

# **Technologie internetowe**

## **Cascading Style Sheets**

**Paweł Rajba**

[pawel@ii.uni.wroc.pl](mailto:pawel@ii.uni.wroc.pl)  
<http://www.kursy24.eu/>

# Zawartość modułu

---

- Wprowadzenie
- Wstawienie arkusza CSS, Reguły CSS
- Drzewo dokumentu
- Grupowanie, jednostki długości, kolory i napisy
- Ustalanie wartości, kaskada, rodzaje mediów
- Komentarze i uwagi, selektory
- Pseudoelementy, pseudoklasy
- Box model, przegląd właściwości
- Pozycjonowanie, wyświetlanie

# Wprowadzenie

---

- Dlaczego powstał CSS?
- Jak tego używać?
- Walidator  
<http://jigsaw.w3.org/css-validator/>

# Wstawienie arkusza CSS

---

- Bezpośrednio w dokumencie

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Moja strona domowa</TITLE>
    <STYLE type="text/css">
      H1 { color: red }
      P { color: blue }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>Moja fantastyczna strona domowa</H1>
    <P>Tutaj zobaczycie to, czego nigdzie indziej...
  </BODY>
</HTML>
```

# Wstawienie arkusza CSS

---

- Przez odwołanie do innego dokumentu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Moja strona domowa</TITLE>
    <LINK rel="StyleSheet" type="text/css" href="mojstyl.css">
  </HEAD>
  <BODY>
    <H1>Moja fantastyczna strona domowa</H1>
    <P>Tutaj zobaczycie to, czego nigdzie indziej...
  </BODY>
</HTML>
```

# Reguły CSS

---

- Reguła składa się z dwóch części
  - selektora (np. `h1`)
  - deklaracji (np. `color:blue`)
- Deklaracja również składa się z dwóch części
  - właściwości (np. `color`)
  - wartości (np. `blue`)
- Przykład
  - `h1 { color : blue; }`

# Reguły CSS

---

- Reguły możemy umieszczać
  - Wewnątrz arkuszy CSS
  - Jako atrybut *style* dowolnego znacznika
    - np. `<div style="text-align : center; color : #808000;">...</div>`
- Reguły można wczytać z zewnętrznego pliku
  - `@import "mystyle.css"`
  - `@import url("mystyle.css")`

