

Kaskadowe Arkusze Stylów

Kaskadowość to zasady, które określają w jaki sposób przeglądarka wybiera właściwość CSS dla danego elementu.

BRAK ZADEKLAROWANYCH WŁAŚCIWOŚCI

Właściwości się inicjalizowane jeśli arkusz stylów (nasz, czy przeglądarki) nie zawiera danej właściwości (czy to w precyzyjnej regule czy przez dziedziczenie).

Domyślnym kolorem dla fontów będzie czarny. Rodzina fontów będzie ustawione na "serif" (w praktyce "times new roman"), font-size na medium (w praktyce 16px), a margin na 0 (dla każdego kierunku).

Większość właściwości jest inicjalizowanych, bo w praktyce nigdy nie deklarujemy wszystkich właściwości (nawet części).

DZIEDZICZENIE

Część właściwości jest dziedziczona. Na przykład nadanie koloru elementowi <html> (jako elementowi głównemu) sprawi, że wszyscy jego potomkowie będą dziedziczyć ten kolor (o ile oczywiście nie mają precyzyjnych reguł określających daną właściwość).

```
html {  
    color: red;  
}
```

```
<h1> Tytuł</h1>  
<p>tekst</p>
```

```
<!-- będą miały czerwony kolor tekstu -->
```

ELEMENT DZIEDZICZY PO RODZICU

```
<body>
  <header>
    <div>
      <p></p>
    </div>
  </header>
</body>
```

```
html header {font-size: 20px;}
div {font-size: 15px}
```

<p> będzie miało 15px, ponieważ dziedziczy po rodzicu (niezależnie od tego jak ważny jest selektor).

Przy ustaleniu dziedziczenia **ważność nie ma znaczenia.**

Decyduje element, który jest bliżej ostatecznego elementu.

```
<header class="top">
  <div>
    <p>Tekst</p>
  </div>
</header>
```

```
html header.top {color: red;}
div {color: blue;}
```

/ Element p będzie niebieski, bo dziedziczy bezpośrednio po rodzicu. Dziedziczenie zachodzi bo element p nie jest wskazany w żadnym selektorze precyzyjnie */*

A gdy rodzic nie ma ustawionej wartości?

Decyduje jego rodzic.

```
<header class="top">
  <div>
    <p>
      <span>tekst</span>
    </p>
  </div>
</header>
```

```
html header.top {color: red;}
div {color: blue;}
```

/* By ustalić wartość rodzica, który nie ma ustawione precyzyjne wartości, przeglądarka też przypisuje mu wartość dziedziczoną. W tym wypadku **p** dziedziczy po **div** kolor blue, a potem **span** dziedziczy kolor blue po **p** */

PRECYZJA NADPISUJE DZIEDZICZENIE

Jeśli selektor jest precyzyjny, to jego właściwości zawsze nadpisują właściwości odziedziczone.

```
html body div {font-size: 20px; color: #333;}  
p {font-size: 10px;}
```

```
<div class="large">  
  <p>tekst</p> /* p ma kolor #333 (dziedziczy po rodzicu) i  
                font-size 10px, bo precyzja nadpisuje  
                dziedziczenie */  
</div>
```

Kilka reguł pasuje - co wtedy?

```
<div class="blue">  
  <p class="red">Tekst</p> /*jaki kolor ?*/  
</div>
```

```
body div p {color: green}  
div>p {color: yellow}  
div.blue p {color: blue}  
p.red {color: red}
```


Klasyfikacja medalowa - najlepszy sposób

medal brązowy: selektory odwołujące się do elementów (znaczników) i pseudoelementów np. `div p {}` to dwa brązowe medale. Brązowy medal to także selektor atrybutu (nawiasy kwadratowe `[]`)

medal srebrny: selektor odwołujące się do klas i pseudoklas np. `.red {}` to srebrny medal. `.red p.big {}` to dwa medale srebrne i jeden brązowy.

medal złoty: selektor odwołujący się do identyfikatora np. `#header` to jeden złoty medal.

* - selektor gwiazdki (każdy element) nie jest liczony

Klasyfikacja medalowa

W klasyfikacji olimpijskiej (a takie zasady należy przyjąć) jeden srebrny medal jest warty więcej niż wszystkie brązowe, a jeden złoty więcej niż wszystkie brązowe i srebrne razem wzięte.

```
html body header div p /* 5 medali brązowych */
```

```
div.blue p /*dwa medale brązowe i jeden srebrny */
```

Druga reguła jest ważniejsza

Klasyfikacja medalowa

Do klasyfikacji nie bierze się kombinatorów (a więc "spacja", "+" czy ">" nie mają znaczenia). Pamiętajmy też, że psuedolementy są liczone tak samo jak elementy, a pseudoklasy jak klasy.

```
a:hover /* jeden medal srebrny i jeden brązowy */
```

```
input.email:focus /* dwa srebrne i jeden brązowy */
```

```
p::before /* dwa brązowe */
```

Jeśli kilka reguła pasuje do tego samego elementu i mają taką samą wagę, to **decyduje kolejność**. **Nowsza wartość nadpisuje starszą**.

```
<header>
  <div class="main">
    <div class="article">
      <p>Tekst</p> <!-- będzie czerwony -->
    </div>
  </div>
</header>
```

```
header .article>p { color: blue;}
div.main p {color: red;}
```