

JAK STWORZYĆ STRONĘ RESPONSYWNA?

HTML+ CSS3 (MediaQueries)!

Ustala się pewną liczbę zakresów wielkości/rozdzielczości okna przeglądarki i dla każdego z nich ustawia się odrębny wygląd za pomocą arkuszy stylów CSS (ang. *Cascading Style Sheet*)

HTML - METATAG *VIEWPORT*

- Metatag `viewport` określa sposób wyświetlania strony na urządzeniach mobilnych – pozwala ustawić opcje skalowania, w tym także domyślne przybliżenie.
- Znacznik ten, podobnie inne meta-znaczniki musi znajdować się w sekcji `head` dokumentu HTML, a jego składnia wygląda następująco:

```
<meta name="viewport" content="...">
```

- Wartością atrybutu `content` powinna być lista par dyrektywa-wartość określających sposób wyświetlania strony na urządzeniach mobilnych. W polu tym możemy wpisać wiele różnych dyrektyw, oddzielając je przecinkami.

DOSTĘPNE ATRYBUTY DLA *VIEWPORT*

- `width`, `height` – określają szerokość i wysokość wyświetlanego obszaru strony. Można podać wartość liczbową (w pikselach), np.

```
<meta name="viewport" content="width=320, height=480">
```

- Dla atrybutu `width` możemy również podać specjalną wartość `device-width`, której użycie powoduje, że szerokość obszaru wyświetlania jest równa szerokości ekranu urządzenia (tzn. strona jest rozciągnięta w 100% na ekranie urządzenia)
- Dla dyrektywy `height` możemy użyć analogicznej wartości, `device-height`
- `initial-scale` – w atrybucie tym ustawiamy domyślne przybliżenie (zoom) dla strony. Wartość ta jest zazwyczaj zależna od przeglądarki, lecz możemy wymusić by nasza mobilna wersja strony domyślnie wyświetlała się np. w 10-krotnym przybliżeniu:

```
<meta name="viewport" content="initial-scale=10">
```

Możemy ją również pokazać w całej swej okazałości, bez skalowania:

```
<meta name="viewport" content="initial-scale=1">
```

DOSTĘPNE ATRYBUTY DLA *VIEWPORT*

- `minimum-scale`, `maximum-scale` – atrybuty te określają minimalne i maksymalne przybliżenie/oddalenie strony w obszarze przeglądarki. Minimalny współczynnik skalowania nie może być mniejszy niż 0, natomiast maksymalny nie może być większy niż 10. Prosty przykład użycia:

```
<meta name="viewport" content="minimum-scale=1,  
maximum-scale=5">
```

- `user-scalable` – pozwala włączyć, bądź też wyłączyć możliwość skalowania strony przez użytkownika. Jako wartość przyjmuje angielskie *yes* lub *no*.

```
<meta name="viewport" content="user-scalable=yes">
```

KIEDY UŻYWAĆ METATAGU *VIEWPORT*?

- Znacznik ten mówi o tym, jak strona powinna być wyświetlana na urządzeniu mobilnym. Zatem jeśli strona nie jest dostosowana do urządzeń mobilnych, **można całkowicie pominąć użycie meta-tagu `viewport`.**
- Jeśli strona jest dostosowana do urządzeń mobilnych, należy użyć:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

KIEDY UŻYWAĆ METATAGU VIEWPORT?

Strona mobilna bez metatagu *viewport*



Strona mobilna z metatagiem *viewport*



Fragment pliku HTML:

```
...
<div id="blok-gl">
  <div id="blok-zaw">
    <h2>TEMAT 1</h2> ...
  </div>

  <div id="kolumna-lewa">
    <h2>Temat 2</h2> ...
  </div>

  <div id="kolumna-prawa">
    <h2>Temat 3</h2> ...
  </div>
</div>
...
```

Fragment pliku CSS:

```
html,body{ background... font...}
p, blockquote, pre, code, ol, ul{}
/* Elementy strukturalne */
#blok-gl{
  width: 80%; margin: 0 auto;
  background: #fff; padding: 20px; }
#blok-zaw{
  width: 54%; float: left;
  margin-right: 3%; }
#kolumna-lewa{
  width: 20%; float: left;
  margin-right: 3%; }
#kolumna-prawa{
  width: 20%; float: left; }
```

Zmiana struktury dokumentu – widok standardowy

TEMAT 1

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego drukarza do wypełnienia tekstem próbnej książki. Pięć wieków później zaczął być używany przemyśle elektronicznym, pozostając praktycznie niezmienionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageMaker.

