

**3.01.2022**

**Responsywność strony** - serwis WWW płynnie dostosowywał się do rozdzielczości ekranu każdego urządzenia komputerowego.

Obecnie z Internetu korzystają użytkownicy:

- komputerów stacjonarnych,
- notebooka,
- tabletu, ( dwa położenia → poziom, pion)
- smartfonów, ( dwa położenia → poziom, pion)
- innych urządzeń mobilnych.

Musimy pamiętać o tym, aby serwis WWW płynnie dostosowywał się do rozdzielczości ekranu każdego urządzenia.

**Responsive Web Design (RWD)** to nic innego, jak projekt strony internetowej, której układ automatycznie dostosowuje się do wielkości ekranu urządzenia. Wyświetlane na ekranie treści skalują się tak, aby były one wygodne w odbiorze dla użytkownika. Strony responsywne cechuje uniwersalność, czego dowodem jest ich wygląd na smartfonie czy tablecie.

Testowanie responsywności strony → test RWD

<https://it-develop.pl/mobile-rwd-test.html>

#### Dlaczego warto zmienić stronę na responsywną?

1. Przeglądarka Google wyżej indeksuje responsywne strony. Oznacza to, że przestarzałe strony muszą liczyć się z mniejszą liczbą odwiedzin.
2. Ponadto, posiadanie reponsywnej strony WWW będzie oznaczało dla użytkownika, że dany sklep/firma/portal jest na czasie. Materiały, które odpowiednio skalują się do rozmiaru ekranu umożliwiają dokładniejsze i przyjemniejsze przeglądanie. Dzięki temu, firma zyska wówczas wielu nowych gości i klientów.
3. Responsywne strony internetowe stanowią dobrą alternatywę dla mobilnych stron internetowych, które pochłaniają więcej czasu i pieniędzy. Mobilna strona internetowa jest witryną dodatkową, która uaktywnia się urządzeniach przenośnych (więc trzeba ponieść dodatkowe koszty związane z jej tworzeniem i utrzymaniem), natomiast w przypadku stron responsywnych, wystarczy jedna wersja.
4. Całodobowa obecność w sieci sprawiła, że konsumenci dokonują zakupów przez urządzenia mobilne, więc warto nie zostać w tyle i zmienić swoją stronę na RWD.

#### **Strony mobilne a responsywne**

Wersje mobilne stron internetowych często mają przedrostek m, który poprzedza adres URL np.

<http://m.wp.pl>  
<http://m.zalando.pl>.

### **Czy strony responsywne mają jakieś wady?**

Strony RWD posiadają obecnie jeden **niewielki minus**. Mianowicie, jeżeli użytkownik zażyczy sobie, aby jego obecna strona została „przemieniona” w stronę responsywną, to musi liczyć się z kosztami, które niestety przewyższają stworzenie strony internetowej od zera. Jest to kompletnie nieopłacalne posunięcie, chyba że ktoś jest mocno przywiązany do swojej obecnej strony internetowej i jest gotów poświęcić znaczną ilość pieniędzy. W innym przypadku, lepiej zaprojektować stronę od nowa, wprowadzić pewną świeżość i odmłodzić wygląd witryny. Takie zmiany zawsze wychodzą na korzyść użytkownika. Ponadto, z roku na rok zmienia się podejście do designu stron www, dlatego lepiej jest zaprojektować zupełnie coś nowego, na czasie, poprawić nawigację na stronie tak, aby serwis stał się naprawdę łatwy w obsłudze.

## Jak używać @media query? Kilka przydatnych przykładów.

Media query to “zmienna” wprowadzona w CSS3. Obecnie spotykana jest ona na każdej stronie typu RWD. Praktycznie bez @media query RWD by nie istniało. Czasami stosuje się tylko określenie @media bez dodatku query (zapytanie, pytanie).

przykład:

```
@media only screen and (max-width: 768px) {  
  /* For mobile phones: */  
  [class*="col-"] {  
    width: 100%;  
  }  
}
```

## Co zatem robi @media?

Media to swego rodzaju zapytanie – warunek informujący przeglądarkę o tym jaki kod CSS ma użyć w konkretnym przypadku np. inny dla:

1. komputera stacjonarnego
2. laptopa
3. tabletu
4. komórki

Czyli rozpoznaje na jakim urządzeniu pracuje odbiorca i strona dobiera ułożenie oraz parametry strony w zależności od urządzenia odbiorczego.

## Jakie wartości może przyjmować @media?

Tabela przedstawia dostępne oznaczenia typów mediów.	
Oznaczenie	Przeznaczenie
all	dla każdego typu urządzeń
aural	tylko dla syntezy mowy i dźwięku
braille	dla urządzeń przeznaczonych dla niewidomych, które umożliwiają czytanie
embossed	dla drukarek braila
handheld	bezprzewodowe urządzenia ręczne
print	dla podglądu wydruku oraz materiałów, które zostaną wydrukowane
projection	dla prezentacji projektorowych
screen	dla komputerów z kolorowym ekranem
tty	dla dalekopisów, terminali albo przenośnych urządzeń z ograniczonymi możliwościami wyświetlania
tv	dla telewizora

## Zadanie 1.

Wykonaj szkielet zaliczeniowy zadań w JS.

Przyciski w będą działać po wykonaniu zadań.



Uwaga: Przyciski Kod2, Kod3, Kod4 wyświetlają:

1. plik \*.html
2. plik \*.css
3. oraz wyłuczenie jako tekst ( można użyć komentarza)

## Zadanie 2.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Darmowy Kurs CSS</title>
    <style>
      @media screen {
        div {
          background-color:red;
        }
      }
    </style>
  </head>
  <body>
    <div>to jest przykładowy element - div</div>
  </body>
</html>
```

Wytłumaczenie:

Reguła **@media screen** mówi przeglądarce internetowej, że czerwony kolor tła dla elementu div ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w komputerze z kolorowym ekranem (screen).

===== Koniec Zadania2=====

### Zadanie 3.

Posługując się Zadaniem 1 oraz Zadaniem 2 (tabela poniżej) oraz tabel powyżej Zadania 1.

Wykonaj zadanie:

1. Wykonaj definicję w CSS zewnętrznym, nazwa pliku CSS → nazwisko\_ucznia.css (bez polskich liter, małymi literami).
2. gdy strona internetowa zostanie wyświetlona jako: prezentacja projektorowych, lub dla komputerów z kolorowym ekranem lub dla dalekopisów, terminali albo dla telewizora to element div zielony w pozostałych przypadkach czerwony.

Wy tłumaczenie:

@media może znaleźć się również w definicji pliku **CSS**.

