

Temat: Selektory, właściwości oraz wartości w CSS.

3.2 Selektory, właściwości oraz wartości w CSS

ZAGADNIENIA

- Rodzaje selektorów
- Podstawowe właściwości i ich wartości
- Przykłady zastosowań właściwości CSS

Selektory

Selektory CSS mogą się odnosić do elementów HTML, takich jak: konkretne znaczniki, identyfikatory (id), klasy (class) oraz inne atrybuty.

Jeżeli selektorem jest dowolny znacznik języka HTML, to oznacza, że mamy do czynienia z tzw. **selektorem prostym**. Formatowanie znaczników polega na definiowaniu selektorów za pomocą ich nazw.

PRZYKŁAD 3.6.

```
h2 {
  color: red;
}
```

Możemy również grupować selektory znaczników, które mają być identycznie formatowane.

PRZYKŁAD 3.7.

```
h1, p, i {
  color: red;
}
```

Do identyfikacji poszczególnych elementów używamy w języku HTML identyfikatorów, zapisywanych skrótem id. Szczególnie często stosujemy je w definiowaniu elementów blokowych `<div>`. Każdy identyfikator ma unikatową nazwę w obszarze jednego dokumentu. Nazwy tej można użyć jako selektora do formatowania elementu, który ona identyfikuje. Należy pamiętać, że zapis selektora identyfikatora jest poprzedzony zawsze znakiem #.

PRZYKŁAD 3.8.

Dla elementu `<div id="stopka"> </div>` definiowanie formatowania w arkuszu stylów CSS wygląda następująco:

```
#stopka {
```

```
  text-align: center;
  color: green;
}
```

Inną metodą definiowania selektora jest użycie klasy, która również służy do rozróżniania elementów języka HTML. Klasy definiujemy za pomocą atrybutu `class`. Selektor klasy odszukuje każdy element w dokumencie HTML, który ma ten atrybut o takiej samej nazwie. Ponieważ atrybutu klasy możemy używać wielokrotnie w obrębie jednego dokumentu, to jeśli używamy go jako selektora, możemy formatować wiele elementów jednocześnie. W pliku css selektor klasy definiujemy, poprzedzając nazwę kropką.

PRZYKŁAD 3.9.

Dla elementu `<div class="stopka"> </div>` definiowanie formatowania w arkuszu stylów CSS wygląda następująco:

```
.stopka {
  text-align: center;
  color: red;
}
```

Jeżeli chcemy sformatować część takich samych znaczników na stronie internetowej, możemy również użyć atrybutu `class`. Pozwoli to na zróżnicowanie formatowania takich samych znaczników.

PRZYKŁAD 3.10.

Dla elementu `<p class="inny"> Tekst </p>` definiowanie formatowania w arkuszu stylów CSS wygląda następująco:

```
p.inny {
  color: red;
}
```

W CSS stosujemy również **selektory uniwersalne**, oznaczone znakiem *. Mają one taką samą ważność jak definiowany styl elementu najwyższego poziomu, czyli **body**.

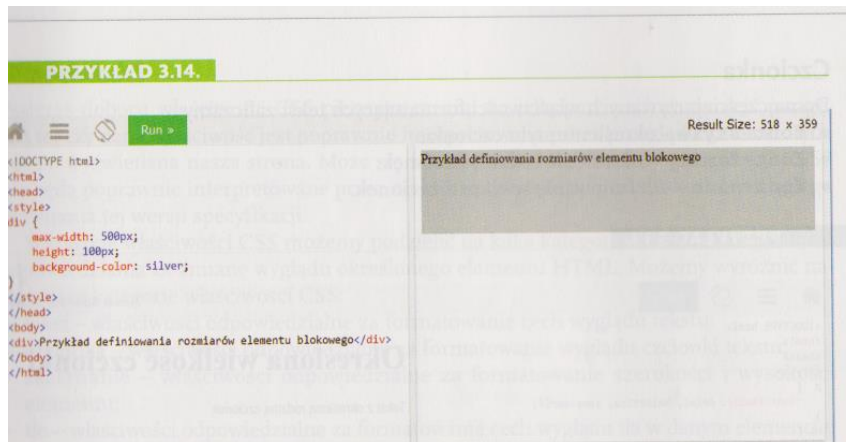
Hierarchiczny układ elementów HTML pozwala również na stosowanie selektorów zależnych od elementów umieszczonych wyżej w hierarchii, na tym samym poziomie lub niżej. Mówimy wtedy o **selektorach potomka** (ang. *descendant selector*), **selektorach dziecka** (ang. *child selector*) lub **selektorach braci** (ang. *adjacent sibling selector*).

