CSS – kaskadowe arkusze stylów

CSS

- CSS – Cascading Style Sheets – Kaskadowe Arkusze Stylów
- CSS jest językiem definiującym wygląd dokumentu HTML
- CSS został stworzony przez World Wide Web Consortium (W3C) w odpowiedzi na coraz bardziej rozbudowane strony internetowe, a co za tym idzie, coraz większe trudności w zarządzaniu kodem

CSS

- Pierwsza wersja CSS1 – 1996 r.
- CSS2.1 – ostateczna publikacja jako standard w 2011 r. (pierwsza w 1998 r. jako CSS 2)
- CSS3 – specyfikację podzielono na moduły, które są publikowane oddzielnie (pierwszy draft w 1999 r., ostatnie publikacje w 2014 r.)
- CSS4 – podobnie jak CSS3 jest podzielony na moduły (pierwsze publikacje od 2007 r., ostatnia 2015 r.)

CSS

- Wsparcie przeglądarek:
  - wszystkie najnowsze przeglądarki przechodzą test Acid3 (http://acid3.acidtests.org/) z notą 100/100
  - pomimo tego niektóre właściwości mogą nie być wspierane przez konkretne przeglądarki, lub wymagają zastosowania, tzw. CSS hacks lub CSS filters (aktualnie najczęściej stosuje się przedrostki, np. `-moz-linear-gradient` dla Firefoxa)

CSS

- Zalety:
  - rozdzielenie warstwy prezentacji od warstwy danych => przejrzystość kodu
  - zmiany wprowadza się w jednym miejscu => oszczędność czasu i pieniędzy
  - zmniejszenie ilości przesyłanych danych (plik CSS zazwyczaj jest wczytywany z pamięci podręcznej przeglądarki jeśli strona była już wcześniej odwiedzana)
  - poprawa pozycjonowania elementów na stronie względem stosowanego przed CSS layoutu opartego na tabelach

CSS

- Wady (a raczej braki):
  - brak możliwości sformatowania przodka/rodzica elementu
  - brak możliwość zarządzania zachowaniem pseudoklas (takich jak `:hover`) po stronie klienta– nie można ich wyłączyć, co stwarza ryzyko nadużyć
  - brak możliwość nadawania nazw regułom (jeśli selektor się zmieni, reguła przestaje go obejmować)
  - brak możliwości odwołania się do dowolnego fragmentu tekstu bez zastosowania znacznika (poza `:first-letter` i `:first-line`)

Definiowanie stylów

- 3 sposoby definiowania stylów
  - bezpośrednio wewnątrz znacznika
    ```
    <p style="color: red;">Akapit o czerwonym kolorze</p>
    ```
  - w treści dokumentu HTML w znacznikach `<style></style>`
    ```
    <style>
    p.green {color: green;}
    </style>
    ```

Definiowanie stylów

- w zewnętrznym pliku (i to jest generalnie standard)

```
<html>
<head>

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

</head>
</html>
```

Kaskada stylów

- Dlaczego KASKADOWE arkusze stylów?
  - Style "rozpływają się" po dokumencie HTML – dziecko/potomek dziedziczy po rodzicu/przodku (czyli element znajdujący się w innym elemencie, dziedziczy jego właściwości)

Kaskada stylów

– Mają swój priorytet – hierarchię:

1. Domyślny arkusz przeglądarki WWW
2. Domyślny arkusz użytkownika przeglądarki
3. Zewnętrzne arkusze stylów i definicje stylów w sekcji body dokumentu
4. Definicje stylów w atrybucie `style` elementu

(im styl jest bliżej formatowanego elementu, tym wyższa jest jego pozycja w hierarchii)

Kaskada stylów

- Aby podnieść rangę reguły o 1 poziom, należy dodać na jej końcu wyrażenie `!important`

```
.green {
    color: green !important;
}
```

Kaskada stylów

- Wyższy priorytet nad stylem ogólnym będzie miał również styl dla konkretnego selektora:

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

będzie niżej w hierarchii od

```
p.blue {color: blue;}
```

Kaskada stylów

- Style mogą się na siebie nakładać i kumulować

Jeśli style definiują wygląd różnych właściwości danego selektora, to wszystkie właściwości zostaną zmienione.