```
div {  
  background-color:gold;  
}
```

```
@media handheld, tv {  
  div {  
    background-color:red;  
  }  
}
```

Reguła @media handheld, tv mówi przeglądarce internetowej, że czerwony kolor tła dla elementu div ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w urządzeniu przenośnym (handheld) lub w telewizorze (tv), w przeciwnym razie kolor tła elementu div będzie złoty, bo reguła CSS, która określa złoty kolor tła elementu div, nie zostanie nadpisana. Przecinek w naszej regule @media odpowiada za słowo "lub".

Nasza reguła @media może znaleźć się również w atrybucie media elementu style.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Darmowy Kurs CSS</title>  
  
    <style media="handheld, tv">  
      div {  
        background-color:red;  
      }  
    </style>  
  </head>  
  <body>  
    <div>to jest przykładowy element - div</div>  
  </body>  
</html>
```

Tym razem zapis handheld, tv naszej reguły @media znalazł się w atrybucie media elementu style, który został umieszczony w części head dokumentu HTML. Zapis tego typu będzie mówił przeglądarce internetowej, że reguły CSS znajdujące się w danym elemencie style mają zostać dołączone do danego dokumentu HTML tylko wtedy, gdy nasza strona internetowa zostanie wyświetlona w urządzeniu przenośnym (handheld) lub w telewizorze (tv).

@media może znaleźć się również w nieco innym miejscu, w pliku HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Darmowy Kurs CSS</title>

    <link rel="stylesheet" media="handheld, tv" href="arkusz.css" >
  </head>
  <body>
    <div>to jest przykładowy element - div</div>
  </body>
</html>
```

Tym razem zapis handheld, tv naszej reguły @media znalazł się w atrybucie media, w elemencie link, w części head dokumentu HTML. Zapis tego typu będzie mówił przeglądarce internetowej, że plik CSS arkusz.css ma zostać dołączony do danego dokumentu HTML tylko wtedy, gdy nasza strona internetowa zostanie wyświetlona w urządzeniu przenośnym (handheld) lub w telewizorze (tv).

===== Koniec Zadania3=====

## Zadanie 4.

Posługując się treścią pod Zadaniem 3.

Wykonaj zadanie:

1. Wykonaj definicję w CSS zewnętrznym, nazwa pliku CSS → nazwisko\_ucznia\_z3.css (bez polskich liter, małymi literami).
2. gdy strona internetowa zostanie wyświetlona na: urządzeniu o kolorowym ekranie, reguła @media rozpozna wartości szerokości okna przeglądarki internetowej wg reguły: większej niż 390 pikseli oraz mniejsza niż 820 pikseli to kolor liter czerwony o wielkości 2 cm w pozostałych przypadkach kolor liter niebieski o wielkości 1 cm, tekstem do testowania będzie Twoje nazwisko oraz data wykonywania tego zadania.
3. Testowanie odbywa się poprzez zmianę szerokości okna przeglądarki. Podczas zmiany powinny zmieniać się kolor i wielość liter.

Dodatkowe zapisy, jakie pojawiły się wraz ze specyfikacją CSS3 dodatkowe właściwości dla reguł @media.

- width określenie wartości szerokości okna przeglądarki internetowej
- height określenie wartości wysokości okna przeglądarki internetowej
- device-width określenie wartości rozdzielczości ekranu urządzenia (szerokość)

- `device-height` określenie wartości rozdzielczości ekranu urządzenia (wysokość)
- `color` określenie liczby bitów na kolor lub określenie czy urządzenie posiada kolorowy ekran
- `color-index` określenie wartości głębi kolorów, które obsługuje dane urządzenie
- `aspect-ratio` określenie wartości proporcji szerokości do wysokości okna przeglądarki internetowej
- `device-aspect-ratio` określenie wartości proporcji szerokości do wysokości rozdzielczości ekranu urządzenia
- `grid` określenie urządzenia z ograniczonymi możliwościami wyświetlania
- `monochrome` określenie liczby bitów na piksel w urządzeniach monochromatycznych, jednokolorowych
- `orientation` określenie orientacji pionowej lub poziomej urządzenia
- `resolution` określenie wartości gęstości pikseli dla danego urządzenia
- `scan` określenie czy urządzenie posiada skanowanie obrazu progresywne czy międzyliniowe

Niektóre z wyżej wymienionych oznaczeń umożliwiają określenie minimalnej lub maksymalnej wartości, co wymaga dodania przedrostka **min** lub **max**.

Żeby jeszcze bardziej ułatwić nam tworzenie reguł `@media`, zostały dodane kolejne dodatkowe zapisy, którymi możemy się posługiwać.

#### **dodatkowe operatory logiczne dla reguł `@media`.**

- `and` operator "i", służy do tworzenia bardziej precyzyjnych warunków w regule `@media`
- przecinek operator "lub", służy do tworzenia bardziej precyzyjnych warunków w regule `@media`
- `not` operator "negacji", służy do tworzenia bardziej precyzyjnych warunków w regule `@media`
- `only` operator przeznaczony dla starszych przeglądarek internetowych