PRZYKŁAD 3.11.

```
ul li {color:red; font-size:20px;}
```

Tego typu zapis wykorzystuje selektor potomka. Oznacza to, że elementy wykazu na drugim poziomie zagnieżdżenia będą zapisane kolorem czerwonym oraz czcionką o wielkości 20 px, podczas gdy elementy wykazu pierwszego poziomu będą miały domyślny kolor i rozmiar czcionki.

Omówienie lekcji
podczas zajęć live
teams , środa godz.
9.00



Rys. 3.10. Przykład formatowania rozmiaru obiektu

Tło

Do najczęściej używanych właściwości formatujących tło zaliczamy:

- **background-color** – określenie koloru tła;
- **background-image** – wstawienie obrazu strony jako tła;
- **background-position** – ustawienie grafiki tła w określonej pozycji.

PRZYKŁAD 3.15.

```

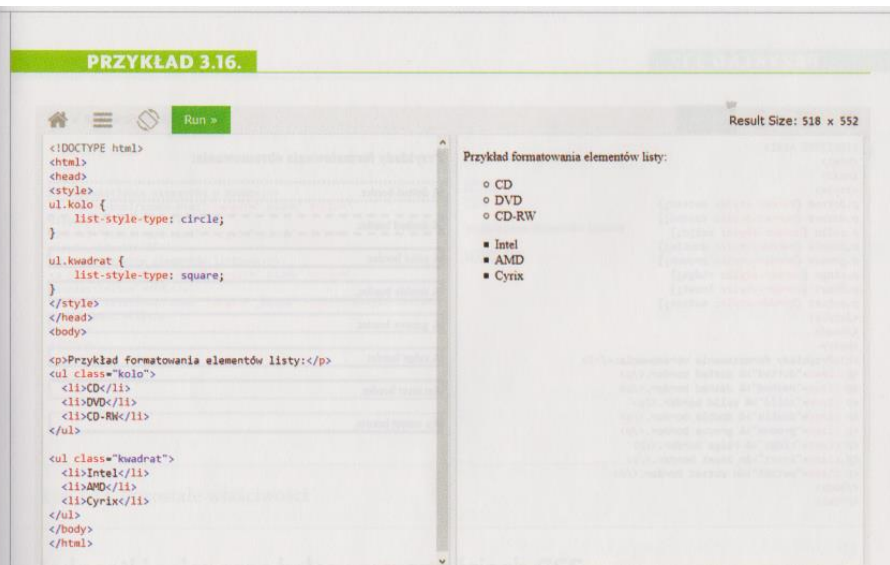
body {
  background-image: url("/grafika/tlo.jpg");
}
  
```

Zastosowanie takiego stylu spowoduje umieszczenie obrazka z pliku tlo.jpg jako tła dla całej strony.

Wykazy (listy)

Do najczęściej używanych właściwości formatujących listy zaliczamy:

- **list-style-type** – określenie graficznych punktów listy;
- **list-style-image** – zdefiniowanie pliku graficznego jako punktora.



Rys. 3.11. Formatowanie listy w CSS

Obramowanie

Do najczęściej używanych właściwości formatujących obramowanie elementów zaliczamy:

- **border-style** – określa styl obramowania;
- **border-color** – określa kolor obramowania;
- **border-radius** – definiuje zakres zaokrąglenia krawędzi obramowania;
- **border-image** – definiuje graficzny styl obramowania.