Kaskada stylów

Przykład:

```
.green { color: green;}
.bold {font-weight: bold;}
.uppercase {text-transform: uppercase;}

<p class="green bold uppercase">test</p>
```

będzie wyglądać tak: **TEST**

Tworzenie reguł stylów

- Konstrukcja stylu CSS

```
selektor {
        właściwość: wartość;
        właściwość: wartość;
        …
        właściwość: wartość;
}
```

Tworzenie reguł stylów

- Selektorem może być:
  - element HTML, np. `body, p, a, div, form, td, li`

```
body {…;}
p {…;}
div, h3, a {…;}
```

Tworzenie reguł stylów

– element HTML posiadający określony atrybut, np.

```
<p class="red">, <div id="username">,
<input name="password" …>
.red {…;}
#username {…;}
input[name="password"]{…;}
```

Tworzenie reguł stylów

– selektor uniwersalny *, który odpowiada wszystkim elementom dokumentu

```
* {color: pink;}
```

– kombinacja powyższych

```
table.pricelist#discounts {
   font-weight: bold;
}
```

Tworzenie reguł stylów

– pseudoklasa, czyli element powiązany z selektorem, ale nie określony precyzyjnie

```
p:first-letter {
   font-weight: bold;
   color: red;}
div:first-line {
   font-style: italic;}
h1:hover {
   text-transform: uppercase;
   color: violet}
```

Tworzenie reguł stylów

Pseudoklasy formatujące łącza:

Pseudoklasa	Co formatuje
`:link`	Nieodwiedzone łącze
`:visited`	Odwiedzone łącze
`:hover`	Łącze, nad którym znajduje się kursor
`:active`	Aktywne łącze (w momencie kliknięcia)
`:focus`	Aktualnie wykorzystywane łącze (ostatnio kliknięte)

Tworzenie reguł stylów

Wszystkie selektory dostępne są na stronie internetowej W3C, pod adresem:

http://www.w3schools.com/cssref/css_selectors.asp

Tworzenie reguł stylów

- Właściwością może być, np. kolor czcionki (`color`), lewy margines (`margin-left`), zaokrąglenie rogów (`border-radius`), obramowanie (`border`), kolor tła (`background-color`), wielkość czcionki (`font-size`), szerokość (`width`) i wiele innych

Tworzenie reguł stylów

- Wartość, w zależności od właściwości, może być podawana
  - jako słowo kluczowe, np.

```
p {color: red;}
div.right {float: right;}
#login-link {text-decoration: none;}
```

Tworzenie reguł stylów

– w różnych jednostkach miary

```
.big {font-size: 24px;}
body {width: 80%;}
.red-bg {background-color: #9fadc0;}
table.border5 {border-width: 5pt;}
```

Jednostki miary: px, pt, pc, mm, cm, in, %, em, ex, #RRGGBB, rgb(r: 0-255 lub 0-100%, g, b)

Tworzenie reguł stylów

– lub jako kombinacja

```
form {border: 5px solid rgb(255, 12, 35);}
p {margin: 100px 150px 100px 80px;}
a.button {
      padding: 5pt 15pt 5pt 15pt;
      border: 1px solid black;
      border-radius: 10px;
      background-image: url("img/button.png");
}
```