```
@media all and (min-width:480px) and (max-width:800px) {
  div {
    background-color:red;
  }
}
```

Reguła `@media all and (min-width:480px) and (max-width:800px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość szerokości okna przeglądarki internetowej będzie większa niż 480 pikseli oraz (and) będzie mniejsza niż 800 pikseli, tyczy się to wszystkich urządzeń (all), w jakich zostanie wyświetlona nasza strona internetowa.

Uwaga W nawiasie nie stawiamy średnika. Jeżeli chcemy zdefiniować więcej właściwości dla danego urządzenia to wykorzystujemy do tego operator `and`, np. `(min-width:480px) and (max-width:800px) and (max-height:800px)` itd.

=====Koniec zadania 4=====



# Algorytm wykonania responsywne strony.

## Zadanie 5.

### 1) Wybór rozdzielczości urządzeń mobilnych

Interesuje nas szerokość.

Przyjmujemy następujące szerokości dla urządzeń dla, których wykonujemy stronę:

- 1050px (komputer)
- 950px (tablet poziomo)
- 750px (tablet pionowo)
- 450px (telefon poziomo)
- 300px (telefon pionowo)

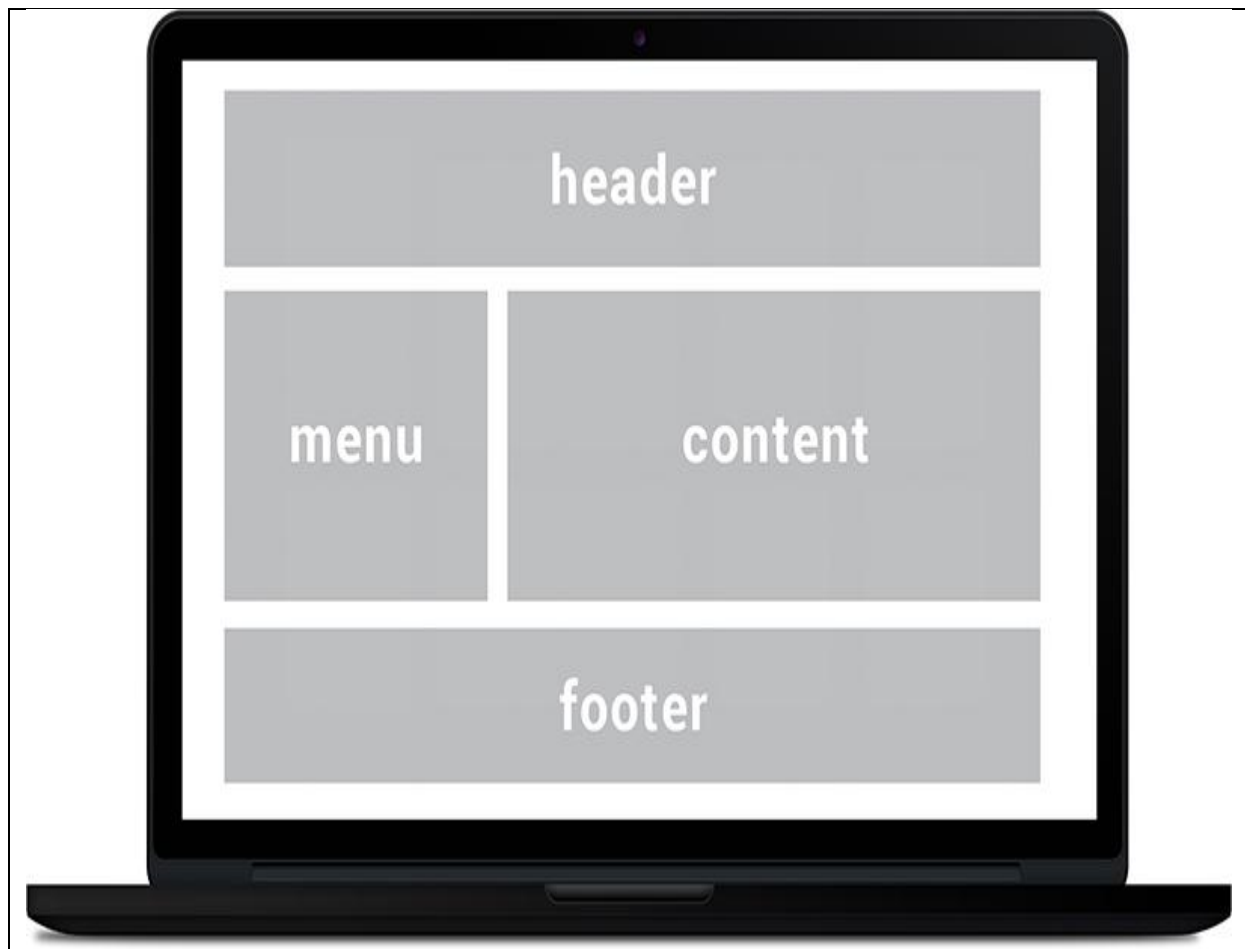
Gdy chcesz wykonać stronę dla konkretnych urządzeń, dokładne parametry może odszukać na stronie:

<a href="http://www.screensiz.es/">http://www.screensiz.es/</a>
-----------------------------------------------------------------

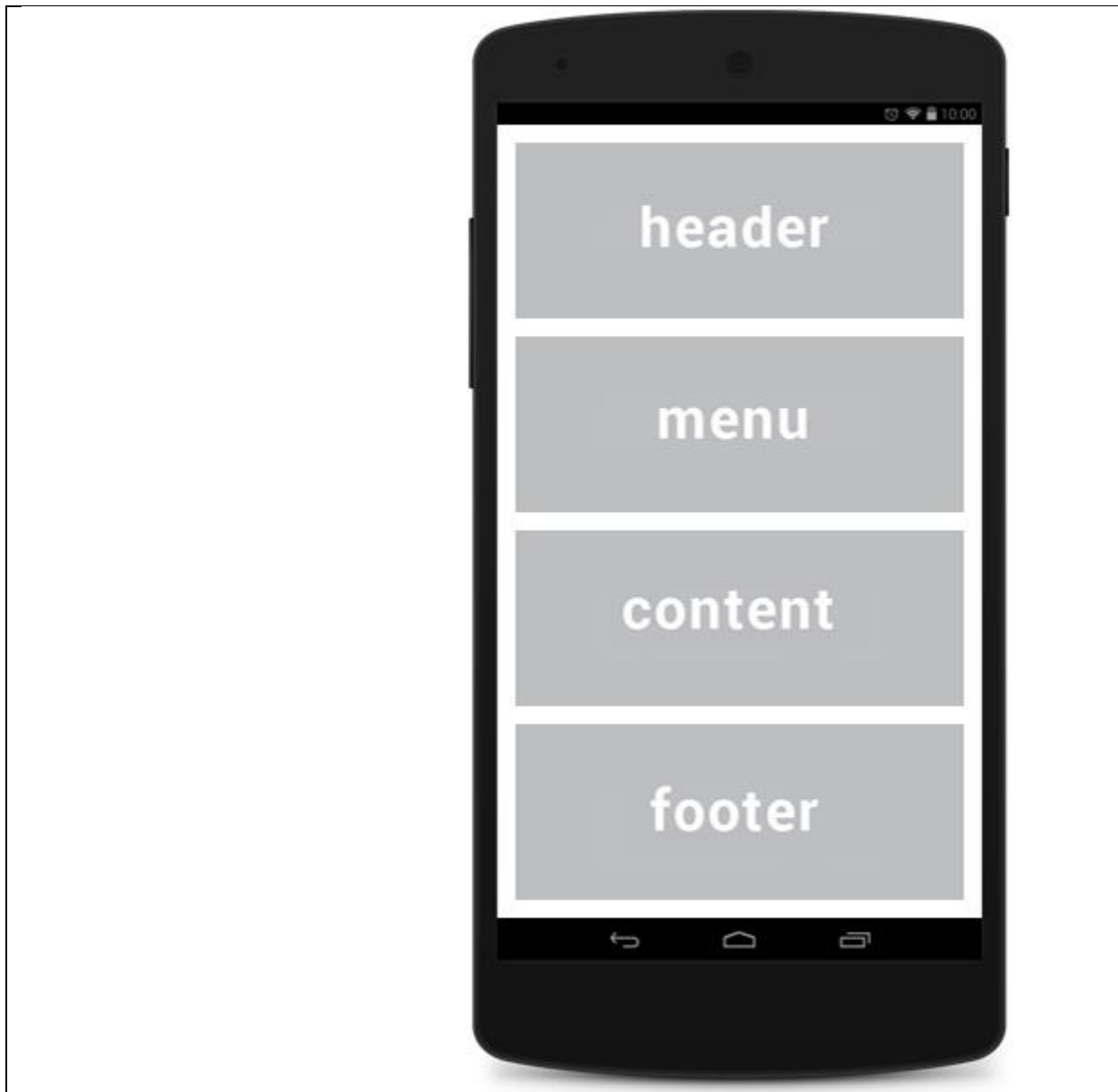
### 2) Zaproponowanie Layout strony

layout → urządzenia o dużej szerokości.

- header → nagłówek
- menu → lista linków
- kontent → zawartość
- footer → stopka



layout → urządzenia o małej szerokości.



### 3) Responsywną strony z zastosowaniem HTML5

#### Założenia/czynności:

1) Załóż folder na Twoim nośniku danych o nazwie:

strona\_responsywna\_nazwisko\_ucznia                      np.  
strona\_responsywna\_oliwiak

2) Załóż dwa podfoldery '**css**' '**images**' w folderze strona\_responsywna\_nazwisko\_ucznia

struktura podfolderów:

- '**css**' → arkusz stylów znajduje się w folderze pod nazwą,
- '**images**' → obrazy znajdują się w folderze

3) Poproś nauczyciela o następujący plik **gdansk3.jpg**. Nagraj do folderu **'images'**.

4)projektujemy stronę dla następujących urządzeń

- 1050px (komputer)
- 950px (tablet poziomo)
- 750px (tablet pionowo)
- 450px (telefon poziomo)
- 300px (telefon pionowo)

5)Ważna uwaga:

**Responsywna strona musi posiadać następując linijkę kodu:**

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

linia kodu stosowana jest aby zdefiniować szerokość okna przeglądarki dla różnych urządzeń,

linia kodu jest tzw. **Media Queries**.

**Media Queries**→ reguły [ stosowane, w celu wykonania stron na różne urządzenia] w pełni opublikowane w CSS3.

6)instrukcja HTML5→krótki opis

<b>instrukcja</b>	<b>wytłumaczenie</b>
<header>	Nagłówek dla naszej strony lub sekcji
<aside>	umieszczamy treść, która nie jest bezpośrednio związana z główną treścią strony, a jest dodatkiem – takie lekkie odbiegnięcie od tematu.
<nav>	W tym znaczniku powinna znajdować się główna nawigacja naszej strony, czyli odnośniki pozwalające poruszać się po witrynie
<section>	Tworzy sekcję dokumentu, czyli grupę z treścią, która różni się od innych treści zawartych na stronie.
<footer>	Zazwyczaj zawiera informacje o elemencie, w którym się znajduje, np. o autorze, odnośniki z menu, dane kontaktowe, dodatkowe informacje czy linki społecznościowe.
<main>	Stworzony po to, by wskazywał główną treść na stronie. Główną to znaczy taką, której poszukuje użytkownik, czyli bez menu strony, logo, stopki i innych niepotrzebnych do przeczytania artykułu elementów. Używając tego znacznika urządzenia takie jak czytniki ekranów będą wiedziały gdzie znajduje się najważniejsza treść strony.
<article>	umieszczamy treść, która ma jakieś znaczenie i mogłaby zostać umieszczona na stronie bez względu na jej pozostałą treść.
<canvas>	umożliwia rysowanie prosto w dokumencie HTML, wykorzystując skrypty JavaScript

7)Dokonaj kopiowania pliku głównego i nadaj mu nazwę

**index\_nazwisko\_ucznia.html**

i nagraj w folderze:

strona\_responsywna\_nazwisko\_ucznia                      np.  
strona\_responsywna\_oliwiak

8)Zmień treści linków oraz nazwę pliku w <link rel> patrz poniżej

nie wszystkie linki na stronie [poniżej] będą działać !!!!!

