

Paweł Rajba

pawel@cs.uni.wroc.pl

<http://pawel.ii.uni.wroc.pl/> | <http://itcourses.eu/>

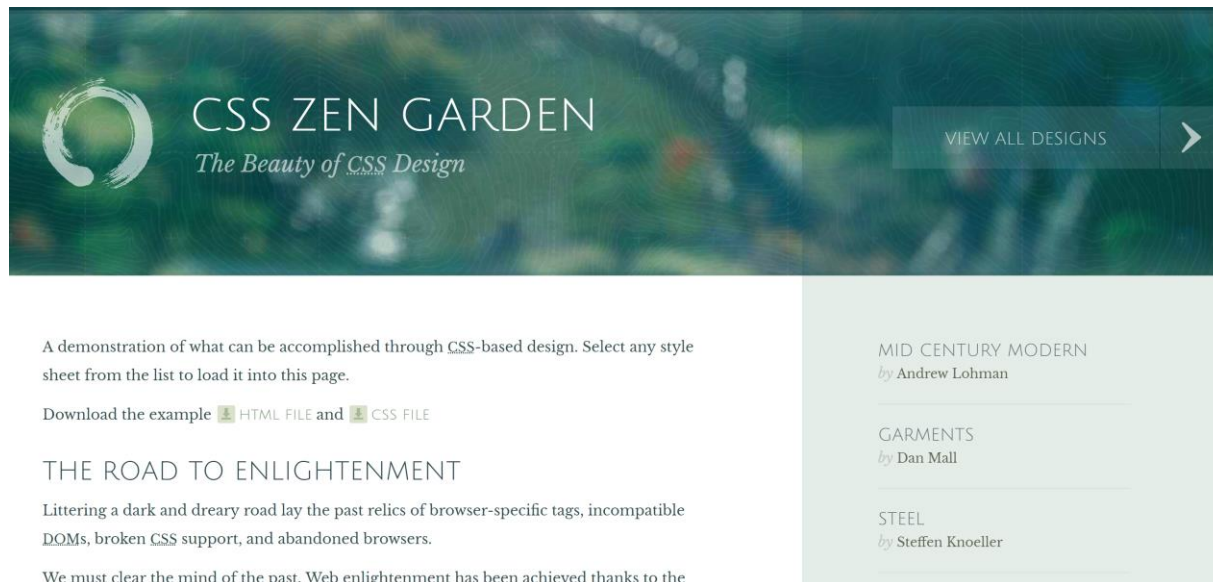
CSS

Agenda

- Wprowadzenie
- Osadzenie stylów CSS, Reguły CSS
- Podstawy
 - Grupowanie, jednostki długości, ustalanie wartości
 - Kolory i napisy, komentarze i uwagi, rodzaje mediów
- DOM revisited
- Selektory, pseudoelementy, pseudoklasy
- Box model, przegląd właściwości
- HTML Layouts,
 - Pozycjonowanie, wyświetlanie
 - Flexible box layout

Wprowadzenie

- Dlaczego powstał CSS?
- Warstwa prezentacji vs. warstwa treści
- CSS Zen Garden



Wprowadzenie

- Walidator
 - <http://jigsaw.w3.org/css-validator/>
- Narzędzia
 - <https://jsfiddle.net/>
 - <https://jsbin.com/>
 - <https://codepen.io/pen/>
 - Narzędzia deweloperskie Google Chrome
- CSS3 wprowadza sporo nowości,
przykładowe zestawienie:
 - <http://tutorialzine.com/2013/10/12-awesome-css3-features-you-can-finally-use/>

Osadzenie stylów CSS

- Bezpośrednio w dokumencie

```
<!DOCTYPE html>
<html>
  <head>
    <title>Moja strona domowa</title>
    <style type="text/css">
      h1 { color: red }
      p { color: blue }
    </style>
  </head>
  <body>
    <h1>Stowarzyszenie mruczających owieczek</h1>
    <p>Aby się zapisać do...</p>
  </body>
</html>
```

Osadzenie stylów CSS

- Poprzez odwołanie do innego dokumentu

```
<!DOCTYPE html>
<html>
  <head>
    <title>Moja strona domowa</title>
    <link rel="stylesheet" type="text/css" href="site.css">
  </head>
  <body>
    <h1>Stowarzyszenie mruczających owieczek</h1>
    <p>Aby się zapisać do...</p>
  </body>
</html>
```