Tworzenie reguł stylów

- CSS jest bardzo "czuły" na składnię
- Brak jednego średnika czy nawiasu może powodować niepoprawne wyświetlanie strony
- Wszystkie właściwości i ich wartości znajdują się na stronie W3C, pod adresem

  http://www.w3schools.com/cssref/default.asp

Dziedziczenie

- Dziedziczenie to przejmowanie definicji przez potomków po przodkach
- Dzięki temu zmniejsza się ilość kodu
- Należy jednak pamiętać, że nie wszystkie właściwości są domyślnie dziedziczone po przodkach (aby wymusić dziedziczenie dla danej właściwości, należy zamiast wartości wpisać słowo `inherit`)

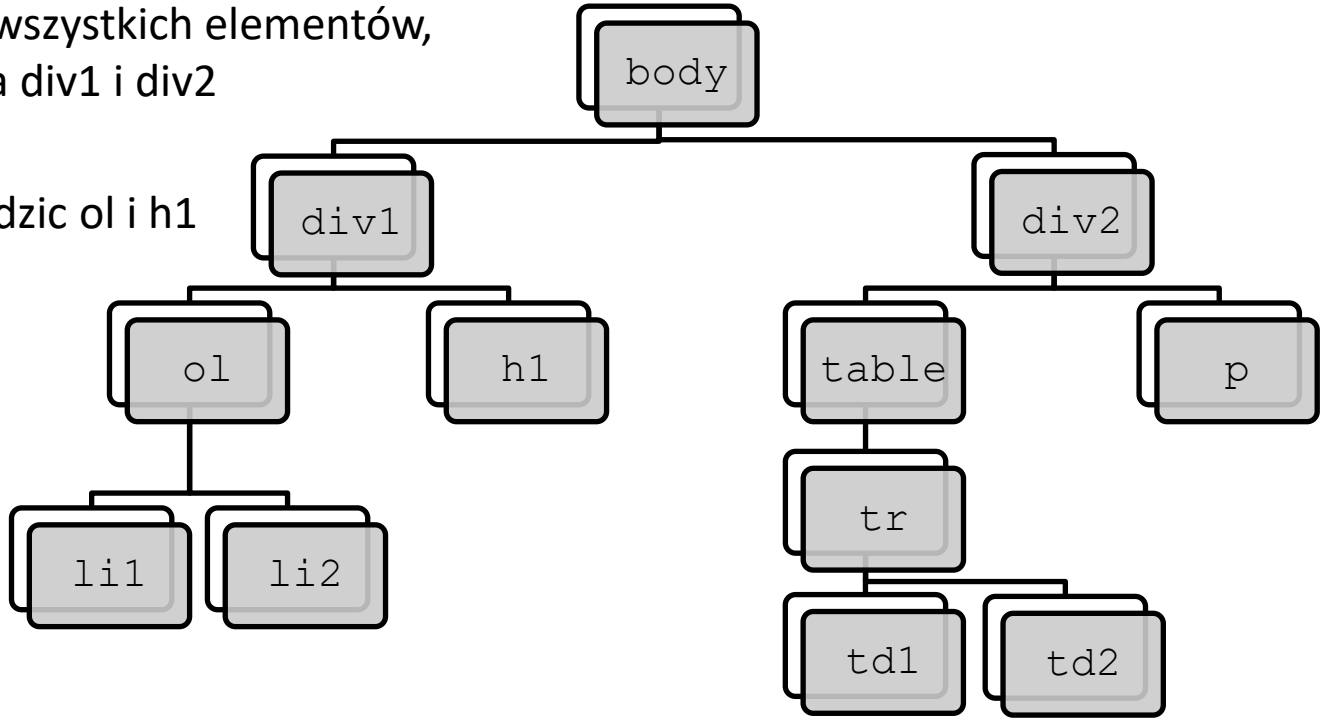
Dziedziczenie

przodek wszystkich elementów,
rodzic dla div1 i div2

dziecko body, rodzic ol i h1

potomek body

rodzeństwo



Dziedziczenie

- Odwoływanie się do potomków – selektory przodka i potomka oddzielamy spacją

```
div#div1 li {…;}
```