Ogólnie znana teza głosi, iż użytkownika może rozpraszać zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.

Temat 2

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.

Temat 3

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego:

Więcej o **float** na stronie:
<http://kurs.browsehappy.pl/CSS/Float>

WŁAŚCIWOŚĆ FLOAT

- Za pomocą właściwości float możemy sprawić, że interesujący nas element HTML stanie się pływającym w obszarze zawartości swojego elementu rodzica elementem HTML.
- Każdy pływający element HTML zazwyczaj zachowuje cechy podobne do cech elementu HTML typu block, dzięki czemu między innymi wiemy, na jakie właściwości CSS może reagować dany element HTML
- Treść do wyświetlenia występująca w kodzie HTML po pływającym elemencie HTML może opływać ten pływający element HTML, ale tylko w sytuacji, gdy wystarczy miejsca w danym obszarze zawartości
- Pływający element HTML występuje w obszarze zawartości swojego elementu rodzica, lecz pływający element HTML nie może bezpośrednio powiększać swoim rozmiarem rozmiaru swojego elementu rodzica
- Domyślnie element HTML nie jest pływającym w obszarze zawartości swojego elementu rodzica elementem HTML.

FLOAT: LEFT

- Za pomocą wartości left właściwości float możemy sprawić, że interesujący nas element HTML stanie się pływającym w obszarze zawartości swojego elementu rodzica elementem HTML.
- Element HTML będzie pływał w obszarze zawartości swojego elementu rodzica w kierunku lewej krawędzi liniowej swojego elementu rodzica.

PRZYKŁAD FLOAT: LEFT

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Przykład float</title>
</head>
<body>
    <div id="rodzic">
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        <div id="dziecko">div#dziecko</div>
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    </div>
</body>
</html>
```

```
html { font-size:16px;}

#rodzic { font-size:1.3em;}

#dziecko {
    float:left;
    height:100px;
    background-color:red;
}
```

[illegible]

FLOAT: RIGHT

- Za pomocą wartości right właściwości float możemy sprawić, że interesujący nas element HTML stanie się pływającym w obszarze zawartości swojego elementu rodzica elementem HTML.
- Element HTML będzie pływał w kierunku prawej krawędzi liniowej swojego elementu rodzica.

PRZYKŁAD FLOAT: RIGHT

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Przykład float</title>
</head>
<body>
  <div id="rodzic">
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    <div id="dziecko">div#dziecko</div>
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  </div>
</body>
</html>
```

```
html { font-size:16px;}

#rodzic { font-size:1.3em;}

#dziecko {
  float:right;
  height:100px;
  background-color:red;
}
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAA
AAAAAAAAAA
AAAAAAAAAA
AAAAAAAAAA
AAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```



Fragment pliku HTML:

```
...
<div id="blok-gl">
  <div id="blok-zaw">
    <h2>TEMAT 1</h2> ...
  </div>

  <div id="kolumna-lewa">
    <h2>Temat 2</h2> ...
  </div>

  <div id="kolumna-prawa">
    <h2>Temat 3</h2> ...
  </div>
</div>
...
```

Fragment pliku CSS:

```
#blok-gl{ width: 90%; }

#blok-zaw{ width: 100%; }

#kolumna-lewa{
  width: 100%; clear: both;
}

#kolumna-prawa{
  width: 100%; clear: both;
}
```

Zmiana struktury dokumentu – widok mobilny

TEMAT 1

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego drukarza do wypełnienia tekstem próbnej książki. Pięć wieków później zaczął być używany przemyśle elektronicznym, pozostając praktycznie niezmiennym. Spopularyzował się w latach 60. XX w..

Ogólnie znana teza głosi, iż użytkownika może rozpraszać zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającą wygląda to „zbyt czytelnie” po polsku.

Temat 2

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron uż

Temat 3

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputedfgsdgfrgdsge

Więcej o **clear** na stronie:
<http://kurs.browsehappy.pl/CSS/Float>

WŁAŚCIWOŚĆ CLEAR

- Za pomocą właściwości clear możemy określić czy interesujący nas element HTML może opływać pływający element HTML.

CLEAR: NONE

- Wartość none jest domyślną wartością właściwości clear.
- Domyślnie element HTML może opływać pływający element HTML.

PRZYKŁAD CLEAR: NONE

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Przykład float</title>
</head>
<body>
  <div id="rodzic">
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    <div id="dziecko-1">
      div#dziecko-1<br>float:left;</div>
      <div id="dziecko-2">
        div#dziecko-2<br>float:right;</div>
        BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
        <div id="dziecko-3">div#dziecko-3</div>
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    </div>
  </body>
</html>
```

```
html {font-size:16px;}
#rodzic {font-size:1.3em;}
#dziecko-1 {float:left;
            height:500px;
            background-color:blue;}
#dziecko-2 {float:right;
            height:400px;
            background-color:red;}
#dziecko-3 {background-color:green;}
```