PRZYKŁAD 3.17.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
</style>
</head>
<body>
<h3>Przykłady formatowania obramowania:</h3>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
</body>
</html>
```

Rys. 3.12. Przykłady obramowania w CSS

Tabele

Do najczęściej używanych właściwości formatujących tabele zaliczamy:

- **caption-side** – określa położenie tytułu tabeli;
- **border-spacing** – definiuje odstępy między komórkami.

Pozostałe właściwości

Oprócz wymienionych możemy spotkać właściwości ułatwiające wyświetlanie elementu, formatowanie wypływania treści kontenera, oblewanie tekstem elementu czy właściwości definiujące rodzaju kursora.

Właściwość display reguluje sposób wyświetlania elementów – tworzy z nich, na żądanie autora strony, elementy blokowe, liniowe czy punkty wykazu.

Właściwość overflow pozwala zarządzać przepełnieniem treści bloku czy kontenera.

Float umożliwia zdefiniowanie sposobu otaczania elementów tekstem. Natomiast właściwość cursor określa typ kursora myszki, który ma obowiązywać na danym obszarze.

! UWAGA

Zestawienie najpopularniejszych właściwości i ich wartości specyfikacji CSS3 zamieszczono jako załączniki, na końcu podręcznika.

PRZYKŁAD 3.18.

```
<!DOCTYPE html>
<html>
<body>
<p>wyświetlanie elementów w bloku</p>
<a href="/html/index.html" target="_blank" style="display:block">HTML</a>
<a href="/css/index.html" target="_blank" style="display:block">CSS</a>
<p>wyświetlanie elementów liniowo</p>
<a href="/html/index.html" target="_blank" style="display:inline">HTML</a>
<a href="/css/index.html" target="_blank" style="display:inline; cursor:wait">CSS</a>
</body>
</html>
```

Rys. 3.13. Pozostałe właściwości

Jednostki miar oraz kolory w wartościach CSS

Kaskadowe arkusze stylów określają cechy wielu elementów. Stosowane są przy tym różne jednostki miar. Możemy podzielić je na jednostki względne i bezwzględne.

Do jednostek miar względnych zaliczamy:

- **px** – piksele;
- **em** – proporcje elementu względem wielkości czcionki;
- **ex** – proporcję względem wysokości x danej czcionki;
- **%** – procenty.

PRZYKŁAD 3.19.

```
p{
font-size:10%;
margin-left:20px;
}
```

Jednostki miar bezwzględnych to:

- **in** – cal;
- **pt** – punkt (jednostka wywodząca się z typografii; 72 pt = 1 in);
- **cm** – centymetr;
- **mm** – milimetr;
- **pc** – pica (jednostka wywodząca się z typografii; 1 pc = 12 pt).

PRZYKŁAD 3.20.

```
p{
font-size:12pt;
margin-left:2in;
}
```

Dodatkowo w tworzeniu kaskadowych arkuszy stylów używa się jednostek miary kątów oraz jednostek czasu:

- **ms** – milisekunda;
- **s** – sekunda;
- **deg** – stopień;
- **rad** – radian.

PRZYKŁAD 3.21.

```
h5 {  
  pause-after: 10s;  
}
```

W wartościach często podajemy nazwy kolorów za pomocą kilku dostępnych metod. Kodowanie kolorów w CSS możemy wykonać przez:

- podanie nazwy koloru,
`h3 {color:red;}`
- zapis za pomocą modelu RGB z wykorzystaniem wartości liczbowych,
`#stopka {color:rgb(255,0,0);}`
- zapis za pomocą modelu RGB z wykorzystaniem wartości procentowych,
`p {color:rgb(100%,0%,0%);}`
- zapis za pomocą modelu RGBA,
`#menu {background-color: rgba(255, 0, 0, 0.3);}`
- zapis za pomocą modelu HLS,
`.menu {background-color: hsl(120, 100%, 50%);}`
- zapis za pomocą modelu HSLA,
`p.stopka {background-color: hsla(120, 100%, 25%, 0.3);}`
- zapis korzystający z systemu szesnastkowego,
`p {color:#FF0000;}`

Dodatkowo możemy określić stopień przezroczystości właściwością `opacity` – stosujemy wartości od 0.0 (pełna przezroczystość) do 1.0 (brak przezroczystości).

```
p {background-color:rgb(255,0,0);opacity:0.7;}
```

SPRAWDŹ SWOJĄ WIEDZĘ

1. Scharakteryzuj poszczególne rodzaje selektorów.
2. Wymień najczęściej używane właściwości CSS.
3. W jaki sposób można przeprowadzić kodowanie kolorów w CSS?