- Odwoływanie się do dzieci – selektory rodzica i dziecka oddzielamy znakiem >

```
div#div2 > table {…;}
```

Dziedziczenie

- Reguły przodka są dziedziczone przez potomków (nie wszystkie, np. `color` – tak, `border` – nie)
- Przodek i potomek mogą mieć jednak różne reguły dla tych samych właściwości
- Aby przesłonić regułę przodka, należy zdefiniować regułę dla potomka

Dziedziczenie

```
body {
    color: green;
    font-weight: bold;}
p.red {color: red;}
```

Wszystkie <p> będą miały kolor zielony i pogrubioną czcionkę, poza <p> klasy red, które będą czerwone.

Dziedziczenie

- To, która reguła zostanie zastosowana, zależy od poziomu jej szczegółowości:

```
div p {color: blue;}
p {color: brown}
```

Wszystkie `<p>` znajdujące się w sekcji `<div>` będą niebieskie, pozostałe brązowe.

Właściwości czcionek

- CSS wspiera 5 rodzajów czcionek:
  - szeryfowa: zakończenia znaków posiadają ozdoby w postaci kresek, przedłużeń, zawijanych końcówek

  Zażółć gęślą jaźń (Times New Roman)

  - bezszeryfowa

  Zażółć gęślą jaźń (Arial)

Właściwości czcionek

– pochyła

Zażółć gęślą jaźń (Times New Roman Italic)

– fantazyjna

Zazółc gesla jazn (Freestyle Script)

– monotypiczna – każdy znak ma taką samą szerokość

Zażółć gęślą jaźń (Consolas)

Właściwości czcionek

- Za pomocą CSS możemy ustawić następujące właściwości:
  - rodzaj czcionki (`font-family: Arial, "Times New Roman";`)
  - rozmiar czcionki (`font-size: 12pt / 1.2em / 75% / small / medium / x-large / smaller / larger;`)

Właściwości czcionek

– styl czcionki (

`font-style: normal / italic / oblique`

`font-weight: normal / bold / light / lighter / bolder;`

`font-variant : small-caps;`)

– interlinię (`line-height: 2em`)

Formatowanie tekstu

- Tekst może zostać sformatowany m.in. za pomocą:
  - wyrównania poziomego (`text-align: left / right / center / justify;`)
  - wyrównania pionowego (`vertical-align: top / middle / bottom;`)

Formatowanie tekstu

- wcięcia w pierwszym wierszu (`text-indent: 10mm / -10px;`)
- odstępów między literami (`letter-spacing: 2pt;`) i wyrazami (`word-spacing: -2px;`)
- dekorowania (

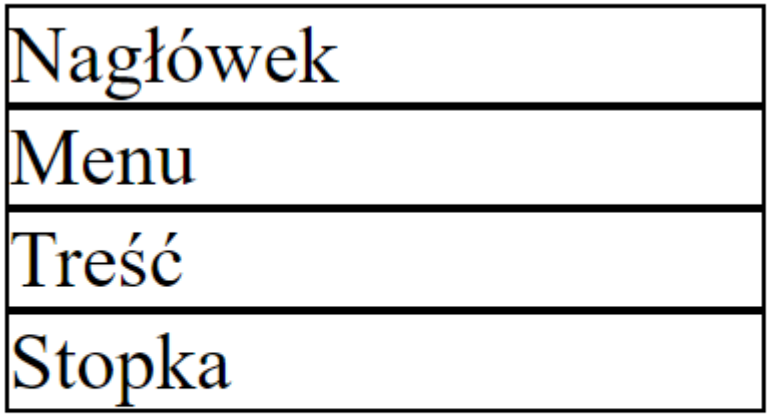
```
text-decoration: none / underline / overline / line-through / blink;
text-shadow: 0px -5px 10px #FF0000;)
```