AA
AA

div#dziecko-1	AAAAAAABBBBE	div#dziecko-2
div#dziecko-2	BBBBBBBBBBBBBE	div#dziecko-3
div#dziecko-3	CCCCCCCCCCCCCC	float:right
float:left	float:center	float:right

CLEAR: LEFT

- Za pomocą wartości left właściwości clear możemy sprawić, że interesujący nas element HTML nie będzie mógł opływać elementu HTML posiadającego właściwość float wraz z wartością left.

PRZYKŁAD CLEAR: LEFT

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Przykład float</title>
</head>
<body>
  <div id="rodzic">
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    <div id="dziecko-1">
      div#dziecko-1<br>float:left;</div>
    <div id="dziecko-2">
      div#dziecko-2<br>float:right;</div>
      BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
    <div id="dziecko-3">div#dziecko-3</div>
    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  </div>
</body>
</html>
```

```
html {font-size:16px;}
#rodzic {font-size:1.3em;}
#dziecko-1 {float:left;
  height:500px;
  background-color:blue;}
#dziecko-2 {float:right;
  height:400px;
  background-color:red;}
#dziecko-3 {clear:left;
  background-color:green;}
```

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA



CLEAR: RIGHT

- Za pomocą wartości right właściwości clear możemy sprawić, że interesujący nas element HTML nie będzie mógł opływać elementu HTML posiadającego właściwość float wraz z wartością right.

PRZYKŁAD CLEAR: RIGHT

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Przykład float</title>
</head>
<body>
  <div id="rodzic">
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    <div id="dziecko-1">
      div#dziecko-1<br>float:left;</div>
      <div id="dziecko-2">
        div#dziecko-2<br>float:right;</div>
        BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
        <div id="dziecko-3">div#dziecko-3</div>
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      </div>
    </body>
  </html>
```

```
html {font-size:16px;}
#rodzic {font-size:1.3em;}
#dziecko-1 {float:left;
  height:500px;
  background-color:blue;}
#dziecko-2 {float:right;
  height:400px;
  background-color:red;}
#dziecko-3 {clear:right;
  background-color:green;}
```

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

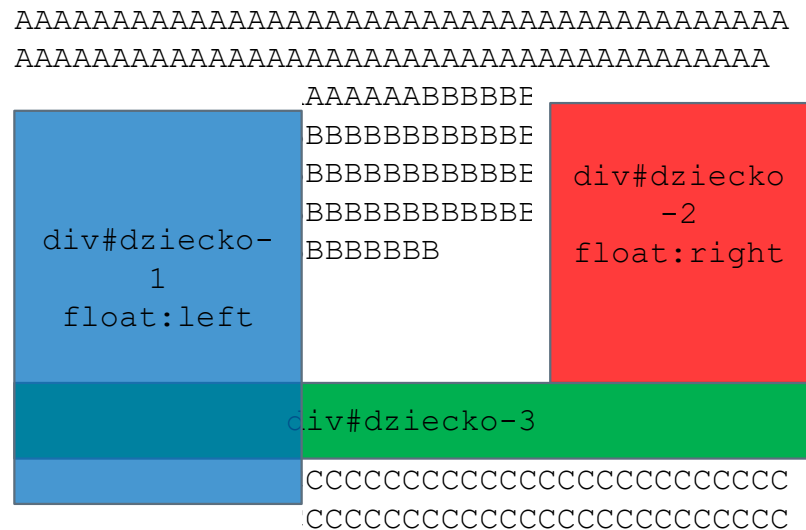
AAAAAABBBBBBE
BBBBBBBBBBBBBE
BBBBBBBBBBBBBE
BBBBBBBBBBBBBE
BBBBBBBBB

div#dziecko-1
float:left

div#dziecko-2
float:right

div#dziecko-3

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC



CLEAR: BOTH

- Za pomocą wartości both właściwości clear możemy sprawić, że interesujący nas element HTML nie będzie mógł opływać żadnego pływającego elementu HTML.