```
<!DOCTYPE HTML>
<html lang='pl'>
<head>
    <meta charset='utf-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1,
maximum-scale=1'>
    <title>Strona RWD</title>
    <link rel='stylesheet' href='css/style_nowak.css'>
</head>
<body>

<div class='container'>

    <!-- header -->
    <header>
        
    </header>

    <!-- sidebar -->
    <aside>
        <nav>
            <ul>
                <li><a href="index_nowak.html">Start</a></li>
                <li><a href="#">O RWD</a></li>
                <li><a href="#">CSS a RWD</a></li>
                <li><a href="#">Realizacja RWD</a></li>
                <li><a href="kod_css_nowak.html">KOD pliku CSS z
wy tłumaczeniami</a></li>
            </ul>
        </nav>
    </aside>

    <!-- main -->
    <section id="main">
        <h1>RWD (ang. responsive web design) STRONY RENSPO NSY WNE</h1>
        <p><u>Responsive Web Design (RWD)</u> to nic innego, jak
projekt strony internetowej, której układ automatycznie dostosowuje się do
wielkości ekranu urządzenia. Wyświetlane na ekranie treści skalują się
tak, aby były one wygodne w odbiorze dla użytkownika. Strony responsywne
cechuje uniwersalność, czego dowodem jest ich wygląd na smartfonie czy
tablecie. Urządzenia mobilne sprawiły, że przeglądanie stron internetowych
nie jest już domeną komputerów stacjonarnych czy laptopów. Całodobowa
obecność w sieci sprawiła, że konsumenci dokonują zakupów przez urządzenia
mobilne, więc warto nie zostać w tyle i zmienić swoją stronę na RWD.</p>
    </section>

    <!-- footer -->
    <footer>
        <p>RWD_ZSE.pl © 2020</p>
    </footer>

</div>
```

```
</body>
</html>
```

## 9)plik CSS.

a) kopiowanie treści poniżej w ramce, jak nazwa w punkcie b)

```
* {padding:0;margin:0;}

html {width:100%;height:100%;}
body {font-family:Arial, Helvetica, Sans-serif;
font-size:14px;line-height:1.5;
font-weight:400;color:#434343;background:#FFF;}

.container {margin:0 auto;}

header {margin:25px 0;}
header img {width:100%;}

aside {float:left;margin:0 0 25px;}
aside nav ul {list-style:none;}
aside nav ul li {margin-bottom:1px;}
aside nav ul li a {display:block;text-
decoration:none;color:#000;background:#FFC000;padding:8px 20px;}
aside nav ul li a:hover{background:#212121;color:#FFF;}

#main {float:right;margin:0 0 25px;}

footer {clear:both;color:#FFF;background:#121212;padding:15px 0 15px
30px;margin:0 0 25px;}