# Drzewo dokumentu

---

- Rozpatrzmy poniższy dokument

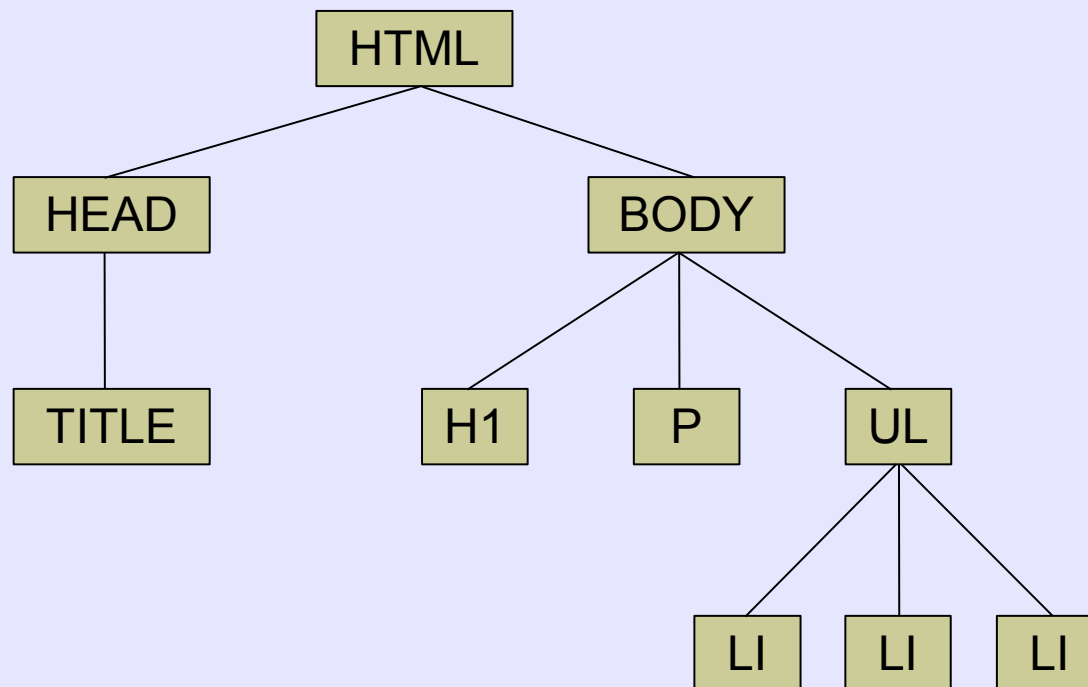
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Moja strona domowa</TITLE>
  </HEAD>
  <BODY>
    <H1>Moja fantastyczna strona domowa</H1>
    <P>Tutaj zobaczycie to, czego nigdzie indziej...
    <UL>
      <LI>Tygrysy</LI>
      <LI>Zebry</LI>
      <LI>Małpiszony</LI>
    </BODY>
</HTML>
```



# Drzewo dokumentu

---

- I odpowiadające mu drzewko



# Drzewo dokumentu

---

- Terminologia
  - Rodzic (parent)
    - element B zawiera bezpośrednio pewną ilość elementów. Jest on rodzicem tych elementów.
  - Dziecko (child)
    - element A jest dzieckiem elementu B, kiedy B jest rodzicem A
  - Potomek (descendant)
    - element A jest potomkiem elementu B, kiedy
      - A jest dzieckiem B lub
      - A jest dzieckiem C, który jest potomkiem B

# Drzewo dokumentu

---

- Terminologia c.d.
  - Przodek (ancestor)
    - element A jest przodkiem elementu B, kiedy B jest potomkiem A
  - Brat/siostra (sibling)
    - element A jest bratem elementu B, kiedy A i B mają wspólnego rodzica
    - A jest poprzedzającym/następującym bratem B, kiedy występuje wcześniej/później w drzewie dokumentu

# Drzewo dokumentu

---

- Terminologia c.d.
  - Element wcześniejszy (preceding element)
    - element A jest wcześniejszy niż element B, kiedy
      - A jest przodkiem B lub
      - A jest poprzedzającym bratem B
  - Element późniejszy (following element)
    - element A jest późniejszy niż element B, kiedy B jest elementem wcześniejszym niż element A

# Drzewo dokumentu

---

- Dziedziczenie
  - Elementy dziedziczą wartości po swoim rodzicu z drzewa dokumentu
    - (jeśli to dziedziczenie jest dopuszczone)
  - Przykład:
    - `<H1>To jest <EM>ważny</EM> nagłówek</H1>`
      - Jeśli kolor w EM nie jest ustawiony, a H1 ma kolor czerwony, to EM też będzie miało kolor czerwony.

# Grupowanie

---

- Grupowanie deklaracji

- `H1 { font-weight: bold }`  
`H1 { font-size: 12pt }`

- `H1 {`  
`font-weight: bold;`  
`font-size: 12pt;`  
`}`

- Grupowanie selektorów

- `H1 { font-family: sans-serif }`  
`H2 { font-family: sans-serif }`  
`H3 { font-family: sans-serif }`

- `H1, H2, H3 { font-family: sans-serif }`

# Jednostki długości

---

- Jednostki długości:
  - relatywne:
    - em – szerokość znaku M
    - ex – wysokość linii
    - px – 1 pixel
  - absolutne
    - in – cale (2.54 cm)