PRZYKŁAD CLEAR: BOTH

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Przykład float</title>
</head>
<body>
  <div id="rodzic">
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    <div id="dziecko-1">
      div#dziecko-1<br>float:left;</div>
      <div id="dziecko-2">
        div#dziecko-2<br>float:right;</div>
       BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
        <div id="dziecko-3">div#dziecko-3</div>
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      </div>
    </body>
  </html>
```

```
html {font-size:16px;}
#rodzic {font-size:1.3em;}
#dziecko-1 {float:left;
  height:500px;
  background-color:blue;}
#dziecko-2 {float:right;
  height:400px;
  background-color:red;}
#dziecko-3 {clear:both;
  background-color:green;}
```

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBBBBBBBBBB

BBBBBBBBBBB
BBBBBBBBBBB
BBBBBBBBBBB
BBBBBBBBBBB
BBBBBBBBBBB
BBBBBBBBB

div#dziecko-1
float:left

div#dziecko-2
float:right

div#dziecko-3

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

Fragment pliku HTML:

```
<p class="pasek">
  <a href="#">Lewa kolumna</a> |
  <a href="#">Prawa kolumna</a>
</p>

<div id="blok-zaw">
  <h2>TEMAT 1</h2>
</div>

<div id="kolumna-lewa">
  <h2>Temat 2</h2>
</div>

<div id="kolumna-prawa">
  <h2>Temat 3</h2>
</div>
```

Fragment pliku CSS:

```
#blok-zaw{
  width: 54%;
  float: left;
  margin-right: 3%; }

#kolumna-lewa{
  width: 20%;
  float: left;
  margin-right: 3%; }

#kolumna-prawa{
  width: 20%;
  float: left; }

.pasek{display: none;}
```

Ukrywanie zawartości – widok standardowy

TEMAT 1

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego drukarza do wypełnienia tekstem próbnej książki. Pięć wieków później zaczął być używany przemyśle elektronicznym, pozostając praktycznie niezmienionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageMaker.

Ogólnie znana teza głosi, iż użytkownika może rozpraszać zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.

Temat 2

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.

Temat 3

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego:

Fragment pliku HTML:

```
<p class="pasek">
  <a href="#">Lewa kolumna</a> |
  <a href="#">Prawa kolumna</a>
</p>

<div id="blok-zaw">
  <h2>TEMAT 1</h2>
</div>

<div id="kolumna-lewa">
  <h2>Temat 2</h2>
</div>

<div id="kolumna-prawa">
  <h2>Temat 3</h2>
</div>
```

Fragment pliku CSS:

```
#blok-zaw{
  width: 100%; }

#kolumna-lewa{
  display: none; }

#kolumna-prawa{
  display: none; }

.pasek {display: inline;}
```

Ukrywanie zawartości – widok mobilny

[Lewa kolumna](#) | [Prawa kolumna](#)

TEMAT 1

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego drukarza do wypełnienia tekstem próbnej książki. Pięć wieków później zaczął być używany przemyśle elektronicznym, pozostając praktycznie niezmienionym. Spopularyzował się w latach 60. XX w. .

Ogólnie znana teza głosi, iż użytkownika może rozpraszać zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającą wygląda to „zbyt czytelnie” po polsku.

JAK AUTOMATYCZNIE DOSTOSOWAĆ LAYOUT STRONY DO URZĄDZENIA?

TEMAT 1

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego drukarza do wypełnienia tekstem próbnej książki. Pięć wieków później zaczął być używany przemyśle elektronicznym, pozostając praktycznie niezmiennym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageMaker.

Ogólnie znana teza głosi, iż użytkownika może rozpraszać zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.

Temat 2

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.

Temat 3

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.

TEMAT 1

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego drukarza do wypełnienia tekstem próbnej książki. Pięć wieków później zaczął być używany przemyśle elektronicznym, pozostając praktycznie niezmiennym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageMaker.

Ogólnie znana teza głosi, iż użytkownika może rozpraszać zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.

Temat 2

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.

Temat 3

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego, że wygląda to „zbyt czytelnie” po polsku.



TEMAT 1

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego drukarza do wypełnienia tekstem próbnej książki. Pięć wieków później zaczął być używany przemyśle elektronicznym, pozostając praktycznie niezmiennym. Spopularyzował się w latach 60. XX w.

Ogólnie znana teza głosi, iż użytkownika może rozpraszać zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego wygląda to „zbyt czytelnie” po polsku.

Temat 2

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron u.

Temat 3

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego enionym. Spopularyzował się w latach 60. XX w. wraz z publikacją arkuszy Letrasetu, zawierających fragmenty Lorem Ipsum, druków na komputerach osobistych, jak Aldus PageM zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron u.



Lewa kolumna | Prawa kolumna

TEMAT 1

Lorem Ipsum jest tekstem stosowanym jako przykładowy wypełniacz w przemyśle poligraficznym. Został po raz pierwszy użyty w XV w. przez nieznanego drukarza do wypełnienia tekstem próbnej książki. Pięć wieków później zaczął być używany przemyśle elektronicznym, pozostając praktycznie niezmiennym. Spopularyzował się w latach 60. XX w.

Ogólnie znana teza głosi, iż użytkownika może rozpraszać zrozumiała zawartość strony, kiedy ten chce zobaczyć sam jej wygląd. Jedną z mocnych stron używania Lorem Ipsum jest to, że ma wiele różnych „kombinacji” zdań, słów i akapitów, w przeciwieństwie do zwykłego: „tekst, tekst, tekst”, sprawiającego wygląda to „zbyt czytelnie” po polsku.

CSS3 I *MEDIA QUERIES*!

- W CSS3 dodano *zapytania mediów* (ang. ***media queries***), co umożliwiło tworzenie stron www pod kątem różnych rozmiarów okna przeglądarki.
- *Media queries* pozwalają precyzyjnie dopasować arkusz CSS nie tylko do typu czy grupy mediów, ale przede wszystkim do możliwości wyświetlacza urządzenia.

MEDIA QUERY W CSS2

- Media Query – w CSS2 to nic innego jak wybór medium, dla którego ma być serwowany bieżący plik CSS. Czyli możemy zdecydować, które pliki wyświetlać w wersji do druku, a które dla np. odbiorników telewizyjnych.
- Typy mediów:
 - **all** - dla wszystkich urządzeń
 - **aural** - syntezatory mowy (*słuchowe arkusze stylów*)
 - **braille** - urządzenia do czytania *braillem* (dla niewidomych)
 - **embossed** - drukarki *brailła*
 - **handheld** - urządzenia ręczne (zwykle mały ekran, czarno-białe)
 - **print** - materiały drukowane nieprzezroczyste i podgląd wydruku
 - **projection** - prezentacje projektorowe lub nadruki na folii rzutnikowej
 - **screen** - kolorowy ekran komputerowy
 - **tv** – odbiorniki telewizyjne

MEDIA QUERY CSS2– GDZIE?

Media Query w pliku CSS:

- wykorzystuje się dyrektywę **@media**

```
@media print { /* style dla drukarek a aktualnym pliku*/ }
```

Media Query w pliku HTML:

- wykorzystuje się atrybut **media** elementu **LINK** wskazujący zewnętrzny plik arkusza stylów dla konkretnego typu medium

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
  <head>
    <link rel="stylesheet" type="text/css" media="print" href="arkusz.css" />
  </head>
  <body>
    <p>Ciało dokumentu...</p>
  </body>
</html>
```

MEDIA QUERY W CSS3

- Wraz z nadejściem CSS3 rozszerzono możliwości Media Queries.
- W CSS3 można oprócz wskazania typu mediów dodatkowo decydować o dołączaniu odpowiedniego pliku/kodu CSS na podstawie **rozdzielczości ekranu, orientacji wyświetlacza, kolorów**, etc.

MEDIA QUERY (CSS3) W PLIKU HTML

```
<link rel="stylesheet" media="screen and (min-width: 800px) and (min-height: 600px)" href="normal.css">
```

Interpretacja:

Ustawiony jest warunek, by załadować plik normal.css tylko, gdy:

- stronę otworzono z urządzenia posiadającego ekran
- minimalna szerokość ekranu wynosi 800 pikseli
- minimalna wysokość ekranu wynosi 600 pikseli

Jeśli urządzenie, na którym strona została otworzona nie spełnia powyższych kryteriów, wtedy plik normal.css **nie zostanie załadowany**.