@media (min-width: 1100px) {
    .container {width:1050px;}
    aside {width:30%;}
    #main {width:65%;}
    textarea {width:100%;}
}
@media (max-width: 1100px) and (min-width: 1000px) {
    .container {width:950px;}
    aside {width:30%;}
    #main {width:65%;}
    textarea {width:100%;}
}
@media (max-width: 1000px) and (min-width: 800px) {
    .container {width:750px;}
    aside {width:30%;}
    #main {width:65%;}
    textarea {width:100%;}
}
@media (max-width: 800px) and (min-width: 500px) {
    .container {width:450px;}
    aside,
    #main {width:100%;}
    textarea {width:100%;}
}
@media (max-width: 500px) {
    .container {width:90%;}
```

```
    aside,  
    #main {width:100%;}  
        textarea {width:100%;}  
  
}
```

b) Nazwa pliku CSS

Jaka nazwa pliku CSS: → `style_twoje_nazwisko.css`

c) Plik CSS podłączany jest w linii (patrz kod poniżej):

jest:

```
<link rel='stylesheet' href='css/style_nowak.css'>
```

ma być:

```
<link rel='stylesheet' href='css/style_twoje_nazwisko.css'>
```

10) Wykonaj link z menu strony: „wyświetlanie kodu strony” → KOD

W tym celu dokonaj kopiowania pliku w ramce poniżej i nadaj mu odpowiednią nazwę.

nazwa pliku po kopiowaniu → `kod_css_nowak.html` [ ty zmień na twoje nazwisko]  
zmień również w linkach strony.

`<textarea spellcheck="false" rows="10">` → zwróć uwagę na parametr **spellcheck**, odszukaj informację o znaczeniu tego parametru dla instrukcji **textarea**.

```
<!DOCTYPE HTML>  
<html lang='pl'>  
<head>  
    <meta charset='utf-8'>  
    <meta name='viewport' content='width=device-width, initial-scale=1, maximum-scale=1'>  
    <title>Strona RWD</title>  
    <link rel='stylesheet' href='css/style_nowak.css'>  
</head>  
<body>  
  
<div class='container'>  
  
    <!-- header -->  
    <header>  
        
```

```

</header>

<!-- sidebar -->
<aside>
    <nav>
        <ul>
            <li><a href="index_nowak.html">Start</a></li>
            <li><a href="#">O RWD</a></li>
            <li><a href="#">CSS a RWD</a></li>
            <li><a href="#">Realizacja RWD</a></li>
            <li><a href="kod_css_nowak.html">KOD pliku CSS z
wy tłumaczeniami</a></li>
        </ul>
    </nav>
</aside>

<!-- main -->
<section id="main">
    <h1>KOD pliku CSS z wytłumaczeniami</h1>
    <textarea spellcheck="false" rows="10">
/* {padding:0;margin:0;}
/* ustawienie wszystkich odległości oraz marginesów na zero */

/*-----*/
html {width:100%;height:100%;}
/* ustawienie znacznika html szerokość i wysokość na 100% */
/*-----*/
body {
font-family:Arial, Helvetica, Sans-serif;
font-size:14px;line-height:1.5;
font-weight:400;
color:#434343;
background:#FFF;}

/* font-family:Arial, Helvetica, Sans-serif; ->tutaj wpisz wytłumaczenie
font-size:14px;line-height:1.5;      ->tutaj wpisz wytłumaczenie
font-weight:400;                    ->tutaj wpisz wytłumaczenie
color:#434343;                      ->tutaj wpisz wytłumaczenie
background:#FFF;}                  ->tutaj wpisz wytłumaczenie */
/*-----*/

</textarea>
</section>

<!-- footer -->
<footer>
    <p>RWD_ZSE.pl © 2020</p>
</footer>

</div>

```



```
</body>  
</html>
```

////////////////////////////////////

11)Wykonaj opcję **O RWD** tak aby wyświetlał się tekst po uruchomieniu tej opcji.

W tym celu:

1. musisz dokonać kopiowania pliku z ramki powyżej.
2. nadać mu nową nazwę: o\_rwd\_nazwisko\_ucznia.html
3. zamiast konstrukcji textarea (wykasuj), wpisz tekst ( jest poniżej), dokonaj formatowania tekstu (podkreślenia, pogrubienia, wyliczenia).

zmień linię kodu <li><a href="#">O RWD</a></li>

na <li><a href="#" o\_rwd\_nowak.html ">O RWD</a></li> [tutaj Twoje nazwisko]

### **TEKST DO WYŚWIETLANIA w opcji O RWD**

#### **Dlaczego warto zmienić stronę na responsywną?**

1. Przeglądarka Google wyżej indeksuje responsywne strony. Oznacza to, że przestarzałe strony muszą liczyć się z mniejszą liczbą odwiedzin.
2. Ponadto, posiadanie responsywnej strony www będzie oznaczało dla użytkownika, że dany sklep/firma/portal jest na czasie. Materiały, które odpowiednio skalują się do rozmiaru ekranu umożliwiają dokładniejsze i przyjemniejsze przeglądanie. Dzięki temu, firma zyska wówczas wielu nowych gości i klientów.
3. Responsywne strony internetowe stanowią dobrą alternatywę dla mobilnych stron internetowych, które pochłaniają więcej czasu i pieniędzy. Mobilna strona internetowa jest witryną dodatkową, która uaktywnia się urządzeniach przenośnych (więc trzeba ponieść dodatkowe koszty związane z jej tworzeniem i utrzymaniem), natomiast w przypadku stron responsywnych, wystarczy jedna wersja.
4. Całodobowa obecność w sieci sprawiła, że konsumenci dokonują zakupów przez urządzenia mobilne, więc warto nie zostać w tyle i zmienić swoją stronę na RWD.

12)Uzupełnij dwie pozostałe opcje:

<li><a href="#">CSS a RWD</a></li>

<li><a href="#">Realizacja RWD</a></li>

Użyj kopiowania plików. Teksty do wyświetlenia musisz ustalić sam, nadać odpowiednie nazwy [z nazwiskiem], zmienić linki są tam teraz tylko # a muszą być pełna nazwa pliku [z Twoim nazwiskiem].

13)Wykonaj testowanie strony RWD ( wykonanej w punktach 1 do 12). Użyj czterech sposobów ( opisane poniżej )

→ pod adresem

<https://www.aurainweb.pl/2018/08/6-darmowych-narzedzi-do-testowania-responsywnosci-strony/>

→ z użyciem przeglądarki Firefox

przyciśnij Shift+Ctrl+M

→ z użyciem przeglądarki Chrome

najpierw klawisz funkcyjny F12 wtedy uruchomią się narzędzia deweloperskie. Tam jedną z opcji jest właśnie „podejrzenie” wyglądu strony – nie tylko po prostu na ekranie smartfona czy tabletu, ale wręcz na ekranach różnych modeli smartfonów i tabletów – i „przeklikanie” jej w wybranym wyglądzie.

→ wyświetl stronę, którą chcesz testować w dowolnej przeglądarce i zmień wymiary okna przeglądarki i po zmianie wymiarów okna widok będzie się samoczynnie zmieniał.

14) w plik CSS (poniżej w ramce) wykonaj komentarze na podstawie informacji umieszczonych poniżej ramki z plikiem CSS .

[Kliknij aby dotrzeć do informacji o komendach CSS, które wpiszesz do komentarzy.](#)

