

przydatne strony

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

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

http://www.poradnik-webmastera.com/kursy/javascript/data_i_czas_obiekt_date.php

Style w JS

Javascript udostępnia nam właściwość **element.style**, która zawiera w sobie informacje o stylowaniu dla danego elementu. Może ona zwracać dane wartości stylu, lub je ustawiać.

```
<title>Zmiana stylu elementu</title>
<script >
  function zmienStyl() {
    var zmienionyDiv = document.getElementById('niesformatowanyDIV');
    zmienionyDiv.style.backgroundColor='#33CCCC';
    zmienionyDiv.style.borderColor = '#336699';
    zmienionyDiv.style.borderWidth = '5px';
    zmienionyDiv.style.borderStyle = 'solid';
    zmienionyDiv.style.fontFamily = 'Tahoma, Arial';
    zmienionyDiv.style.lineHeight = '23px';
    zmienionyDiv.style.fontSize = '20px';
    zmienionyDiv.style.color = '#CCFFFF';
    zmienionyDiv.style.width = '200px';
    zmienionyDiv.style.height = '200px';
    zmienionyDiv.style.textAlign = 'center';
```

- w CSS mamy **border-width**, a w Javascript **borderWidth**
- w CSS mamy **background-color**, a w Javascript **backgroundColor**
- w CSS mamy **border-width**, a w Javascript **borderWidth**
- w CSS mamy **font-size**, a w Javascript **fontSize**
- w CSS mamy **font-family**, a w Javascript **fontFamily**
- itp.

```
    zmienionyDiv.innerHTML = 'zostałem sformatowany przez JS'
```

```
  }
</script>
```

```
</head>
```

```
<body>
```

```
  <div id="niesformatowanyDIV" style="background-color : #EEEEEE; width: 200px; height: 100px">
    po kliknięciu nastąpi zmiana koloru elementu div
  </div>
  <input type="button" value="zmien styl elementu div" onclick="zmienStyl()" />
```

Style w JS – skrócony zapis- operator „with”

```
zmienionyDiv.style.backgroundColor='#33CCCC';  
zmienionyDiv.style.borderColor = '#336699';  
zmienionyDiv.style.borderWidth = '5px';  
zmienionyDiv.style.borderStyle = 'solid';  
zmienionyDiv.style.fontFamily = 'Tahoma, Arial';  
zmienionyDiv.style.lineHeight = '23px';  
zmienionyDiv.style.fontSize = '20px';  
zmienionyDiv.style.color = '#CCFFFF';  
zmienionyDiv.style.width = '200px';  
zmienionyDiv.style.height = '200px';  
zmienionyDiv.style.textAlign = 'center';
```

```
with (zmienionyDiv.style) {  
    backgroundColor='#33CCCC';  
    borderColor = '#336699';  
    borderWidth = '5px';  
    borderStyle = 'solid';  
    fontFamily = 'Tahoma, Arial';  
    lineHeight = '23px';  
    fontSize = '20px';  
    color = '#CCFFFF';  
    width = '200px';  
    height = '200px';  
    textAlign = 'center';  
}
```

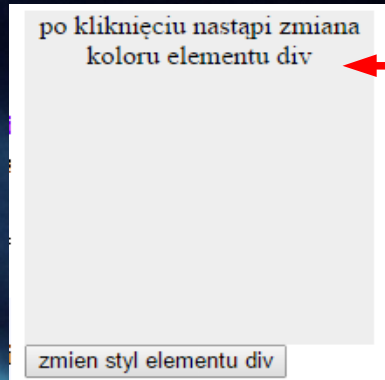
Style w JS – ustalenie klasy- właściwość **className**

```
<head>
<meta charset="utf-8">
  <title>Zmiana stylu elementu</title>
</style>
.nowyDiv{
background-color: #338CCC;
border-color :#336699;
border-width :5px;
border-style :solid;
font-family :'Tahoma, Arial';
line-height :23px;
font-size :20px;
color :#CC22FF;
width :200px;
height :200px;
text-align :center;
}
</style>
<script >
  function zmienStyl () {
var zmienionyDiv = document.getElementById('niesformatowanyDIV');
zmienionyDiv.className="nowyDiv";

zmienionyDiv.innerHTML = 'zostałem sformatowany przez JS'
}
</script>
</head>
```

```
<body>
  <div id="niesformatowanyDIV" style=
"background-color : #EEEEEE; width: 200px; height:
100px">
    po kliknięciu nastąpi zmiana koloru elementu div
  </div>
  <input type="button" value="zmien styl elementu div"
onclick="zmienStyl()" />
</body>
```