Oczywiście takich warunków może być więcej!

MEDIA QUERY (CSS3) W PLIKU CSS

Ten sam przykład:

- stronę utworzono z urządzenia posiadającego ekran
- minimalna szerokość ekranu wynosi 800 pikseli
- minimalna wysokość ekranu wynosi 600 pikseli

a) z dedykowanymi stylami od razu w pliku CSS:

```
@media screen and (min-width: 800px) and (min-height: 600px) {  
    /* zawartość pliku normal.css */  
}
```

b) z dedykowanymi stylami w pliku zewnętrznym (normal.css):

```
@import url(normal.css) screen and (min-width: 800px) and (min-height:  
600px); }
```

NOWE WARUNKI W CSS3 – WYMIARY EKRANU

- Można zdecydować, na ekranach o jakiej rozdzielczości wczytywać nasz kod CSS. Do wyboru jest oczywiście długość i szerokość ekranu (mowa o rozdzielczości).
- Możemy określać zarówno minimalne, jak i maksymalne wielkości, a atrybuty za to odpowiedzialne to: `min-width`, `max-width`, `min-height`, `max-height`.
- Nie trzeba kierować się wyłącznie rozdzielczością ekranu. Często bywa tak, że użytkownik ma dużo mniejszy obszar „widoczności” strony WWW, aniżeli rozdzielczość ekranu. Wynika to z faktu, iż różne programy mogą pojawiać się jako toolbar po prawej lub lewej stronie ekranu.
- W tym celu powstały kolejne atrybuty dla Media Queries pozwalające rozpoznać „prawdziwą” szerokość/wysokość wyświetlanej strony, tj. `device-width`, `device-height`, `min-device-width`, `max-device-width`, `min-device-height` oraz `max-device-height`.

NOWE WARUNKI W CSS3 – ORIENTACJA EKRANU

Dla orientacji ekranu mamy do wyboru dwa atrybuty:

- `orientation: portrait` – orientacja pionowa
- `orientation: landscape` – orientacja pozioma

NOWE WARUNKI W CSS3 – *ASPECT RATIO*

Aspect Ratio jest to stosunek wysokości do szerokości ekranu (lub obszaru aktualnie wyświetlonej strony).

Przykład, w którym szerokość ekranu jest dwa razy większa od wysokości „widocznego” obszaru roboczego strony:

```
@media all and (device-aspect-ratio: 32/16) { ... }
```

Przy wyborze *Aspect Ratio* mamy do dyspozycji następujące atrybuty: `aspect-ratio` oraz `device-aspect-ratio`.

NOWE WARUNKI W CSS3 – KOLORY

Można też warunkowo korzystać ze stylów CSS w zależności od kolorów, jakie obsługuje urządzenie użytkownika. Do dyspozycji są następujące atrybuty: `color`, `min-color`, `max-color`, `color-index`, `min-color-index`, `max-color-index`, `monochrome`, `min-monochrome`, `max-monochrome`.

Kilka praktycznych przykładów użycia:

```
@media (color) { /* urządzenie obsługuje kolory */ }

@media (min-color-index: 256) { /* głębokość kolorów -
minimum 256 kolorów */ }

@media (monochrome) { /* urządzenie jest
monochromatyczne - składa się jedynie z barw prostych */
}
```

RESPONSYWNY ROZMIAR TEKSTU

- Rozmiar tekstu można ustawiać za pomocą jednostki "vw", która oznacza "viewport width".
- Viewport to rozmiar okna przeglądarki. 1vw = 1% szerokości viewport. Jeśli viewport jest szeroki na 50cm, 1vw oznacza 0.5cm.
- Przykład:

```
<h1 style="font-size:10vw">Hello World</h1>
```

MEDIA QUERY – GDZIE? W CSS!

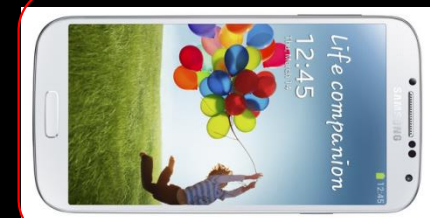
```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="pl" lang="pl">
<head>
<link
  rel="Stylesheet"
  type="text/css"
  href="arkusz.css" />
</head>
<body>
<p>Ciao! dokumentu...</p>
</body>
</html>
```

arkusz.css

```
/* Smartfony (portrait i
landscape) ----- */
@media screen and (max-width:
600px)
{
  /* Style */
}
```

```
/* Smartfony (landscape) -----
----- */
@media screen and
(min-width: 600px) and
(orientation:landscape)
{
  /* Style */
}
```

```
/* Smartfony (portrait) -----
----- */
@media screen and
(min-width: 600px) and
(orientation:portrait)
{
  /* Style */
}
```