    - cm, mm
    - pt – punkty (1/72 cala)
    - pc – picas (12 pt)
  - procentowe: np. 120%

# Kolory i napisy

---

- Kolory

- `H1 { color: maroon }`
- `P { color: #f00 } /* #rgb */`
- `P { color: #ff0000 } /* #rrggbb */`
- `P { color: rgb(255,0,0) } /* 0 - 255 */`
- `P { color: rgb(100%, 0%, 0%) }`

- Napisy

- `"this is a 'string'"`
- `"this is a \"string\""`
- `'this is a "string"'`
- `'this is a \'string\''`



# Ustalanie wartości

---

- W CSS są trzy rodzaje wartości:
  - Określone (specified)
  - Wyliczone (computed)
  - Faktyczne (actual)

# Ustalanie wartości

---

- Wartości określone
  - Są ustalane wg ustalanie następujących kryteriów:
    - Uzyskiwana jest wartość z kaskady (w arkuszu wprost mamy wartość dla danej właściwości)
    - Jeśli wartość jest dziedziczona, używana jest ta wartość
    - Używana jest wartość domyślna
- Wartości wyliczane
  - Wartości „określone” mogą być absolutne (red, 2mm) lub relatywne (auto, 2em, 120%)
  - Dla wartości relatywnych trzeba dokonać obliczeń
    - Dla wartości absolutnych obliczeń wykonywać nie trzeba

# Ustalanie wartości

---

- Wartości faktyczne
  - Wartość ostatecznie przypisana
  - Może się różnić od wartości wyliczonej, gdy np. zostanie obliczona wielkość czcionki na 15px, a w systemie dostępne są czcionki 14px i 16px
  - Inaczej mówiąc: wartości faktyczne to wartości wyliczone lub określone po zaokrągleniach lub przybliżeniach

# Kaskada

---

- Arkusze styli mogą mieć trzy różne źródła pochodzenia:
  - Arkusz autora (author)
  - Arkusz użytkownika (user)
  - Arkusz przeglądarki (user agent)

# Kaskada

---

- Sposób wyszukiwania wartości w arkuszu
  - Znaleźć wszystkie deklaracje, które dopasowują się do elementu i właściwości (wyszukanie odpowiednich selektorów)
  - Pierwsze sortowanie jest wg wag i źródła pochodzenia:
    - domyślnie, reguły authora mają wyższy priorytet od reguł usera
    - jeśli reguła jest !important to porządek ten się ulega odwróceniu
    - reguły !important są ważniejsze od reguł normalnych
    - reguły user agenta mają najniższy priorytet

# Kaskada

---

- Sposób wyszukiwania wartości w arkuszu
  - Drugie sortowanie jest wg szczegółowości: bardziej szczegółowe selektory mają wyższy priorytet od tych ogólnych.
  - Ostatecznie, jeśli dwie reguły mają ten sam priorytet, to wygrywa ta, która pojawiła się później.

# Kaskada

---

- Sposób wyszukiwania wartości w arkuszu

Przykład:

- ```
/* From the user's style sheet */
P { text-indent: 1em ! important }
P { font-style: italic ! important }
P { font-size: 18pt }
```
- ```
/* From the author's style sheet */
P { text-indent: 1.5em !important }
P { font: 12pt sans-serif !important }
P { font-size: 24pt }
```

```
text-indent: 1em
font-style: italic
font-size: 12pt
font-family: sans-serif
```

# Rodzaje mediów

---

- Po co określać rodzaj medium
- Określenie rodzaju medium
  - W arkuszu stylów:

```
@import url("loudvoice.css") aural;  
@media print { /* definicja arkusza styli */ }
```
  - Jako atrybut znacznika LINK

```
<LINK rel="stylesheet" type="text/css"  
      media="print, handheld" href="foo.css">
```
- Rozpoznawane typy mediów
  - all, aural, braille, embossed, handheld, print, projection, screen, tty, tv



# Komentarze i uwagi

---

- Komentarze `/* ... */`
- Dwa różne obiekty: "napis", stała
  - `color: "red"` (źle)
- Można przełamywać wiersze
  - `A[TITLE="a not s\  
o very long title"]  
{/*...*/}`  
  
`A[TITLE="a not so very long title"]  
{/*...*/}`

# Selektory

---

- Universal selector
  - \*, jeżeli po \* jest coś jeszcze, \* można pominąć
  - Dopasowuje się do każdego elementu drzewa dok.
  - \*.gruby .gruby (równoważne)  
\*#myid #myid (równoważne)
- Type selectors
  - dopasuje się do każdego wystąpienia konkretnego elementu
  - H1 { font-family: sans-serif }

# Selektory

---

- Descendant selectors
  - Dopasuje się do elementu, który jest potomkiem innego elementu
  - Przykład: weźmy dwie reguły:
    - `H1 { color: red }`      `EM { color: red }`
  - No i jest kłopot z poniższym:
    - `<H1>To jest <EM>bardzo</EM> ważny tekst</H1>`

# Selektory

---

- Descendant selectors
  - Dopasuje się do elementu, który jest potomkiem innego elementu
  - Przykład: weźmy dwie reguły:
    - H1 { color: red }      EM { color: red }
  - No i jest kłopot z poniższym:
    - <H1>To jest <EM>bardzo</EM> ważny tekst</H1>
  - Rozwiązanie:
    - H1 { color: red }      EM { color: red }
    - H1 EM { color: blue }

# Selektory

---

- Child selectors
  - Dopasuje się do elementu, który jest dzieckiem innego elementu
  - BODY > P { color: red }
  - DIV OL>LI P { color: green }
- Adjacent sibling selectors
  - H1 + H2 { margin-top: 5mm }
  - H1.opener + H2 { margin-top: 5mm }  
(dotyczy H2)

# Selektory

---

- Attribute selectors
  - [att] – dopasuje się wtedy, gdy atrybut att jest ustawiony, niezależnie od jego wartości
    - H1[title] { color : blue }
  - [att=val] – dopasuje się wtedy, gdy atrybut att jest ustawiony na wartość val
    - SPAN[class="main"] { color : red }
    - SPAN[align="left"][class="g"] {color: blue}

# Selektory

---

- Attribute selectors c.d.
  - [att~=val] – dopasuje się wtedy, gdy wartością atrybutu att jest lista, w której jeden z elementów jest równy dokładnie val
    - P[align~= "left"] { color : green }  
// dopasuje się dla np. rel="left right"
  - [att|=val] – dopasuje się wtedy, gdy wartością atrybutu att jest lista, w której elementem jest val lub słowo rozpoczynające się słowem val-
    - \*[lang|= "en"] { color : blue }  
// dopasuje się wtedy, gdy atrybut lang  
// będzie ustawiony na np. "en", "en-US"

# Selektory

---

- Class selectors