Style w JS – ustalenie klasy- właściwość **className** przykład 2



```
<body>  
  <div id="niesformatowanyDIV" class="staryDiv">  
    po kliknięciu nastąpi zmiana koloru elementu div  
  </div>  
  <input id="a1" type="button" value="zmien styl elementu div" />  
  
  <script >  
    function zmienStyl() {  
      var zmienionyDiv = document.getElementById('niesformatowanyDIV');  
      if(zmienionyDiv.className=="staryDiv" ){  
        zmienionyDiv.className="nowyDiv";  
        zmienionyDiv.innerHTML = 'div sformatowany przez JS';  
      }  
      else  
      {  
        zmienionyDiv.className="staryDiv";  
        zmienionyDiv.innerHTML = ' div sformatowany przez JS';  
      }  
    }  
    document.getElementById("a1").addEventListener('click', zmienStyl, false);  
  </script>  
</body>
```

```
<style>  
.nowyDiv{  
background-color: #338CCC;  
border-color :#336699;  
border-width :5px;  
border-style :solid;  
font-family :'Tahoma, Arial';  
line-height :23px;  
font-size :20px;  
color :#CC22FF;  
width :200px;  
height :200px;  
text-align :center;  
}  
.staryDiv{  
background-color:#EEEEEE;  
width :200px;  
height :200px;  
text-align :center;  
}  
</style>
```



Style w JS – ustalenie klasy- właściwość className -przykład- zdarzenia myszy, zaznaczony wiersz

```
<style>
.zaznaczony{
background-color: #CCFF99;
border-color :#99FF33;
border-width :5px;
border-style :solid;
line-height :23px;
font-size :20px;
color :#5555AE;
height :50px;
text-align :center;
}
</style>
```

```
<script>
function zaznacz(element)
{
    element.className='zaznaczony';
}
function odznacz(element)
{
    element.className='odznaczony';
}
</script>
```

Adresy i telefony organizatorów:

Pracownik	Funkcja	Telefon	Email
Małgorzata Nowak	Dyrektor	(12) 222 2222	nowak_m@gmail.com
Marian Kowalski	zastępca dyrektora	(12) 222 2222	kowal@gmail.com
Maria Jaworska	Sędzia	(12) 222 2222	jaworska@gmail.com
Piotr Wawrzecki	Sędzia	(12) 222 2222	wawrzecki_p@gmail.com

Adresy i telefony organizatorów:

Pracownik	Funkcja	Telefon	Email
Małgorzata Nowak	Dyrektor	(12) 222 2222	nowak_m@gmail.com
Marian Kowalski	zastępca dyrektora	(12) 222 2222	kowal@gmail.com
Maria Jaworska	Sędzia	(12) 222 2222	jaworska@gmail.com
Piotr Wawrzecki	Sędzia	(12) 222 2222	wawrzecki_p@gmail.com

```
<table border="1" width="70%">
<caption>Adresy i telefony organizatorów:</caption>
<tr>
<th>Pracownik</th><th>Funkcja</th><th>Telefon</th><th>Email</th>
</tr>
<tr class="odznaczony" onmouseover="zaznacz(this)" onmouseout="odznacz(this)">
<td>Małgorzata Nowak</td><td>Dyrektor</td><td>(12) 222 2222 </td><td><a href="mailto:nowak@gmail.com">nowak_m@gmail.com</a></td>
</tr>
<tr class="odznaczony" onmouseover="zaznacz(this)" onmouseout="odznacz(this)">
<td>Marian Kowalski</td><td>zastępca dyrektora</td><td>(12) 222 2222 </td><td><a href="mailto:kowal@gmail.com">kowal@gmail.com
</a></td>
</tr>
```

Obiekt Date()

Javascript udostępnia nam gotowy obiekt Date() który umożliwia nam w łatwy sposób manipulowanie czasem oraz datami.