Osadzenie stylów CSS

- Na poziomie znaczników (inline)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Moja strona domowa</title>
  </head>
  <body>
    <h1 style=" color: red">Stowarzyszenie m. owieczek</h1>
    <p style="color: blue">Aby się zapisać do...</p>
  </body>
</html>
```

Reguły CSS

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

Reguły CSS

- Własne implementacje w przeglądarkach
 - WebKit -web-kit-property
 - Mozilla -moz-property
 - Internet Explorer -ms-property
 - Opera -o-property
 - W3C property
- Reguły można wczytać z zewnętrznego pliku
 - @import "mystyle.css"
 - @import url("mystyle.css")

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)

Jednostki długości

- Nowe jednostki pomocne w RWD
 - ch
 - Relative to width of the "o" (zero)
 - rem
 - Relative to font-size of the root element
 - vw
 - Relative to 1% of the width of the viewport
 - vh
 - Relative to 1% of the height of the viewport
 - vmin
 - Relative to 1% of viewport's* smaller dimension
 - vmax
 - Relative to 1% of viewport's* larger dimension

https://www.w3schools.com/cssref/css_units.asp

<https://webdesign.tutsplus.com/articles/7-css-units-you-might-not-know-about--cms-22573>

Jednostki długości

- Więcej o jednostkach
 - Ciekawe artykuły
 - <https://www.w3.org/Style/Examples/007/units.pl.html>
 - <https://webplatform.github.io/docs/tutorials/understanding-css-units/>
 - https://www.quirksmode.org/blog/archives/2010/04/a_pixel_is_not.html
 - Peter-Paul Koch: A Pixel is not a Pixel, Fronteers 2012
 - Nagranie
 - <https://vimeo.com/52851511>
 - Transkrypcja
 - <https://fronteers.nl/congres/2012/sessions/a-pixel-is-not-a-pixel-peter-paul-koch>

Kolory i napisy

■ Kolory

- `H1 { color: maroon }`
- `P { color: #foo } /* #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)
- Ostateczne wartości wynikają z kilku czynników:
 - Dostępnych arkuszy:
 - autora (author), użytkownika (user), przeglądarki (user agent)
 - Szczegółowości selektorów
 - Zastosowania reguły !important
 - Dostępnych czcionek

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"]  
{ /* ... */ }
```


Rodzaje mediów

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

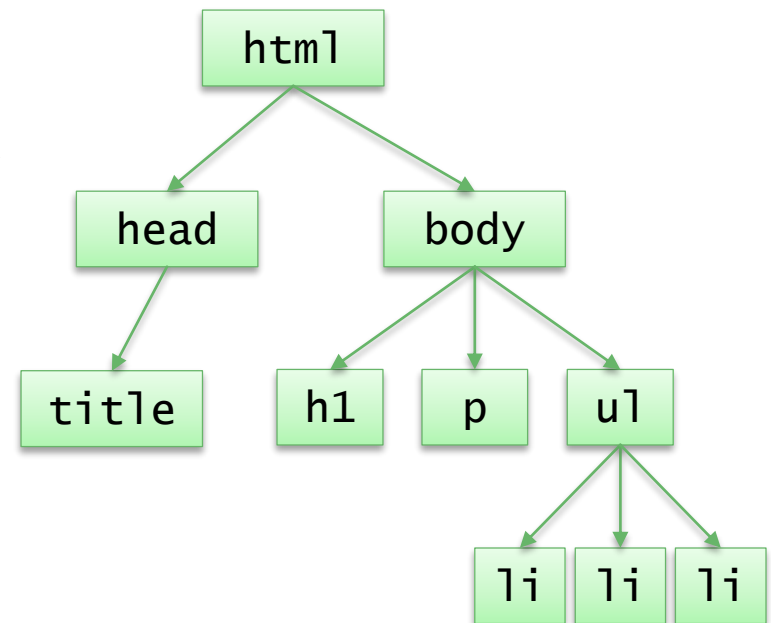
```
@import url("loudvoice.css") aural;  
@media print { /* definicja arkusza stylów */ }
```
 - Jako atrybut znacznika link