  - W języku HTML możemy użyć kropki "." jako alternatywę dla składni "class~=".
    - DIV.value to to samo co DIV[class~=value]
  - Przykład:
    - .zielony { color : green }  
<H1 class="zielony">Zielony nagłówek</H1>
    - P.czerwony.gruby { color: red; font-weight: bold }  
<P class="czerwony gruby maly">Akapit</P>



# Selektory

---

- ID selectors
  - Każdy element HTML może mieć atrybut o nazwie id
  - Atrybut ten wyróżnia spośród innych to, że jego wartości są w obrębie dokumentu unikalne.
  - Przykład:
    - reguła `H1#chapter1 { text-align: center }`
    - dopasuje się do `<H1 id="chapter1">...</H1>`

# Pseudoelementy

---

- :first-line
  - Dopasuje się do pierwszego wiersza
  - Przykład
    - `P:first-line { text-transform: uppercase }`
    - `<P>`To jest bardzo długi tekst, który nie zmieści się w jednym wierszu i trzeba go wyświetlić w kilku wierszach. Zostaną wówczas dodane fikcyjne tagi.`</P>`
    - `<P><P:first-line>`To jest bardzo długi tekst, który nie`</P:first-line>` zmieści się w jednym wierszu i trzeba go wyświetlić w kilku wierszach. Zostaną wówczas dodane fikcyjne tagi.`</P>`

# Pseudoelementy

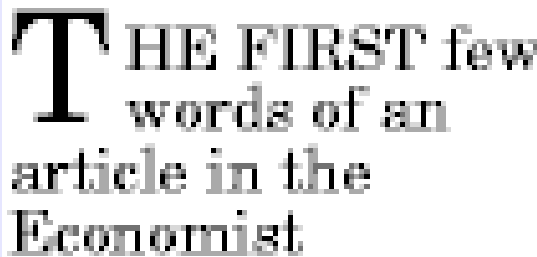
---

- :first-letter
  - Dopasuje się do pierwszej litery
  - Przykład:
    - `<STYLE type="text/css">`
      - `P { font-size: 12pt; line-height: 12pt }`
      - `P:first-letter { font-size: 200%; font-style: italic;`  
`font-weight: bold; float:left }`
      - `SPAN { text-transform: uppercase }`
    - `</STYLE>`
    - `<P><SPAN>The first</SPAN> few words of an article in`  
`The Economist.</P>`

# Pseudoelementy

---

- :first-letter c.d.
  - Przykład c.d.
    - `<P><SPAN><P:first-letter>T</P:first-letter>he first</SPAN> few words of an article in the Economist.</P>`



**T**HE FIRST few  
words of an  
article in the  
Economist

# Pseudoklasy

---

- :first-child
  - Dopasowuje się do elementu będącego pierwszym dzieckiem swojego rodzica
  - Przykład:
    - `DIV > P:first-child { color: navy }`
    - `<DIV>`  
`<P>Ala ma kota</P><P>Ala ma dwa koty</P>`  
`</DIV>`
    - `<DIV>`  
`<H1>Ala ma kota</H1><P>a tu nie działa</P>`  
`</DIV>`

# Pseudoklasy

---

- `:before` i `:after`
  - Określa co będzie umieszczone przed/za danym blokiem lub elementem
  - Zawartość określamy przez właściwość
    - `content`, gdzie wartość to kombinacja
      - napisów, przy czym można użyć w napisie `\A` co daje złamanie wiersza , ale widoczne jest tylko w połączeniu z `white-space: pre`
      - linków do obrazków
      - liczników

# Pseudoklasy

---

- `:before` i `:after`, używanie liczników
  - Funkcja `counter()`. Składnie
    - `counter(name)`
    - `counter(name, style)`
      - `name` to nazwa licznika
      - `style` są następujące
        - `disc` | `circle` | `square` | `decimal` | `decimal-leading-zero` | `lower-roman` | `upper-roman` | `lower-greek` | `lower-latin` | `upper-latin` | `armenian` | `georgian` | `lower-alpha` | `upper-alpha` | `none`
  - Inne przydatne właściwości
    - `counter-reset: { licznik [liczba] }+`
    - `counter-increment: { licznik [liczba] }+`

# Pseudoklasy

---

- :link – link nieodwiedzony
- :visited – link odwiedzony
- :hover – myszka nad linkiem
- :active – link aktywny
- :focus – focus ustawiony na linka
- Przykłady
  - A:focus:hover { color: lime }
