a1"></div>
  <button onclick="uruchomStoper()">uruchom stoper</button>
  <button onclick="zatrzymajStoper()">zatrzymaj stoper</button>
</script >
var t;
var c=0;
function uruchomStoper() {
  document.getElementById("a1").innerHTML=c;
  c+=1;
  t=setTimeout("uruchomStoper()", 1000);
}
function zatrzymajStoper() {
  clearTimeout(t);
}
</script>
</body>
```

4

uruchom stoper

zatrzymaj stoper

# setTimeout() -rozbudowany stoper

```
<body >
<div id="a1"></div>
<button onclick="uruchomStoper()">uruchom stoper</button>
<button onclick="zatrzymajStoper()">zatrzymaj stoper</button>
<button onclick="resetStoper()">reset stoper</button>
<script >
var t;
var c=0;
function uruchomStoper() {
//poniższa instrukcja jest potrzebna w przypadku,
//gdy użytkownik kliknie kolejno kilka razy na pierwszy button
clearTimeout(t);
document.getElementById("a1").innerHTML=c;
c+=1;
t=setTimeout("uruchomStoper()", 1000);
}
function zatrzymajStoper() {
clearTimeout(t);
}
function resetStoper() {
c=0;
//zatrzymujemy stoper
clearTimeout(t);
uruchomStoper();
}
</script>
</body>
```

2

uruchom stoper

zatrzymaj stoper

reset stoper

# setTimeout() -prosta galeria

```
<script>
var liczba_klatek=5;
var opoznienie=1000;
obrazek0=new Image( ) ;
obrazek0.src="image/image0.jpg";
obrazek1=new Image( ) ;
obrazek1.src="image/imagel.jpg";
obrazek2=new Image( ) ;
  obrazek2.src="image/image2.jpg";
obrazek4=new Image("image/image3.jpg" ) ;
obrazek3=new Image( ) ;
obrazek3.src="image/image3.jpg";
obrazek4=new Image( ) ;
obrazek4.src="image/image4.jpg";
var licznik=2;
function zmien( )
{ document.images[0].src=eval("obrazek"+licznik+".src")
  licznik++;
  // if(licznik+1>liczba_klatek)
  //   licznik=0;
  licznik=licznik%5;
  setTimeout("zmien( )",opoznienie);
}
</script>
```



```
<body onload="zmien()" bgcolor="aquamarine">

<table border="1">
<tr>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
</tr>
</table>
</body>
```

# setInterval()

`setInterval('nazwaFunkcji()', milisekundy, param1, param2....);`

Funkcja `setInterval()` pozwala na cykliczne uruchomienie kodu JavaScript po upływie określonego czasu.

Param1, param2... - opcjonalnie

`toLocaleTimeString();` -wyswietla czas

`toLocaleString();` //wyswietla datę i czas

## Przykład – zegar

```
<body >
<div id="zegar" style="height:50px; width:350px"></div>
<script type="text/javascript">
function wyswietlCzas () {
var czas = new Date ();
//document.getElementById("zegar").innerHTML =czas.toLocaleTimeString(); -wyswietla czas
document.getElementById("zegar").innerHTML =czas.toLocaleString(); //wyswietla datę i czas
}
//ponizej funkcja "wyswietlCzas()" wywołuje się co sekundę
setInterval ("wyswietlCzas()", 1000);
</script>
</body>
```

3.01.2015, 17:55:28

# Przykład z zegarem – II sposób

Zastosujemy funkcję anonimową:

```
<body >  
  <div id="zegar" style="height:50px; width:350px"></div>  
  <script >  
    setInterval(function(){var czas = new Date();  
                  document.getElementById("zegar").innerHTML =czas.toLocaleTimeString();}, 1000);  
  </script>  
</body>
```

3.01.2015, 17:55:28



# Przykład z zegarem –

dołączymy efekty graficzne - zmieniające się tło zegara

```
<body >
  <div id="zegar" style="height:50px; width:350px"></div>
<script type="text/javascript">
function wyswietlCzas () {
  var czas = new Date();
  var str=czas.toLocaleString() ;
  document.getElementById("zegar").innerHTML =czas.toLocaleString();
  }
  setInterval("wyswietlCzas()", 1000);

function zmienStyl () {
  var element =document.getElementById("zegar");
  element.style.backgroundColor="rgb("+i+", "+j+", "+k+)";
  document.getElementById("zegar").style.textAlign="center";
  if (i>255 || i<0) {di=-di;}
  if (j>255 || j<0) {dj=-dj;}
  if (k>255 || k<0) {dk=-dk;}
  i+=di;
  j+=dj;
  k+=dk;
  }
// zmienne globalne:
  var i=0, j=240, k=100;
  var di=3, dj=2, dk=15, dz=1;
setInterval("zmienStyl()", 100 );
</script>
</body>
```

3.01.2015, 17:40:09

# Cd przykładu z zegarem –

Prosta  
animacja