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

DOM revisited

- Rozpatrzmy poniższy dokument i odpowiadające mu drzewo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mruczące owieczki</title>
  </head>
  <body>
    <h1>Mruczące owieczki</h1>
    <p>Wrogowie</p>
    <ul>
      <li>Lwy</li>
      <li>Tygrysy</li>
      <li>Bracia Wezuwiusz</li>
    </ul>
  </body>
</html>
```



DOM revisited

- Terminologia
 - Rodzic (parent)
 - Dziecko (child)
 - Potomek (descendant)
 - Przodek (ancestor)
 - Brat/siostra (sibling)
 - Poprzedzający/następujący
 - Element wcześniejszy (preceding element)
 - Element późniejszy (following element)

DOM revisited

- Dziedziczenie
 - Elementy dziedziczą wartości po swoim rodzicu z drzewa dokumentu
 - jeśli to dziedziczenie jest dopuszczane
 - 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.

Selektory

- Universal selector

- Dopasowuje się do każdego elementu DOM
- *, jeżeli po * jest coś jeszcze, * można pominąć
 - *.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>`
 - 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>The first 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>`

THE 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 }

Nowe selektory w CSS3

- :first-of-type
- :last-of-type
- :only-of-type
- :nth-child(#)
- :nth-last-child(#)
- :nth-of-type(#)
- :nth-last-of-type(#)
- :last-child
- :root
- :empty
- :not(...)
- ::selection
- :enabled
- :disabled
- :checked
- :valid
- :invalid
- :optional
- :required

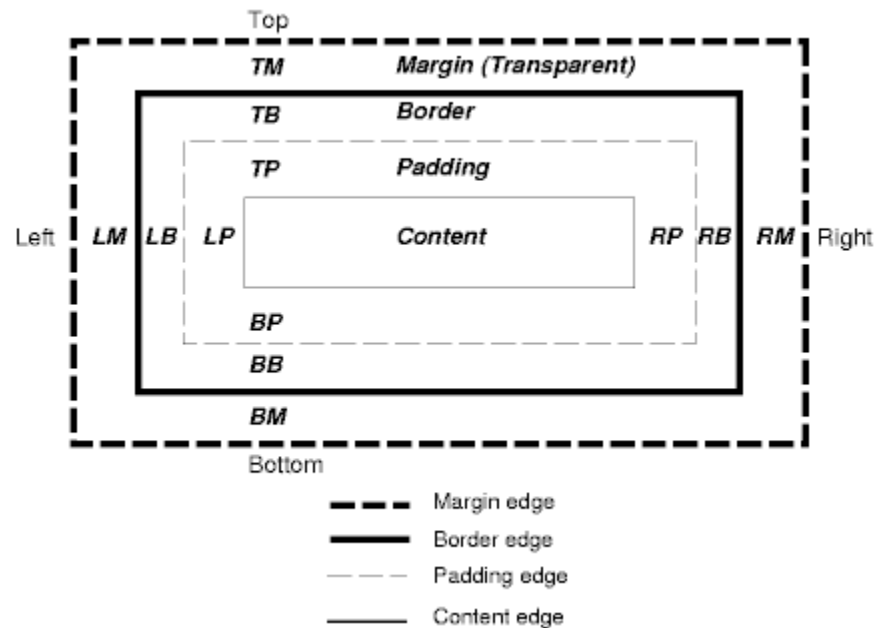


Demo

- [CSS/o2-selectors.html](#)

Box model

- Dla każdego elementu drzewa dokumentu jest tworzony prostokątny blok.
- Każdy blok zawiera cztery obszary:
 - content,
 - padding,
 - border,
 - margin



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 }

Przegląd właściwości



- Od CSS3 możemy definiować własne czcionki
 - Deklaracja

```
@font-face {  
  font-family: 'NazwaNowejCzcionki';  
  src: url('PTC55F-webfont.eot');  
  src: url('PTC55F-webfont.woff') format('woff'), ...  
  (dodatkowe pliki)  
  ... opcjonalnie font -weight, -style, -stretch,...  
}
```
 - Użycie