Aby wyświetlić na ekranie bieżącą datę jaka jest ustawiona w systemie klienta (a nie na serwerze!) musimy utworzyć nowy obiekt: var czas = new Date() - konstruktor bez parametrów zwróci bieżącą datę i czas

```
var czas = new Date(); //utworzenie nowego obiektu Date  
document.write(czas); //wypisanie na ekran naszego obiektu
```

```
<!DOCTYPE html>  
<html>  
<body >  
<script >  
    var czas = new Date();  
    document.write(czas);  
</script>  
</body>  
</html>
```

```
Fri Jan 02 2015 16:25:57 GMT+0100 (Środkowoeuropejski czas stand.)
```

Obiekt Date() - inne sposoby wypisania daty

toString() – zamienia datę na łańcuch znaków, który zwracany jest jako wynik.

```
<!DOCTYPE html>
<html>
<body >
  <div id="test1"></div>

  <script >
    var czas = new Date();
    var element = document.getElementById('test1');
    var str=czas.toString();
    element.innerHTML =str;
  </script>
</body>
</html>
```

Fri Jan 02 2015 16:14:48 GMT+0100 (Środkowoeuropejski czas stand.)

toLocaleString()	Zwraca aktualną datę i godzinę (czas lokalny) jako łańcuch znaków zapisanych w formacie: "Friday, June 03, 1994 13:25:12".
UTC(rok, miesiąc, dzień, godziny,	Zwraca liczbę milisekund, które upłynęły od godziny 00:00:00 dnia 1 stycznia 1970 roku do podanej daty (czas uniwersalny), np. wywołanie Date.UTC(96,1,1,0,0,0) zwróci liczbę 849398400000.

Obiekt Date()- metody get*

Metoda	Opis
getDate()	Zwraca dzień miesiąca jako liczbę całkowitą z zakresu od 1 do 31.
getDay()	Zwraca dzień tygodnia jako liczbę całkowitą taką, że 0 oznacza niedzielę, 1 - poniedziałek itd.
getHours()	Zwraca godzinę jako liczbę całkowitą z zakresu od 0 do 23.
getMinutes()	Zwraca minuty jako liczbę całkowitą z zakresu od 0 do 59.
getMonth()	Zwraca miesiąc jako liczbę całkowitą z zakresu od 0 do 11, gdzie 0 oznacza styczeń, a 11 - grudzień.
getSeconds()	Zwraca sekundy jako liczbę całkowitą od 0 do 59.
getTime()	Zwraca liczbę milisekund, które upłynęły od godziny 00:00:00 dnia 1 stycznia 1970 roku (czas uniwersalny -GMT).
getTimeZoneOffset()	Zwraca różnicę czasu między czasem lokalnym a czasem Greenwich (GMT) wyrażoną w minutach.
getFullYear()	Zwraca liczbę lat, które upłynęły od roku 1900. Jeśli jest rok 1999 to zwrócona zostanie liczba 99, jeśli będzie 2020 to zwrócona będzie liczba 120
parse(data)	Konwertuje łańcuch znaków podany jako data na liczbę milisekund, które upłynęły od godziny 00:00:00 dnia 1 stycznia 1970 roku do podanego dnia (np. wywołanie Date.parse("May 23,1996") zwróci liczbę
getFullYear()	- zwraca pełną liczbę reprezentującą rok (np. 1999 lub 2000)
getMillisecond()	- zwraca milisekundy (wartość z przedziału 0 - 999)