```
7 <body >
8 <div id="zegar" style="height:50px; width:350px"></div>
9 <script type="text/javascript">
0 function wyswietlCzas() {
1   var czas = new Date();
2   var str=czas.toLocaleString() ;
3   document.getElementById("zegar").innerHTML =czas.toLocaleString();
4   }
5 setInterval("wyswietlCzas()", 1000);
6 function zmienStyl() {
7   var element =document.getElementById("zegar");
8   element.style.backgroundColor="rgb("+i+", "+j+", "+k+)";
9   document.getElementById("zegar").style.fontSize=z+"px";
0   document.getElementById("zegar").style.textAlign="center";
1   if (i>255 || i<0) {di=-di;}
2     if (j>255 || j<0) {dj=-dj;}
3     if (k>255 || k<0) {dk=-dk;}
4     if (z>40 || z<2) {dz=-dz;}
5   i+=di;
6   j+=dj;
7   k+=dk;
8   z+=dz;
9   }
0 // zmienne globalne:
1   var i=0, j=240, k=100;
2   var di=3, dj=2, dk=15, dz=1;
3   var z=2;
4   setInterval("zmienStyl()", 100 );
5 </script>
6 </body>
```

3.01.2015, 17:53:55

# setInterval-znikający tekst

```
5 <script >
6
7
8   function Animuj ()
9   {
10      var element = document.getElementById('napis').style.display;
11      if (element == "none")
12      {
13          document.getElementById('napis').style.display = "inline";
14      }
15      else
16      {
17          document.getElementById('napis').style.display = "none";
18      }
19  }
20  setInterval("Animuj()", 1000);
21 </script>
22 </head>
23 <body>
24   <div>
25       <span id="napis">jakis tekst</span>
26   </div>
27 </body>
```

# setInterval-animowany tekst

```
<style>
#napis
{
    position: absolute;
}
</style>
<script >
var dx=3;
function Animuj ()
{
    var element = document.getElementById('napis');

    if (parseInt(element.style.left) < 50 || parseInt(element.style.left) >250)
    {
        dx=-dx;
    }

    element.style.left = (parseInt(element.style.left) +dx)+"px";
}
setInterval("Animuj()", 100);
</script>
</head>
<body>
    <div>
        <span id="napis" style="left: 100px;">jakis tekst</span>
    </div>
</body>
```

# setInterval-animowany tekst -2

```
<style>
#napis
{
    position: absolute;
}
</style>
<script >
var dx=3;
var dz=1;
function Animuj ()
{
    var element = document.getElementById('napis');

    if (parseInt(element.style.left) < 10 || parseInt(element.style.left) >300)
    {
        dx=-dx;
    }
    element.style.left = (parseInt(element.style.left) +dx)+"px";
}
function AnimujTekst ()
{
    var el = document.getElementById('napis');

    if (parseInt(el.style.fontSize) <5 || parseInt(el.style.fontSize) >20)
    {
        dz=-dz;
    }
    el.style.fontSize = (parseInt(el.style.fontSize) +dz)+"px";
}
setInterval ("Animuj () ", 100);
setInterval ("AnimujTekst () ",100);
</script>
</head>
<body>
<div>
    <span id="napis" style="left: 100px; font-size: 10px">jakistekst</span>
</div>
</body>
```

# *clearInterval()*

Metoda `clearInterval()` jest używana w podobny sposób jak metoda `clearTimeout()`.

## Syntax

```
clearInterval(id_of_setinterval)
```

Parameter	Description
<i>id_of_setinterval</i>	Required. The ID of the timer returned by the <code>setInterval()</code> method

**Note:** To be able to use the `clearInterval()` method, you must use a global variable when creating the interval method:

```
myVar = setInterval("javascript function", milliseconds);
```

**`clearInterval(myVar)`**

# Obiekt location

Obiekt location posiada informacje dotyczące aktualnego adresu URL dokumentu oraz metody pozwalające na operowanie tym adresem.

Właściwość obiektu location

Nazwa	Znaczenie
hash	Część adresu URL znajdująca się za znakiem #.
host	Część adresu URL zawierająca nazwę hosta oraz numer portu.
hostname	Część adresu URL zawierająca nazwę hosta.
href	Pełny adres URL.
pathname	Część adresu URL zawierająca ścieżkę dostępu i nazwę pliku.
port	Część adresu URL zawierająca numer portu.
protocol	Część adresu URL zawierająca nazwę protokołu (np. <i>http</i> , <i>ftp</i> ).
search	Część adresu URL znajdująca się za znakiem zapytania.

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <title>setTimeout</title>
    <meta charset="UTF-8" />
  <script >
    document.write( window.location.pathname+"<br>" );
  </script>
  </head>
  <body>
  </body>
</html>
```