```
body { font-family: NazwaNowejCzcionki, Arial, sans-serif; }
```
- Generatory CSS i pliki czcionek
 - www.fontsquirrel.com
- Czcionki od Google
 - <https://fonts.google.com/>

Przegląd właściwości

- Formatowanie tekstu – czcionki
 - 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
 - text-shadow: (cienie wokół tekstu)
 - 2px /* horizontal */
 - 2px /* vertical */
 - 4px /* blur */
 - #444 /* color */



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

Demo

- [CSS3/o1-fonts.html](#)

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



- Nowości w konfiguracji tła w CSS3
 - Wiele obrazków
 - background:
 url(img1) [position],
 url(img2) [position],
 ...,
 #color;
 - background-repeat: no-repeat | repeat-y,...
 - background-origin: content-box | padding-box | border-box
 - background-size: length|percentage|cover|contain;

Przegląd właściwości

- Nowości w konfiguracji tła w CSS3

- Gradienty

- Liniowy

- ```
background: gradient(
 start-gradientu,
 kolor punkt-stopu,
 ...)
```

- Kołowy

- ```
background: gradient(  
    środek,  
    kształt [elipsa lub koło]  
    rozkład-kolorów-na-promieniu,  
    kolor punkt-stopu, ...);
```

- Fajny tutorial

- <http://jportal.pl/css3-gradienty.html>



Demo

- [CSS3/05-background.html](#)

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 (przezroczysty, 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

- Nowości w obramowaniach w CSS3
 - Właściwości
 - border-radius
 - box-shadow
 - border-image



Demo

- [CSS3/o6-border.html](#)

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 odgórnie
 - 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

Animacje

- Właściwość transition
 - Po przecinku lista właściwości wraz z czasem przejścia
 - Przykład:
 - transition: color 0.2s ease, border-bottom 0.2s ease, background-color 0.2s ease;



Demo

- [CSS3/07-transitions.html](#)
- [CSS3/AnimatedButtons](#)

Animacje

- Warto jeszcze zobaczyć na:
 - transform
 - https://www.w3schools.com/css/css3_2dtransforms.asp
 - animation
 - https://www.w3schools.com/css/css3_animations.asp

HTML Layouts

- Na czym polega budowanie layoutu
- Podstawowe techniki (i krótka historia)
 - HTML tables
 - CSS float property
 - CSS flexbox
 - CSS framework (np. Bootstrap)
- Problemy z mnogością rozdzielczości
 - Fixed width vs. Fluid
- Responsive Web Design
 - ... oraz Bootstrap będzie omówiony za kilka tygodni

Pozycjonowanie

- Pozycjonowanie jest określone następująco
 - position : static | relative | absolute | fixed | sticky
 - Sticky: niewspierane przez starsze przeglądarki
- 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)

Demo

- tabelki.html
- wykazy[1234].html
- wyswietlanie[12].html
- pozycjonowanie.html
 - Sticky:
https://www.w3schools.com/cssref/tryit.asp?filename=trycss_position_sticky
- menu.html
- formularze.html
- szablony/*

Flexible Box Layout

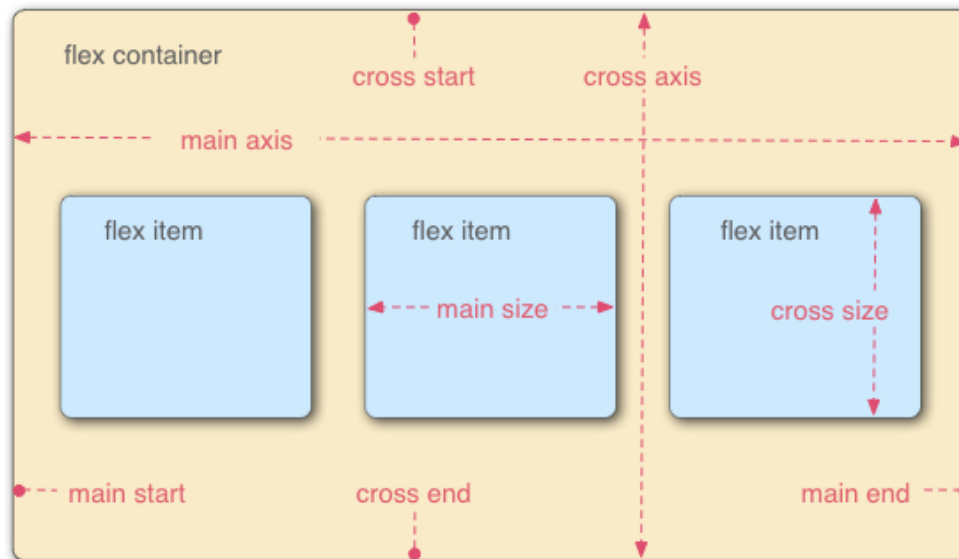
- Nowy sposób budowania layoutu strony
- Poziom wsparcia: <http://caniuse.com/#feat=flexbox>

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			51						
			52					4.4	
8	13	47	53	9.1		9.3		4.4.4	
11	14	48	54	10	40	10	all	52	53
		50	55	TP	41				
		51	56		42				
		52	57						

