

## Kaskadowe arkusze stylów CSS

### Geneza

- oddzielenie struktury dokumentu HTML od reguł prezentacji
- poszerzenie samego HTML

### Korzyści

- przejrzystość dokumentów
- łatwe zarządzanie stylem (wyglądem) serwisu
- oszczędność łącz
- uproszczona aktualizacja
- większe możliwości (szeroki zasób środków)

## Wprowadzenie arkusza CSS

### Arkusze zewnętrzne

Zewnętrzny arkusz to plik tekstowy z rozszerzeniem "css".

Arkusz taki możemy podłączyć do wielu plików HTML.

Dzięki arkuszom zewnętrznym uzyskujemy centralne sterowanie wyglądem serwisu.

### Wykorzystanie arkusza na stronie

#### Krok 1 - Utworzenie arkusza

Zewnętrzny arkusz stylów to plik tekstowy z rozszerzeniem css.

#### Krok 2 – Podłączenie arkusza do strony

```
<head>
...
<link href="ściezka_do_pliku.css" rel="stylesheet" type="text/css">
...
</head>
```

```
<head>
...
<link href="style/arkusz.css" rel="stylesheet" type="text/css">
...
</head>
```

#### UWAGA !

Zewnętrzny arkusz jest najwygodniejszym i najczęściej stosowanym sposobem wprowadzania CSS dla serwisu.  
Jeden arkusz dla kilku plików pozwala na prostą modyfikację wyglądu witryny.

### Arkusze osadzone

Arkusz osadzony umieszczany jest w sekcji head dokumentu HTML.

Zasięg takiego arkusza to pojedynczy dokument HTML.

#### Wykorzystanie arkusza na stronie

<head>

...

<style type="text/css">

arkusz stylów

</style>

...

</head>

### Style wpisane do znacznika (inline)

Zasięg stylu ograniczony do jednego elementu HTML.

Styl wprowadzany przy użyciu atrybutu style.

#### Wykorzystanie stylu na stronie

<znacznik\_html style="definicja stylu">....</znacznik\_html>

<p style="font-family: verdana; color: red;"> Czerwona Verdana </p>

<body style="margin-left: 100px;"> ... </body>

## Kaskada stylów (Cascading Style Sheets)

W dokumencie HTML możemy wykorzystać jednocześnie różne metody wprowadzenia stylu.

Hierarchia arkuszy:

- styl lokalny (inline)
- styl osadzony
- styl zewnętrzny

Warto korzystać z kaskady !!!

## Budowa arkusz stylów

### Definicja stylu

```
nazwa_definicji
{
    cecha_1: wartość;
    cecha_2: wartość;
    ...
}
```

### Selektor prosty

#### Własności:

- nazwą definicji jest tu nazwa elementu HTML
- styl jest stosowany automatycznie do wszystkich elementów danego typu na stronie

```
nazwa_znacza_html
{
    cecha_1: wartość;
    cecha_2: wartość;
    ...
}
```

```
td
{
    font-family: verdana, arial;
    font-size: 8pt;
    background-color: yellow;
}
```

```
body
{
    background-color: green;
}
```

```
h1, h2, h3
{
    color: red;
}
```

**Definicje zawarte w selektorach prostych powinny być uniwersalne !!!**

## Selektor klasy

### Własności:

- definicji możemy nadać dowolną nazwę
- nazwę klasy poprzedzamy kropką
- styl jest stosowany wyłącznie do elementów które wskażemy

**Krok 1** – definicja klasy (kod **CSS**)

```
.dowolna_nazwa_klasy  
{  
    cecha_1: wartość;  
    cecha_2: wartość;  
    ...  
}
```

**Krok 2** – podłączenie stylu (kod **HTML**)

**<znacznik\_html class="dowolna\_nazwa\_klasy">... </znacznik\_html>**

Klasa uniwersalna (może być zastosowana do różnych elementów)

```
.niebieska  
{  
    color: blue;  
}  
  
<td class="niebieska">Tekst niebieski</td>  
  
<p class="niebieska">Tekst niebieski</p>
```

Klasa dla konkretnego typu elementu

```
td.odwrocona  
{  
    background-color: black;  
    color: white;  
}  
  
p.odwrocona  
{  
    background-color: blue;  
    color: yellow;  
}  
  
<td class="odwrocona"> czarne tło, biały tekst </td>  
  
<p class="odwrocona"> niebieskie tło, żółty tekst</p>
```

## Selektor identyfikatora

### Własności:

- selektor może być użyty dla wszystkich elementów HTML mających określony atrybut id
- atrybut id ma unikalną wartość w obrębie dokumentu

**Krok 1** – definicja klasy (kod **CSS**)

```
znacznik_html#id_elementu
{
    cecha_1: wartość;
    cecha_2: wartość;
    ...
}
```

**Krok 2** – podłączenie stylu (kod **HTML**)

**<znacznik\_html id="id\_elementu">...</znacznik\_html>**

```
td#naglowek
{
    font-weight: bold;
}

<td id="naglowek"> Tekst pogrubiony </td>
```

## Pseudoklasy (dla odsyłaczy)

### Własności:

- pseudoklasy najczęściej używane są dla odsyłaczy
- pozwalają one zdefiniować styl odnośników w zależności od warunków zewnętrznych

### Typy pseudoklas:

**a:link** (standardowy wygląd łącza)  
**a:hover** (wygląd łącza po najechaniu myszą)  
**a:visited** (odwiedzone hiperłącze)  
**a:active** (aktywne)

Standardowo łącza bez podkreślenia, po najechaniu myszą podkreślone

```
a
{
    text-decoration: none;
}
a:hover
{
    text-decoration: underline;
}
```

Przykładowe właściwości CSS

Parametry czcionki	
<b>krój</b>	font-family: verdana, arial;
<b>wielkość</b>	font-size: 12pt;
<b>styl</b>	font-style: italic;
<b>grubość</b>	font-weight: bold;

Parametry tekstu	
<b>dekoracja</b>	text-decoration: none/underline/overline/line-through;
<b>odstęp</b>	line-height: px/pt; (między wierszami) word-spacing: px/pt; (między wyrazami) letter-spacing: px/pt; (między literami)

Kolory i tła dla elementów posiadających daną cechę	
<b>kolor</b>	color: kolor/#rrggbb;
<b>kolor tła</b>	background-color: kolor/#rrggbb;
<b>obraz jako tło</b>	background-image: url(nazwa_pliku.jpg);

Obramowanie (top/bottom/left/right)	
<b>kolor</b>	border-color: kolor/#rrggbb;
<b>rodzaj</b>	border-style: solid/dotted/dashed/double/groove;
<b>szerokość</b>	border-width: px/pt;

Rozmiar (dla elementów prostokątnych)	
<b>szerokość</b>	width: px/%;
<b>wysokość</b>	height: px/%;

Wyrównanie	
<b>poziom</b>	text-align: left/right/center/justify;
<b>pion</b>	vertical-align: top/middle/bottom;

Marginesy (top/bottom/left/right)	
<b>zewnątrzny</b>	margin: px/%;
<b>wewnętrzny</b>	padding: px/%;