  - A:hover { color: blue }
  - A.inny:link { color: red }



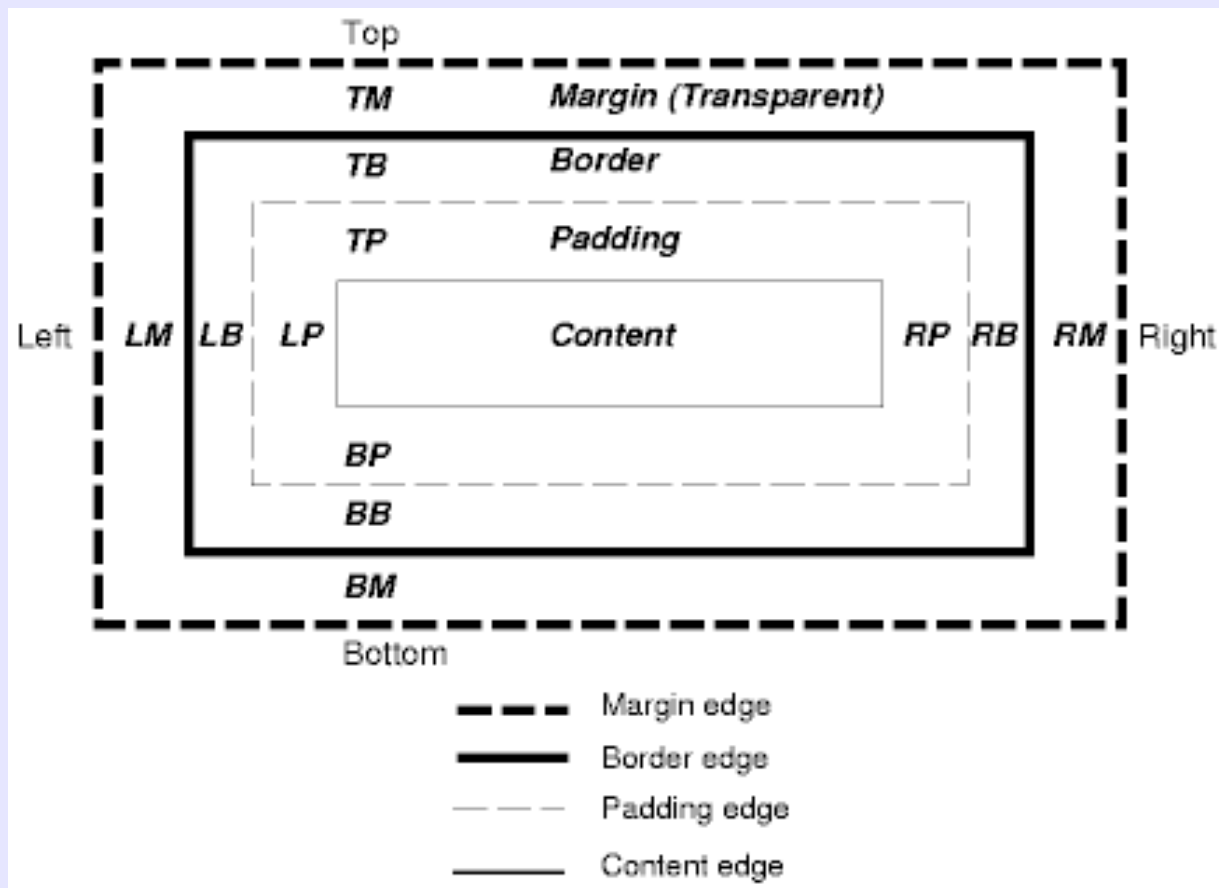
# Box model

---

- Dla każdego elementu drzewa dokumentu jest tworzony prostokątny blok, w którym ten element się znajduje.
- Każdy blok zawiera cztery obszary:
  - content,
  - padding,
  - border,
  - margin

# Box model

- Obrazkowo:



# Przegląd właściwości

---

- Formatowanie tekstu – czcionki
  - font-family – określa rodzaj czcionki
    - nazwa rodziny: Times, Arial
    - rodzina ogólna: serif, sans-serif, cursive, fantasy, monospace
    - Przykład: P { font-family: Arial, Helvetica, sans-serif }
  - font-style – czcionka normalna lub pochyła
    - Wartości: normal, italic, oblique
    - Przykłady:
      - H1 { font-style: italic }
      - H1 EM { font-style: normal }

# Przegląd właściwości

---

- Formatowanie tekstu – czcionki c.d.
  - font-variant – czcionka normalna lub kapitaliki
    - Wartości: normal, small-caps
    - Przykład: P { font-variant: small-caps }
  - font-weight – określa grubość czcionki
    - Wartości:
      - 100, 200, 300, 400, 500, 600, 700, 800, 900
      - normal – to samo co 400, bold – to samo co 700
      - bolder, lighter
    - Przykłady
      - P { font-weight: normal }
      - STRONG { font-weight: bolder }

# Przegląd właściwości

---

- Formatowanie tekstu – czcionki c.d.
  - font-size – określa wielkość czcionki
    - Wartości:
      - stałe: xx-small | x-small | small | medium | large | x-large | xx-large
      - relatywnie: larger | smaller
      - wielkości: 12pt
      - procentowo: 120%
    - Przykłady
      - P { font-size: 12pt; }
      - BLOCKQUOTE { font-size: larger }
      - EM { font-size: 150% }
      - EM { font-size: 1.5em }

# Przegląd właściwości

---

- Formatowanie tekstu – czcionki c.d.
  - font-stretch – określa „gęstość” czcionki
    - Wartości: ultra-condensed, extra-condensed, condensed, semi-condensed, normal (domyślne), semi-expanded, expanded, extra-expanded, ultra-expanded, wider, narrower
  - font – opisuje zestaw właściwości
    - czyli z grubsza to, co było do tej pory o czcionkach, tylko w jednym

# Przegląd właściwości

---

- Formatowanie tekstu
  - text-indent – wcięcie akapitu
    - Wartości: długość w px, pt, procentach, itd.
  - text-align – wyrównywanie tekstu
    - Wartości: left, right, center, justify
  - text-decoration – dekoracja tekstu
    - Wartości: none, underline, overline, line-through, blink
    - Przykład: `A[href] { text-decoration: underline }`
  - text-transform – wielkości liter
    - Wartości: capitalize, uppercase, lowercase, none

# Przegląd właściwości

---

- Formatowanie tekstu
  - color – określa kolor
    - Wartości: kolory w dowolnej postaci
  - line-height – odstęp między wierszami
    - Wartości: normal, dowolna liczba w jednostkach lub %
  - letter-spacing – odstęp między literami
    - Wartości: normal lub długości
  - word-spacing – odstęp między słowami
    - wartości: normal lub długości



# Przegląd właściwości

---

- Formatowanie tekstu
  - white-space – interpretacja znaku spacji
    - Wartości: normal, pre, nowrap
    - P.pre { white-space: pre }
    - P { white-space: normal }
    - TD[nowrap] { white-space: nowrap }
  - vertical-align – wyrównywanie tekstu w pionie