- Określa relacje pomiędzy parentem (flex container) a child items (flex items)
- Utworzenie kontenera flex
 - display: flex
- Terminologia
 - box (2009): <https://www.w3.org/TR/2009/WD-css3-flexbox-20090723/>
 - flexbox (2012): <https://www.w3.org/TR/2012/WD-css3-flexbox-20120322/>
 - flex (bieżąca): <https://drafts.csswg.org/css-flexbox/>

Flexible Box Layout

- Podstawowa terminologia



- Przejdźmy do szczegółów

- ...wspierając się obrazkami z

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Flexible Box Layout

- Układanie elementów
 - Kierunek układania
 - flex-direction: row | row-reverse | column | column-reverse
 - Sposób zawijania, gdy brakuje miejsca
 - flex-wrap: nowrap | wrap | wrap-reverse
 - Jedno property do obu powyższych, np.
 - flex-flow: column wrap | row-reverse wrap-reverse | ...
- Kolejność wyświetlania elementów
 - Pojęcie ***Ordinal Group***
 - Składnia: order: liczba
 - Określa nr grupy (domyślnie wszystkie elementy mają order: 0)
 - Uwaga: jest to grupa, może być zatem więcej niż 1 element

Flexible Box Layout

- Wielkość elementów: właściwość flex
 - Powiększanie, gdy jest wolne miejsce: flex-grow
 - Im większa liczba, tym bardziej element będzie powiększony
 - Wartość 0 oznacza brak powiększania
 - Pomniejszanie, gdy brak wolnego miejsca: flex-shrink
 - Im większa liczba, tym bardziej element będzie pomniejszony
 - Wartość wyjściowa: flex-basis (analogiczne do width)
 - Kombinacja wszystkich elementów: flex
 - Składnia: flex: flex-grow flex-shrink flex-basis
 - Domyślne: flex: 0 1 auto

Flexible Box Layout

- Wyrównywanie elementów
 - Trzy główne property
 - justify-content: flex-start | flex-end | center | space-between | space-around
 - rozłożenie elementów w kontenerze względem main-axis
 - align-items: flex-start | flex-end | center | baseline | stretch
 - rozłożenie elementów w kontenerze względem main-axis
 - align-content: flex-start | flex-end | center | space-between | space-around | stretch
 - rozłożenie wierszy/kolumn (lines) kiedy jest ich więcej
 - nie ma znaczenia, gdy jest tylko jeden wiersz/kolumna (line)
 - Można też nadpisać align-items dla wybranego elementu poprzez property align-self

Flexible Box Layout

■ Do przeczytania:

- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS Flexible Box Layout/Using CSS flexible boxes](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Using_CSS_flexible_boxes)
- <https://www.w3.org/TR/css-flexbox-1/>
- <http://blog.end3r.com/113/css3-flexbox-flexible-box-model/>

Demo

- [CSS3/03a-flex-grow.html](#)
- [CSS3/03b-flex-ordinal-group.html](#)
- [CSS3/03c-flex-main-layout.html](#)

Szpalty

- Multiple columns
 - Dowolny blok można podzielić na szpalty
 - Właściwości:
 - column-count
 - column-gap
 - column-rule
- Do poczytania:
 - <https://css-tricks.com/almanac/properties/c/columns/>

Demo

- [CSS3/o4-multiplecolumns.html](#)

Cross browser compatibility

- Problem ze wsparciem w przeglądarkach
 - Gdzie co działa: <http://caniuse.com/>
- Przydatny skrypt: reset.css
 - <https://meyerweb.com/eric/tools/css/reset/>
- IE Conditional Comments
 - <https://www.sitepoint.com/internet-explorer-conditional-comments/>
- Prefix free plugin
 - <https://leaverou.github.io/prefixfree/>