```
/* uzupełnij komentarze */

* {padding:0;margin:0;}
/* ustawienie wszystkich odległości oraz marginesów na zero */

/*-----*/

html {width:100%;height:100%;}
/* ustawienie znacznika html → szerokość i wysokość na 100% */

/*-----*/

body {
font-family:Arial, Helvetica, Sans-serif;
font-size:14px;line-height:1.5;
font-weight:400;
color:#434343;
background:#FFF;}

font-family:Arial, Helvetica, Sans-serif; /* ->tutaj wpisz wytłumaczenie jak komentarz*/
font-size:14px;line-height:1.5; /* ->tutaj wpisz wytłumaczenie jak komentarz */
font-weight:400; /* ->tutaj wpisz wytłumaczenie jak komentarz */
color:#434343; /* ->tutaj wpisz wytłumaczenie jak komentarz */
background:#FFF;} /* ->tutaj wpisz wytłumaczenie jak komentarz */

/*-----*/
```

```

.container {width:1050px;margin:0 auto;}

.container {width:1050px;margin:0 auto;}          /* ->tutaj wpisz wytłumaczenie /*
/*-----*/

header {margin:25px 0;}
header img {width:100%;}

/*wytłumaczenie dwie linie */

/*-----*/
aside {float:left;width:30%;margin:0 0 25px;}
aside nav ul {list-style:none;}
aside nav ul li {margin-bottom:1px;}
aside nav ul li a {display:block;text-decoration:none;color:#000;background:#FFC000;padding:8px
20px;}
aside nav ul li a:hover{background:#212121;color:#FFF;}

/*wytłumaczenie pięć linii */

/*-----*/

#main {float:right;width:65%;margin:0 0 25px;}

/*wytłumaczenie jedna linia */

/*-----*/

footer {clear:both;color:#FFF;background:#121212;padding:15px 0 15px 30px;}

/*wytłumaczenie jedna linia */

/*-----*/

/*przeczytaj teksty w ramkach */

/* Dołącz do arkusza następujące @media -> poniżej, 6 linijek */
/* dokonaj kopiowania jeszcze raz @media uzupełni wytłumaczenia */
/* 7 linijek poniżej masz wykonanie wytłumaczenia, zauważ, że całość jest jako komentarz*/

@media (min-width: 1100px) {
.container {width:1050px;}
aside {width:30%;}
#main {width:65%;}
textarea {width:100%;}
}

/* wytłumaczenie 5 linie
@media (min-width: 1100px) { ->dla urządzeń większych niż 1100px
.container {width:1050px;} ->szerokość głównego kontenera 1050px

```

```
aside {width:30%;}          -> szerokość aside 30% container
#main {width:65%;}          -> szerokość sekcji main 65% container
textarea {width:100%;}      -> szerokość textarea 100% container
}
*/
```

```
/*-----*/
```

```
@media (max-width: 1100px) and (min-width: 1000px) {
.container {width:950px;}
aside {width:30%;}
#main {width:65%;}
textarea {width:100%;}
}
/*wy tłumaczenie 4 linie patrz wyżej */
```

```
/*-----*/
```

```
@media (max-width: 1000px) and (min-width: 800px) {
.container {width:750px;}
aside {width:30%;}
#main {width:65%;}
textarea {width:100%;}
}
/*wy tłumaczenie 4 linie patrz wyżej */
```

```
/*-----*/
```

```
@media (max-width: 800px) and (min-width: 500px) {
.container {width:450px;}
aside,
#main {width:100%;}
textarea {width:100%;}
}
/*wy tłumaczenie 4 linie patrz wyżej *-> przestawienie układu bloki pod sobą */
```

```
/*-----*/
```

```
@media (max-width: 500px) {
.container {width:90%;}
aside,
#main {width:100%;}
textarea {width:100%;}
}
```

```
/*-----*/
```

## Zapewnienie responsywność strony

Aby nasza strona stała się stroną responsywną musimy dodać do naszego arkusza CSS warunki **Media Queries**.

### Przykłady

Przykładową linię kodu CSS zawierającą warunek Media Queries:

```
@media (max-width: 800px) and (min-width: 600px) {  
/* style css */  
}
```

W kodzie powyżej został podany warunek który definiuje, że wpisane w nawias klamrowy style będą 'aktywne' tylko dla szerokości okna przeglądarki mniejszej niż 800px gdyż ustaliliśmy wartość 'max-width' na 800px i dla szerokości okna przeglądarki większej niż 600px gdyż ustaliliśmy wartość 'min-width' na 600px.

Tak więc wpisane style będą 'aktywne' gdy szerokość okna przeglądarki będzie zawierała się w przedziale od 800px do 600px.

Możemy również stosować taki zapis:

```
@media (min-width: 600px) {  
/* style css */  
}
```

W takim przypadku nie mamy górnej granicy czyli 'max-width' i warunek będzie spełniany za każdym razem kiedy szerokość okna przeglądarki będzie większa niż 600px.

Analogiczna sytuacja jest z zapisem:

```
@media (max-width: 800px) {  
/* style css */  
}
```

W takim przypadku nie mamy dolnej granicy czyli 'min-width' tak więc warunek będzie spełniany za każdym razem kiedy szerokość okna przeglądarki będzie mniejsza niż 800px.

**Wykonujemy szkielet warunków Media Queries bez podawania wartości oraz usuwając skrajne warunki:** (nie jest to konieczne lecz jest dobra praktyka programowania )

```
@media (min-width: px) {  
/* style dla 1050px */  
}  
@media (max-width: px) and (min-width: px) {  
/* style dla 950px */  
}  
@media (max-width: px) and (min-width: px) {  
/* style dla 750px */  
}  
@media (max-width: px) and (min-width: px) {  
/* style dla 450px */  
}
```

```
@media (max-width: px) {  
/* style dla 300px */  
}
```

Wiemy, że nasza strona w tym warunku ma mieć szerokość 1050px, musi więc wyświetlać się w oknie przeglądarki większym niż 1050px tak, aby nie pojawił się na dole poziomy pasek przewijania. Dodajmy jeszcze do naszej **wartości 50px 'luzu'**. na pionowy pasek przeglądania otrzymując w ten sposób wartość 1100px, którą to zastosujemy w pierwszym warunku.

#### **1050px (laptop)**

```
@media (min-width: 1100px) {  
/* style dla 1050px */  
}
```

#### **950px (tablet orientacja pozioma)**

Czas na kolejny warunek. We wcześniejszym zdefiniowaliśmy wartość 1100px jako wartość 'min'. W tym warunku musimy przyjąć tę wartość jako wartość 'max' (aby zachować ciągłość przedziałów). Wartość 'min' określamy w taki sam sposób jak robiliśmy to w poprzednim warunku tak więc otrzymujemy 1000px (950px szerokość strony plus 50px 'luzu')

```
@media (max-width: 1100px) and (min-width: 1000px) {  
/* style dla 950px */  
}
```

#### **750px (tablet orientacja pionowa)**

Kolejny warunek. Sytuacja analogiczna do poprzedniego.