    - Wartości: Baseline, Sub, Super, Top, Text-top, Middle, Bottom, Text-bottom lub wielkość wyrażona w procentach

# Przegląd właściwości

---

- Konfiguracja tła
  - background-color – kolor tła
  - background-image – obrazek tła
    - BODY { background-image: url("marble.gif") }
    - P { background-image: none }
  - background-repeat – określa powtarzanie obrazka
    - Wartości: repeat, repeat-x, repeat-y, no-repeat
  - background-attachment – zaczepienie obrazka
    - scroll (domyślne) – z tekstem
    - fixed – z oknem

# Przegląd właściwości

---

- Konfiguracja tła
  - background-position – pozycja obrazka
    - Jako pozycję możemy wpisać jedną wartość:
      - top, center, bottom, left, center, right
      - długość – określi odległość od lewej krawędzi
    - lub dwie wartości
      - left top, left bottom , right top , right bottom
      - dwie długości – określą odległość od lewej i górnej krawędzi
  - background – kompleksowa konfiguracja tła
    - w szczególności: **background : none** usuwa tło

# Przegląd właściwości

---

- Ustawienie marginesów zewnętrznych
  - margin-top, margin-right, margin-bottom, margin-left
    - Wartości: liczbowe, procentowe lub auto
    - Domyślnie wszystkie mają wartość zero
  - margin
    - Wartości j.w.
    - Znaczenie wartości w zależności od liczby parametrów
      - 1 wartość – ustawia wszystkie strony
      - 2 wartości – #1: top i bottom, #2: left i right
      - 3 wartości – #1: top, #2: left i right, #3: bottom
      - 4 wartości – #1: top, #2: right, #3: bottom, #4: left

# Przegląd właściwości

---

- Ustawienie marginesów zewnętrznych
  - Uwagi:
    - marginesy mogą się czasami skolapsować (collapse), co oznacza, że dwa lub więcej marginesów zostanie zamienione na jeden, zwykle ten o największej wartości
    - marginesy poziome nie kolapsują się
    - marginesy bloków, których pozycje są absolutne lub relatywne nie kolapsują się

# Przegląd właściwości

---

- Ustawienie marginesów wewnętrznych
  - padding-top, padding-right, padding-bottom, padding-left
    - Wartości: liczbowe, procentowe lub auto
    - Domyślnie wszystkie mają wartość zero
  - padding
    - Wartości j.w.
    - Znaczenie wartości w zależności od liczby parametrów (jak w przypadku właściwości margin)

# Przegląd właściwości

---

- Ustawienie obramowania
  - Grubość:
    - Właściwości: border-top-width, border-right-width, border-bottom-width, border-left-width, border-width
    - Wartości
      - długości (np. 1cm)
      - stałe thin, medium, thick (thin<=medium<= thick)
  - Kolor
    - Właściwości: border-top-color, border-right-color, border-bottom-color, border-left-color, border-color
    - Wartości
      - kolor lub transparent (przezroczyty, ale ma grubość)

# Przegląd właściwości

---

- Ustawienie obramowania
  - Styl
    - Właściwości: border-top-style, border-right-style, border-bottom-style, border-left-style, border-style
    - Wartości:
      - none, hidden, dotted, dashed, solid, double (w połączeniu z border-width:thin daje to pojedynczą linię), groove, ridge, inset, outset
  - Kompleksowe:
    - Właściwości: border-top, border-bottom, border-right, border-left, border
    - Wartości: width style color



# Przegląd właściwości

---