Formatowanie tekstu

- Domyślny przepływ obiektów w dokumencie HTML to od lewej do prawej oraz od góry do dołu
- Do zmiany domyślnego przepływu elementów w dokumencie HTML służy właściwość `float`, która może przyjąć wartość `left`, `right` lub `none` (domyślna)
- Obiekt z ustawioną właściwością `float` staje się obiektem przestawnym

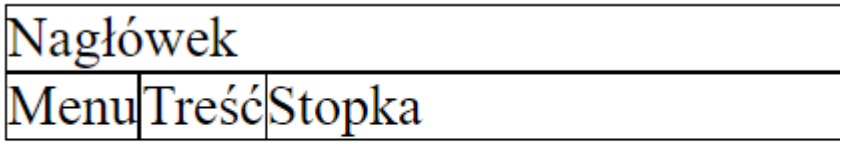
Formatowanie tekstu

```
div {
border: 1px solid
black;}


<div>Nagłówek</div>
<div>Menu</div>
<div>Treść</div>
<div>Stopka</div>
```



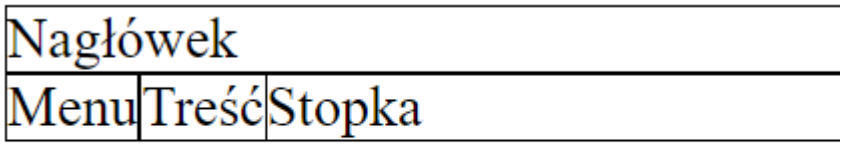
Formatowanie tekstu

```
.left {
 float: left;}

<div>Nagłówek</div>
<div class="left">Menu</div>
<div class="left">Treść</div>
<div>Stopka</div>
```



Formatowanie tekstu

- Dzięki temu rozwiązaniu jesteśmy w stanie zbudować layout strony bez używania tabeli (jak to było w zamierzchłych czasach)
- Wadą jest to, że nigdy do końca nie wiemy jak ostatecznie będzie wyglądał dokument

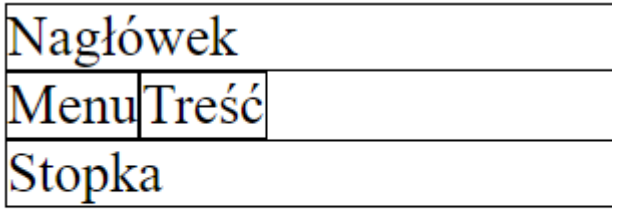


Formatowanie tekstu

- Częściowym rozwiązaniem tego problemu jest ustawienie właściwości `clear` dla elementu następnego po przestawnym
- `clear` przyjmuje wartości `left`, `right` lub `both`. Informuje ona przeglądarkę, po której stronie elementu NIE może pojawić się obiekt przestawny