```
@media (max-width: 1000px) and (min-width: 800px) {  
/* style dla 750px */  
}
```

#### **450px (telefon orientacja pozioma)**

```
@media (max-width: 800px) and (min-width: 500px) {  
/* style dla 450px */  
}
```

#### **300px (telefon orientacja pionowa)**

W tym przypadku nie mamy wartości 'min' tak więc przepisujemy tylko do wartości 'max' wartość 'min' z poprzedniego warunku.

```
@media (max-width: 500px) {  
/* style dla 300px */  
}
```

Możemy teraz przejść do wpisywania w przygotowane warunki style CSS dla naszej strony:

```
@media (min-width: 1100px) {  
.container {width:1050px;margin:0 auto;}
```

```
}

@media (max-width: 1100px) and (min-width: 1000px) {
.container {width:950px;margin:0 auto;}
}

@media (max-width: 1000px) and (min-width: 800px) {
.container {width:750px;margin:0 auto;}
}

@media (max-width: 800px) and (min-width: 500px) {
.container {width:450px;margin:0 auto;}
}

@media (max-width: 500px) {
.container {width:300px;margin:0 auto;}
}
```

Jak widzimy wartość atrybutu margin jest dla każdego warunku taka sama więc możemy wyciągnąć ją poza warunki aby nie powtarzać jej w każdym z nich:

```
.container {margin:0 auto;}
```

```
@media (min-width: 1100px) {
.container {width:1050px;}
}
@media (max-width: 1100px) and (min-width: 1000px) {
.container {width:950px;}
}
@media (max-width: 1000px) and (min-width: 800px) {
.container {width:750px;}
}
@media (max-width: 800px) and (min-width: 500px) {
.container {width:450px;}
}
@media (max-width: 500px) {
.container {width:300px;}
}
```

Z uwagi na to że na rynku istnieje wiele urządzeń mobilnych również takich o wyświetlaczach z rozdzielczością mniejszą niż 300px dobrym rozwiązaniem będzie zamiana dla najmniejszego warunku wartości szerokości strony z podanej w pikselach na procentową. Zastosujmy więc szerokość 90% całej dostępnej szerokości urządzenia.

```
@media (max-width: 500px) {
.container {width:90%;}
}
```

W tym momencie nasza strona jest już responsywna, lecz przy małych rozdzielczościach staje się nieczytelna. Zrobimy jeszcze porządek w taki sposób, aby z układu dwukolumnowego powstał układ jedno kolumnowy, pamiętając oczywiście o pogrupowaniu odpowiednio powtarzających się



parametrów:

```
@media (min-width: 1100px) {  
.container {width:1050px;}  
aside {width:30%;}  
#main {width:65%;}  
}  
@media (max-width: 1100px) and (min-width: 1000px) {  
.container {width:950px;}  
aside {width:30%;}  
#main {width:65%;}  
}  
@media (max-width: 1000px) and (min-width: 800px) {  
.container {width:750px;}  
aside {width:30%;}  
#main {width:65%;}  
}  
@media (max-width: 800px) and (min-width: 500px) {  
.container {width:450px;}  
aside,  
#main {width:100%;}  
}  
@media (max-width: 500px) {  
.container {width:90%;}  
aside,  
#main {width:100%;}  
}
```

# Wykonanie projektu

1) projektujemy stronę dla następujących urządzeń

- 1300+numer z dziennika px (telewizor)
- 1040+numer z dziennika px (komputer)
- 940+numer z dziennika px (tablet poziomo)
- 740+numer z dziennika px (tablet pionowo)
- 440+numer z dziennika px (telefon poziomo)
- 280+numer z dziennika px (telefon pionowo)

2) Przykładowe ułożenia strony

## Wygląd dla dużych szerokości



[Start](#)

[O RWD](#)

[CSS a RWD](#)

[Realizacja RWD](#)

[KOD pliku CSS z tłumaczeniami](#)

### RWD (ang. responsive web design) STRONY RESPONSYWNE

Responsive Web Design (RWD) to nic innego, jak projekt strony internetowej, której układ automatycznie dostosowuje się do wielkości ekranu urządzenia. Wyświetlane na ekranie treści skalują się tak, aby były one wygodne w odbiorze dla użytkownika. Strony responsywne cechuje uniwersalność, czego dowodem jest ich wygląd na smartfonie czy tablecie. Urządzenia mobilne sprawiły, że przeglądanie stron internetowych nie jest już domeną komputerów stacjonarnych czy laptopów. Całodobowa obecność w sieci sprawiła, że konsumenci dokonują zakupów przez urządzenia mobilne, więc warto nie zostać w tyle i zmienić swoją stronę na RWD.

RWD\_ZSE.pl © 2020

## **Wygląd dla małych szerokości**



Start

O RWD

CSS a RWD

Realizacja RWD

KOD pliku CSS z wytłumaczeniami

## RWD (ang. responsive web design)

### STRONY RESPONSYWNE

Responsive Web Design (RWD) to nic innego, jak projekt strony internetowej, której układ automatycznie dostosowuje się do wielkości ekranu urządzenia. Wyświetlane na ekranie treści skalują się tak, aby były one wygodne w odbiorze dla użytkownika. Strony responsywne cechuje uniwersalność, czego dowodem jest ich

przeglądanie stron internetowych nie jest już domeną komputerów stacjonarnych czy laptopów. Całodobowa obecność w sieci sprawiła, że konsumenci dokonują zakupów przez urządzenia mobilne, więc warto nie zostać w tyle i zmienić swoją stronę na RWD.

RWD\_ZSE.pl © 2020

3)numery układów dla dużych szerokości

układ 1

--

LOGO	
MENU	TREŚĆ
STOPKA	

układ 2

LOGO	
TREŚĆ	MENU
STOPKA	

układ 3

STOPKA	
MENU	TREŚĆ
LOGO	

układ 4

STOPKA	
TREŚĆ	MENU
LOGO	

4)numery układów dla **małych szerokości**

układ 1

MENU
LOGO
TREŚĆ
STOPKA

układ 2

LOGO
------

TREŚĆ
MENU
STOPKA

układ 3

TREŚĆ
MENU
LOGO
STOPKA

układ 4

LOGO
------

MENU
TREŚĆ
STOPKA

5)Tematy prac (wymagania dla stron pod tabelą z tematami).

nr w dzien	temat	nr dużego układu	nr małego układu
1.	Bieszczady, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	2
2.	Beskid Śląski, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	1
3.	Podhale, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	4	1
4.	Kaszuby, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	4	4
5.	Kociewie, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	1	3
6.	Mazury, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	1	4
7.	Warmia, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	2	1
8.	Podlasie, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	1	3
9.	Beskid Mały, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	2	2
10.	Beskid Śląsko-Morawski , geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	2
11.	Pieniny, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	1
12.	Kotlina Żywiecka., geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	2	3
13.	Wielkopolska, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	3
14.	Mazowsze, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	4	2
15.	Górny Śląsk, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	4



16.	Dolny Śląsk, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	1
17.	Sudety, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	4
18.	Góry stołowe, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	4	3
19.	Opolszczyzna, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	4	2
20.	Małopolska, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	2	2
21.	Suwalszczyzna, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	1
22.	Ziemia Lubuska, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	4	3
23.	Kurpie, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	2	3
24.	Kujawy, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	2	4
25.	Roztocze, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	1	3
26.	Lubelszczyzna, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	3
27.	Region Świętokrzyski, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	4	3
28.	Region Łowicki, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	1
29.	Puszcza Białowieska, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	2	1
30.	Ziemia Kłódzka, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	1
31.	Pojezierze Drawski, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	1	3
32.	Jura Krakowsko-Częstochowska, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	3
33.	Pobrzeże Szczecińskie, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	2	3
34.	Pobrzeże Koszalińskie, geografia, klimat, historia, kultura, wycieczki, ciekawe miejsca, kuchnia regionalna.	3	1

#### 6)Wymagania dla strony.

- Strona powinna być zainstalowana na serwerze WWW.
- Menu strony powinno mieć co najmniej 7 opcji.
- Działanie wszystkich opcji menu.
- Na ocenę będą wpływały treści merytoryczne poszczególnych opcji.
- Na ocenę będzie wpływać estetyka wykonania strony (dobór kolorów, kroju czcionek itp.)
- Zgodność z założeniami ujętymi w tabeli (duży i mały układ), działanie podczas zmniejszania i powiększania okna przeglądarki.