- Rozmiary obiektów
  - width – szerokość
  - height – wysokość
  - max-width – maksymalna dopuszczalna szerokość
  - min-width – minimalna dopuszczalna szerokość
  - max-height – maksymalna dopuszczalna wysokość
  - min-height – minimalna dopuszczalna wysokość

# Przegląd właściwości

---

- Formatowanie list:
  - list-style-type: rodzaj wypunktowania:
    - wartości: disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-greek | lower-latin | upper-latin | armenian | georgian | lower-alpha | upper-alpha | none
  - list-style-image: url("adres obrazka")
    - obrazek wypunktowania
    - url podajemy względem lokalizacji arkusza CSS
  - list-style-position: { inside | outside }
    - pozycja treści wypunktowania
  - list-style: list-style-type list-style-position list-style-image

# Przegląd właściwości

---

- Formatowanie tabel:
  - caption-side: (top|bottom)
    - Określa miejsce opisu tabeli
  - table-layout
    - Określa algorytm planowania szerokości kolumn
    - Wartości: auto i fixed
      - auto – szerokości są nadawane automatycznie na podstawie zawartości
      - fixed – szerokości są nadawane oddzielnie
        - szerokość kolumny może być wprost określona
        - po równo w przypadku braku określenia szerokości kolumny

# Przegląd właściwości

---

- Formatowanie tabel:
  - border-collapse: (collapse|separate)
    - łączy razem sąsiednie krawędzie obramowania komórek i obramowania tabeli (lub nie łączy)
  - border-spacing: 10px
    - odpowiednik cellspacing
  - empty-cells: (show|hide)
    - pokazuje lub nie obramowanie pustych komórek
  - padding: 4px
    - odpowiednik cellpadding

# Pozycjonowanie

---

- Rodzaje pozycjonowanie
  - statyczne (domyślne), relatywne, absolutne, ustalone
- Pozycjonowanie jest określone następująco
  - position : static | relative | absolute | fixed
- Inne właściwości określające pozycjonowanie
  - left, right, top, bottom – lokalizacja pudełka
  - z-index – kolejność rysowania warstw

# Wyświetlanie

---

- Właściwości określające sposób wyświetlania
  - float – określa sposób opływania pudełka
    - left – pudełko do lewej i opływane z prawej
    - right – pudełko do prawej i opływane z lewej
    - none – pudełko w ogóle nie jest opływane
  - display – sposób wyświetlania pudełka
    - typowe wartości: block, inline, none
  - visibility – czy pokazywać pudełko
    - wartości: visible, hidden
    - różnica pomiędzy display i visibility

# Wyświetlanie

---

- Właściwości określające sposób wyświetlania
  - clear – określa, która krawędź pudełka ma nie przylegać do krawędzi poprzednich pudełek
    - wartości: left, right, both
  - overflow – co zrobić z tym co wystaje
    - wartości: visible, hidden, scroll
  - clip – określa obszar, który ma być wyświetlony
    - wartość: rect(lewa, góra, prawa, dół)

# Przykłady

---

- tabelki.html
- wykazy[1234].html
- wyswietlanie[12].html
- pozycjonowanie.html
- menu.html
- formularze.html
- szablony/\*



# CSS a przeglądarki

---

- Niestety cały czas producenci przeglądarek nie nadążają za pojawiającymi się standardami
- Problem prezentacji treści w przeglądarkach
  - często w każdej przeglądarce dokument wygląda inaczej
- Pomocne narzędzie
  - IETester  
<http://www.my-debugbar.com/wiki/IETester/HomePage>