Formatowanie tekstu

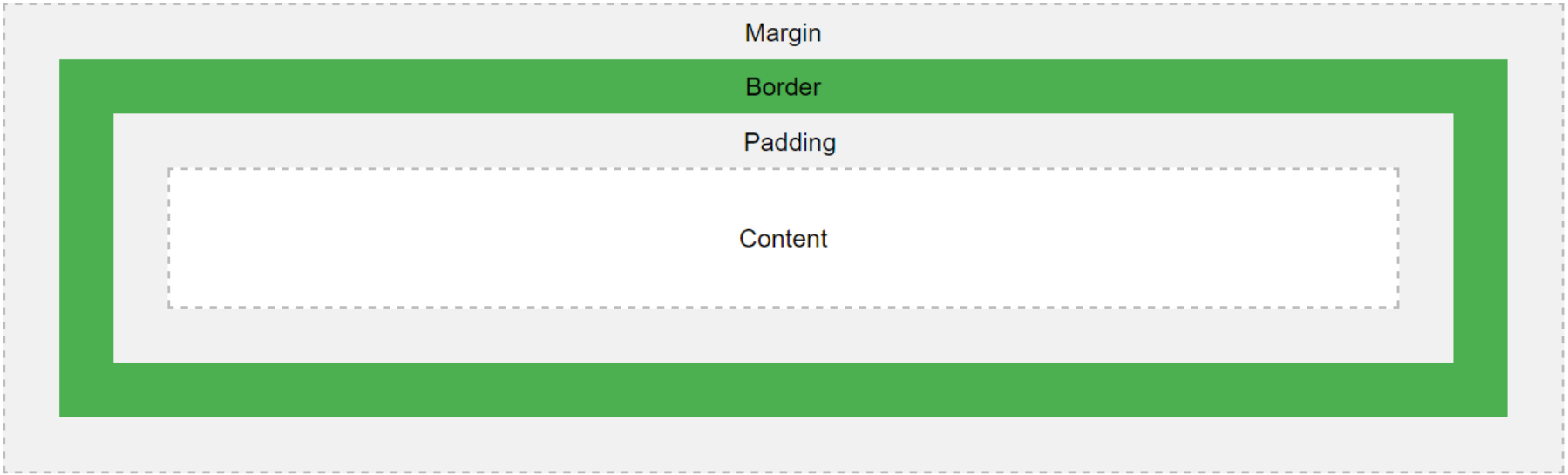
```
.clear{
 clear: both;}

<div>Nagłówek</div>
<div class="left">Menu</div>
<div class="left">Treść</div>
<div class="clear">Stopka</div>
```



Obramowanie i marginesy

- Każdy element HTML znajduje się w kontenerze, któremu można ustawić następujące właściwości:
  - `border` – ramka
  - `padding` – odstęp zawartości od ramki
  - `margin` – odstęp ramki od innych elementów
- Takie rozwiązanie nazywane jest Box Model (model pudełkowy)

Obramowanie i marginesy



źródło: http://www.w3schools.com

Obramowanie i marginesy

Obramowanie

```
border: 2px solid red;
      border-width: 2px 4px 6px 8px;
          border-top-width: 2px;
          border-right-width: 4px;
          border-bottom-width: 6px;
          border-left-width: 8px;
```

Obramowanie i marginesy

```
border: 2px solid red;
    border-style: solid dotted dashed
double;
          border-top-style: solid;
          border-right-style: dotted;
          border-bottom-style: dashed;
          border-left-style: double;
```

Obramowanie i marginesy

```
border: 2px solid red;
    border-color: red blue #FF0000 rgb(255,175, 81);
            border-top-style: red;
            border-right-style: blue;
            border-bottom-style: #FF0000;
            border-left-style: rgb(255, 175, 81);
```

Obramowanie i marginesy

Odstęp

```
padding: 50px;
    padding-top: 2em;
    padding-right: 10%;
    padding-bottom: 2cm;
    padding-left: 1.5ex;
```

Obramowanie i marginesy

Margines

```
margin: 3mm;
    margin-top: 10px;
    margin-right: -1em;
    margin-bottom: 3%;
    margin-left: 1in;
```

Definiowanie stylów tabeli

Style tabel i komórek można definiować za pomocą następujących właściwości:

- `border` (oddzielnie dla tabeli i komórek, ponieważ nie jest dziedziczona)
- `padding`
- `border-spacing` (odstęp między komórkami)

Definiowanie stylów tabeli

- `width`
- `table-layout` (określa, czy szerokość kolumny zależy od najszerszego znajdującego się w niej elementu - `auto`, czy też od ustawionych wartości `width` - `fixed`)
- `text-align` i `vertical-align`
- `empty-cells` (pokazuje – `show` lub ukrywa – `hide` puste komórki)

Kolory i tło

- Większość elementów umożliwia zmianę koloru pierwszoplanowego i tła
- Za zmianę koloru pierwszoplanowego odpowiada właściwość `color` (dla elementów tekstowych będzie to kolor czcionki), natomiast tła – `background-color`