Obiekt Date()- metody get*

```
<!DOCTYPE html>
<html>
<body >
  <div id="test1"></div>

<script >
  var czas = new Date();
  var element = document.getElementById('test1');
  element.innerHTML = "Jest teraz: " + czas.getHours() + ":" + czas.getMinutes() + ":" + czas.getSeconds()
    + "<br>";
  element.innerHTML += "Dnia: " + czas.getDate() + "." + (czas.getMonth()+1) + "." + czas.getFullYear();
</script>

</body>
</html>
```

Obiekt `Date()`- metody `get*`

Przykładowe wyświetlenie godziny może mieć postać:

```
1 | var czas = new Date();  
2 | document.write(czas.getHours())  
3 | //lub  
4 | document.write((new Date).getHours());  
5 |
```

Deklaracja `var dzisiaj=new Date()` tworzy nową zmienną systemowego obiektu `Date`.

Deklaracja `var godzina=dzisiaj.getHours()` powoduje utworzenie nowej zmiennej - `godzina` i przypisanie jej aktualnej godziny.

Obiekt *Date()*- metody *set**

<code>setDate(liczba)</code>	Ustawia dzień miesiąca dla aktualnego obiektu <i>Date</i> , parametr liczba jest liczbą całkowitą z zakresu od 1 do 31
<code>setHours(liczba)</code>	Ustawia godzinę dla aktualnego obiektu <i>Date</i> , parametr liczba jest liczbą całkowitą z zakresu od 0 do 23.
<code>setMinutes(liczba)</code>	Ustawia minuty dla aktualnego obiektu <i>Date</i> , parametr liczba jest liczbą całkowitą z zakresu od 0 do 59.
<code>setMonth(liczba)</code>	Ustawia miesiąc dla aktualnego obiektu <i>Date</i> , parametr liczba jest liczbą całkowitą z zakresu od 0 do 11 (gdzie 0 oznacza styczeń, 1 - luty itd.).
<code>setSeconds(liczba)</code>	Ustawia sekundy dla aktualnego obiektu <i>Date</i> , parametr liczba jest liczbą całkowitą z zakresu od 0 do 59.
<code>setTime(liczba)</code>	Ustawia datę oraz godzinę dla aktualnego obiektu <i>Date</i> , parametr liczba określa liczbę milisekund, które upłynęły od godziny 00:00:00
<code>setYear(rok)</code>	Określa rok dla określonego obiektu <i>Date</i> .

`setMilliseconds()` - ustawia milisekundy.

Ustawianie daty i czasu

```
<!DOCTYPE html>
<html>
<body >

<div id="test7"></div>

<script type="text/javascript">
    var time = new Date();
    time.setYear(2014);
    time.setMonth(4);
    time.setDate(12);
    time.setHours(15);
    time.setMinutes(24);
    time.setSeconds(18);
    document.getElementById('test7').innerHTML = time.getFullYear() + "." + time.getMonth() + "." +
    time.getDate() + " - " + time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds();
</script>
</body>
</html>
```

2014.4.12 - 15:24:18

Ustawianie daty i czasu

Można także stworzyć obiekt Date który będzie zawierał ściśle określoną wartość daty i czasu - do tego celu należy użyć konstruktora który może przyjąć od dwóch (rok, miesiąc) do siedmiu parametrów (rok, miesiąc, dzień, godzina, minuty, sekundy, milisekundy). Jeżeli poda się mniej niż siedem parametrów, to funkcja użyje wartości domyślnych dla parametrów niepodanych (jeden dla dnia i zero dla pozostałych wartości). Ważne jest także aby numer miesiąca był liczbą z przedziału 0-11 (czyli trzeba "normalny" numer miesiąca pomniejszyć o jeden).

```
<body >
<div id="test7"></div>
<div id="test8"></div>
<script type="text/javascript">
  var time = new Date(2014, 0, 3,4);
  document.getElementById('test7').innerHTML = time.getFullYear() + "." + time.getMonth() + "." +
  time.getDate() + " - " + time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds();
  document.getElementById('test8').innerHTML =time.toString();
</script>
</body>
```

2014.0.3 - 4:0:0

Fri Jan 03 2014 04:00:00 GMT+0100 (Środkowoeuropejski czas stand.)