/D:/E14/StronyInternetowe/z16\_setTimeout\_setInterval/obiekt\_location\_wlasnosci.html

# Obiekt location - metody

**assign(URL)** – łąduje nową stroną o adresie URL

**reload()** – przeładowanie bieżącej strony

**replace(URL)** – zastąpienie bieżącej strony stroną o adresie URL

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

```
function zmienStrone()
```

```
{
```

```
    window.location.replace("http://helion.pl/ksiazki/jscpk.htm");
```

```
}
```

```
setTimeout("zmienStrone();", 5000);
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<p>Witryna została przeniesiona pod nowy adres:<br />
```

```
    http://helion.pl/ksiazki/jscpk.htm<br />
```

```
    Za 5 sekund zostaniesz przeniesiony do nowej lokalizacji.
```

```
</p>
```

```
</body>
```



# Obiekt window

Obiekt dokument wykorzystuje dostępne metody i właściwości do modyfikacji dokumentu HTML, aktualnie wczytanego przez przeglądarkę.

- **Podstawowe własności**
  - **name** – nazwa okna
  - **defaultStatus** – domyślny tekst pasku stanu przeglądarki
  - **status** – pasek stanu przeglądarki
  - **closed** – wartość logiczna, czy okno zostało zamknięte

# *Obiekt window*

## Window Size

For Internet Explorer, Chrome, Firefox, Opera, and Safari:

- `window.innerHeight` - the inner height of the browser window
- `window.innerWidth` - the inner width of the browser window

For Internet Explorer 8, 7, 6, 5:

- `document.documentElement.clientHeight`
- `document.documentElement.clientWidth`
- or
- `document.body.clientHeight`
- `document.body.clientWidth`

# *Obiekt window*

- **Podstawowe własności**
  - **parent** – okno będące bezpośrednim przodkiem (np. ramka zawierająca inną ramkę)
  - **self, window** – odwołanie do siebie samego
  - **top** – okno najwyższego poziomu w strukturze (np. ramka zawierająca wszystkie inne ramki)
  - **opener** – referencja do obiektu Window, który utworzył okno lub null, jeśli okno utworzył użytkownik

# *Obiekt window*

- **Podstawowe własności**
  - **document** – referencja do obiektu Document
  - **frames[ ]** – tablica obiektów Window, które reprezentują ramki
  - **history** – referencja do obiektu History
  - **location** – referencja do obiektu Location reprezentującego adres URL dokumentu, jego zmiana powoduje załadowanie nowego dokumentu

# Obiekt window

- Podstawowe własności
  - navigator – odwołuje się do obiektu navigator
    - appName – prosta nazwa przeglądarki
    - appVersion – wewnętrzny numer wersji przeglądarki
    - userAgent – zawartość nagłówka HTTP User-Agent
    - appName – nazwa kodowa przeglądarki
    - platform – platforma sprzętowa na której działa przeglądarka

# Obiekt window

- Podstawowe metody
  - `alert(komunikat)` – wyświetla proste okno z komunikatem
  - `confirm(komunikat)` – wyświetla okno z dwoma przyciskami: Ok i Anuluj
  - `prompt(komunikat, domyślna_wartość)` – wyświetla okno w celu pobrania napisu
  - `focus()`, `blur()` – aktywuje, dezaktywuje klawiaturę w oknie

# Obiekt window

- Podstawowe metody
  - `open("okno.html","nazwa","opcje")` – otwiera nowe okno
    - `width=300`
    - `height=300`
    - `status=no`
    - `location=no`
    - `menubar=no`
    - `resizable=no`
    - `scrollbars=no`
    - `titlebar=no`
    - `toolbar=yes`

# Obiekt window

- `close()` – zamyka okno
- `print()` – drukuje, to samo co po wciśnięciu przycisku drukuj w przeglądarce
- `moveBy(x,y)` – przesuwa okno o  $x, y$
- `moveTo(x,y)` – przesuwa okno do  $x, y$
- `resizeBy(x,y)` – zmienia rozmiar o  $x,y$
- `resizeTo(x,y)` – zmienia rozmiar do  $x, y$
- `scrollBy(x,y)` – przesuwa dokument w oknie o  $x,y$
- `scrollTo(x,y)` – przesuwa dokument w oknie do  $x,y$



# Obiekt window

- `setTimeout(wyrażenie/funkcja, milisekundy)` – odracza wykonanie funkcji
- `clearTimeout(TimeoutID)` – anuluje odroczenie i funkcja nie będzie wykonana
- `setInterval(wyrażenie/funkcja, milisekundy)` – wykonuje wyrażenie co określoną liczbę milisekund
- `clearInterval(IntervallID)` – przerywa wykonywanie funkcji

# ***Obiekt window***