Kolory i tło

- Inne, szczegółowe właściwości związane z kolorami:
  - `border-color`
  - `column-rule-color` (kolor odstępu między kolumnami tekstu)
  - `outline-color` (dodatkowe obramowanie elementu)
  - `text-decoration-color` (tylko Firefox)

Kolory i tło

- Wartość możemy podać w postaci:
  - słowa kluczowego, np. `red`
  - szesnastkowo, np. `#ff0000`
  - w RGB, np. `rgb(255, 0, 0)`
  - w RGBA, np. `rgba(255, 0, 0, 0.3)`
  - w HSL, np. `hsl(120, 70%, 85%)`
  - w HSLA, np. `hsl(120, 70%, 85% , 0.3)`

Kolory i tło

- Tłem elementu może być również obraz
- Służy do tego właściwość `background-image` oraz wartość w postaci ścieżki do pliku, generowanej za pomocą funkcji `url()`
- Gdy plik umieszczony jest na tym samym serwerze co dokument HTML/CSS, w którym jest odwołanie do niego, podajemy ścieżkę względem dokumentu, np.

```
background-image: url("img/bg-image.png");
```

Kolory i tło

- Jeśli chcemy odwołać się do obrazu zamieszczonego w Internecie, musimy podać jego adres www, np.

```
background-image:
url("http://obrazki.pl/logo.jpg");
```

Kolory i tło

- Obrazy użyte jako tło domyślnie są powielane tak, aby zapełnić całą powierzchnię elementu
- Dzięki właściwości `background-repeat` można to zachowanie kontrolować
- `background-repeat` przyjmuje następujące wartości: `repeat`, `repeat-x`, `repeat-y` lub `no-repeat`

Kolory i tło

- Pozwala to dołączać do dokumentu pliki graficzne, które są bardzo małe, ale dzięki odpowiedniemu powieleniu wypełniają całą przestrzeń
- Inną właściwością tła w formie grafiki jest `background-attachment`. Jeśli przyjmuje wartość `scroll` (domyślna), to grafika jest przewijana razem z dokumentem. `fixed` powoduje wyłączenie tej opcji

Kolory i tło

- Jeśli zawartość wykracza poza wielkość elementu, można zdefiniować właściwość `overflow`, która może przyjąć wartość:
  - `visible` – domyślna, zawartość jest widoczna
  - `hidden` – zawartość jest przycinana do wielkości elementu
  - `scroll` – zawsze dostępne są suwaki umożliwiające przewijanie zawartości
  - `auto` – suwaki dodawane są automatycznie, wedle potrzeby

Formatowanie list

- CSS pozwala nam zmienić domyślny format list
- Do najczęściej wykorzystywanych właściwości należą:
  - `list-style-type` – zmiana typu punktora, np. `disc`, `decimal`, `square` (wszystkie typy dostępne na stronie http://www.w3schools.com/cssref/pr_list-style-type.asp)

Formatowanie list

- `list-style-position` – położenie punktora – `inside` lub `outside`
- `list-style-image` – obraz jako punktor, np. `url("dot.jpg")`

Pozycjonowanie elementów na stronie

- Wyróżniamy 5 sposobów pozycjonowania (właściwość `position`) elementów na stronie:
  - statyczne (wartość `static`)
  - względne (`relative`)
  - bezwzględne (`absolute`)
  - stałe (`fixed`)
  - oraz dryfowanie (właściowość `float`, omówiona przy formatowaniu tekstu)

Pozycjonowanie elementów na stronie

- Pozycjonowanie statyczne (`position: static`) jest pozycjonowaniem domyślnym
- Elementy są wyświetlane zgodnie z ich miejscem w dokumencie i standardowym przepływem obiektów

Pozycjonowanie elementów na stronie

- Pozycjonowanie względne (`position: relative`) jest stosowane, gdy chcemy przesunąć obiekt względem jego standardowej pozycji
- Domyślny obszar elementu jest rezerwowany
- Do przesuwania obiektu względem domyślnej pozycji służą właściwości `top`, `right`, `bottom`, `left`