TYPOWE MEDIA QUERY DLA RWD W PLIKU CSS

Jeden plik CSS dla różnych urządzeń/wyświetlaczy

```
/* Smartfony małe(portrait i landscape) ----- */
@media screen and (max-width: 600px) {
/* Style */
}

/* Smartfony duże(landscape) ----- */
@media screen and (min-width: 600px) and
  (orientation:landscape) {
/* Style */
}

/* Smartfony duże(portrait) ----- */
@media screen and (min-width:600px) and
  (orientation:portrait) {
/* Style */
}
```

MEDIA QUERY – GDZIE? W HTML!

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="pl" lang="pl">
<head>
```

```
<link rel="stylesheet"
media="screen
and (max-width: 600px)"
href="small.css" />
```

```
<link rel="stylesheet"
media="screen
and (min-width: 600px)
and (orientation:landscape)"
href="large_land.css" />
```

```
<link rel="stylesheet"
media="screen
and (min-width:600px)
and (orientation:portrait)"
href="large_port.css" />
```

```
</head>
<body>
<p>Ciao! dokumentu...</p>
</body>
</html>
```

small.css

```
/* Style dla smartfonów
(portrait i landscape) -----
--- */
```

large_land.css

```
/* Smartfony (landscape) -----
----- */
```

large_port.css

```
/* Smartfony (portrait) -----
----- */
```



TYPOWE MEDIA QUERY DLA RWD W PLIKU HTML

Plik HTML ze wskazaniem na różne pliki CSS dla różnych urządzeń/wyświetlaczy:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<head>

  <link rel="stylesheet" media="screen and (max-width: 600px)"
  href="small.css" />

  <link rel="stylesheet" media="screen and (min-width: 600px) and
  (orientation:landscape)" href="large_land.css" />

  <link rel="stylesheet" media="screen and (min-width:600px) and
  (orientation:portrait)" href="larg_port.css" /> </head>

<body>
  <p>Ciało dokumentu...</p>
</body>
</html>
```


PODMIANA GRAFIKI, ZNACZNIK <PICTURE>

- <picture> umożliwia łatwe określenie źródła grafiki
- Zamiast skalować pojedynczy obraz do dostępnej powierzchni, można przygotować wiele plików z grafiką o odpowiednim rozmiarze.
- Element <picture> zawiera znaczniki <source> (jeden lub więcej) i jeden znacznik .
- Element <source> ma następujące atrybuty::
 - srcset (wymagany) - określa URL obrazu, który ma być wyświetlany
 - media – zawiera poprawne *media query* normalnie definiowane w CSS
 - sizes – określa pojedynczą wartość width
 - type – definiuje typ MIME

PODMIANA GRAFIKI, ZNACZNIK <PICTURE>

- Przeglądarka użyje pierwszego pasującego do wartości atrybutu *media* elementu <source> i zignoruje następne.
- Element jest wymagany jako ostatni element potomny znacznika <picture>. Znacznik został zachowany: dla wstecznej kompatybilności z przeglądarkami, które nie obsługują elementu <picture> lub w przypadku, gdy nie dopasuje się żaden z elementów <source>.

PODMIANA GRAFIKI, PRZYKŁAD

```
<picture>  
  
  <source media="(min-width: 680px)" srcset="img_duzy.jpg">  
  
  <source media="(min-width: 460px)" srcset="img_maly.jpg">  
  
    
  
</picture>
```

STRONY RESPONSYWNE – DOBRE PRAKTYKI

Należy:

- **unikać popupów i otwierania linków w nowej zakładce** na stronie w wersji dla małych rozdzielczości (smartfony);
- **pisać kod strony poprawnie i zgodnie ze standardami** - trzymanie się wytycznych daje gwarancję, że strona będzie się poprawnie wyświetlać w bardzo różnych i egzotycznych przeglądarkach;
- **ostrożnie używać skryptów JavaScript** - nie każdy skrypt zadziała w każdej mobilnej przeglądarce;
- **unikać stosowania stylu *position: fixed*** - nie działa w większości mobilnych przeglądarek.