Pozycjonowanie elementów na stronie

- Wartość bezwzględna (`position: absolute`) pozycjonuje element względem okna przeglądarki
- Obszar domyślnej pozycji elementu nie jest w tym przypadku rezerwowany

Pozycjonowanie elementów na stronie

- Pozycjonowanie stałe (`position: fixed`) działa podobnie jak bezwzględne, przy czym pozycja elementu nie zmienia się wraz z przewijaniem okna przeglądarki
- Najczęściej wykorzystywane do stałego wyświetlania menu strony

Pozycjonowanie elementów na stronie

- Z pozycjonowaniem elementów wiążą się dodatkowo właściwości:
  - `min- max-width` (minimalna/maksymalna szerokość)
  - `min- max-height` (minimalna/maksymalna wysokość)
  - `visibility` (widoczność)
  - `display` (wyświetlanie)

Pozycjonowanie elementów na stronie

- Właściwość `visibility` może przyjmować następujące wartości:
  - `visible` – element jest wyświetlany
  - `hidden` – element nie jest wyświetlany, ale jego miejsce zostaje zarezerwowane

Pozycjonowanie elementów na stronie

- Bardzo użyteczną właściwością jest `display`, która może przyjmować m.in. wartości:
  - `none` – element nie jest wyświetlany, a jego miejsce nie jest rezerwowane
  - `block` – element jest wyświetlany jako blok
  - `inline` – element jest wyświetlany w linii
  - `table` – element jest wyświetlany jako tabela

Testowanie i walidacja kodu CSS

- Testowanie kodu należy zacząć od sprawdzenia sposobu jego interpretowania przez przeglądarkę
- Należy wziąć pod uwagę, że użytkownicy naszej strony będą korzystać z różnych przeglądarek w różnych wersjach co może stwarzać problemy z prawidłowym wyświetlaniem strony

Testowanie i walidacja kodu CSS

- Związane jest to z różnym stopniem implementacji standardu CSS w przeglądarce oraz zmieniającymi się na przestrzeni czasu standardowymi wartościami
- Najwięcej problemów sprawiają starsze wersje Internet Explorer (9.0 i niższe)
- Dobrą praktyką jest zapewnienie zgodności kodu z maksymalnie możliwą ilością przeglądarek

Testowanie i walidacja kodu CSS

- Wyrobienie sobie nawyku formatowania kodu, jego logicznego ułożenia w dokumencie CSS oraz stosowania komentarzy poprawi jego czytelność i ułatwi wprowadzanie zmian
- Standardem jest stosowanie języka angielskiego w nazewnictwie selektorów (klasy, id, nazwy) i komentarzach – nigdy nie wiemy kto będzie czytał nasz kod

Testowanie i walidacja kodu CSS

- Warto korzystać z edytorów wstępnie formatujących kod, kolorujących składnię oraz podkreślających błędy, np. Notepad++, BlueFish, Kompozer, NetBeans
- Podobnie jak w przypadku kodu HTML, kod CSS można sprawdzić na stronach W3C

Testowanie i walidacja kodu CSS

- Na stronie http://jigsaw.w3.org/css-validator/ możemy podać adres naszej strony, zaimportować plik CSS lub wkleić kod
- Wynikiem będzie dokładna informacja o rodzaju i miejscu błędu
- Jeśli kod nie będzie posiadał błędów – wyświetlą się gratulacje (czego wszystkim życzę)

A jeśli…

… klient chce zapłacić mało (a każdy chce), albo nam się nie chce, tudzież nie mamy zmysłu artystycznego (patrz: prowadzący i przygotowane przez niego przykłady), możemy użyć gotowych rozwiązań (framworków), np. Bootstrap, Foundation, YAML, Gumby i wiele wiele innych

Pytania

?

Dziękuję za uwagę!