```

* {padding:0;margin:0;}

html {width:100%;height:100%;}
body {font-family:Arial, Helvetica, Sans-serif;
font-size:14px;line-height:1.5;
font-weight:400;color:#434343;background:#FFF;}

.container {margin:0 auto;}

header {margin:25px 0;}
header img {width:100%;}

aside {float:left;margin:0 0 25px;}
aside nav ul {list-style:none;}
aside nav ul li {margin-bottom:1px;}
aside nav ul li a {display:block;text-decoration:none;color:#000;background:#FFC000;padding:8px
20px;}
aside nav ul li a:hover{background:#212121;color:#FFF;}

#main {float:right;margin:0 0 25px;}

footer {clear:both;color:#FFF;background:#121212;padding:15px 0 15px 30px;margin:0 0 25px;}

@media (min-width: 1100px) {
    .container {width:1050px;}
    aside {width:30%;}
    #main {width:65%;}
    textarea {width:100%;}
}
@media (max-width: 1100px) and (min-width: 1000px) {
    .container {width:950px;}
    aside {width:30%;}
    #main {width:65%;}
    textarea {width:100%;}
}
@media (max-width: 1000px) and (min-width: 800px) {
    .container {width:750px;}
    aside {width:30%;}
    #main {width:65%;}
    textarea {width:100%;}
}
@media (max-width: 800px) and (min-width: 500px) {
    .container {width:450px;}
    aside,
    #main {width:100%;}
    textarea {width:100%;}
}
@media (max-width: 500px) {
    .container {width:90%;}
    aside,
    #main {width:100%;}
}

```

```
textarea {width:100%;}  
}
```

[Zobacz demo](#)

[Pobierz pliki źródłowe](#)

Zapraszamy również do zapoznania się z drugą częścią poradnika - prezentującą sposób [tworzenia stron responsywnych](#) opartych o wartości procentowe.

<http://www.footstep.pl/blog/strony-responsywne-tutorial>

Więcej przykładów znajdziesz poniżej.

## width

**width** - za pomocą tej właściwości reguły @media, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy szerokość okna przeglądarki internetowej będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość. Wartość tą możemy określić za pomocą dowolnej jednostki, która jest prawidłową jednostką dla właściwości width.

```
@media all and (width:800px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła @media all and (width:800px) mówi przeglądarce internetowej, że czerwony kolor tła dla elementu div ma zostać aktywowany tylko w momencie gdy wartość szerokości okna przeglądarki internetowej będzie dokładnie równa 800 pikseli.

```
@media all and (min-width:800px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła @media all and (min-width:800px) mówi przeglądarce internetowej, że czerwony kolor tła dla elementu div ma zostać aktywowany tylko w momencie gdy wartość szerokości okna przeglądarki internetowej będzie nie mniejsza niż 800 pikseli.

```
@media all and (max-width:800px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (max-width:800px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość szerokości okna przeglądarki internetowej będzie nie większa niż 800 pikseli.

## height

`height` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy wysokość okna przeglądarki internetowej będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość. Wartość tą możemy określić za pomocą dowolnej jednostki, która jest prawidłową jednostką dla właściwości `height`.

```
@media all and (height:300px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (height:300px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość wysokości okna przeglądarki internetowej będzie dokładnie równa 300 pikseli.

```
@media all and (min-height:300px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (min-height:300px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość wysokości okna przeglądarki internetowej będzie nie mniejsza niż 300 pikseli.

```
@media all and (max-height:300px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (max-height:300px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość wysokości okna przeglądarki internetowej będzie nie większa niż 300 pikseli.

## device-width

`device-width` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy szerokość rozdzielczości ekranu urządzenia będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość. Wartość tą możemy określić za pomocą dowolnej jednostki, która jest prawidłową jednostką dla właściwości `width`.

```
@media all and (device-width:800px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (device-width:800px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość szerokości rozdzielczości ekranu urządzenia będzie dokładnie równa 800 pikseli.

```
@media all and (min-device-width:800px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (min-device-width:800px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość szerokości rozdzielczości ekranu urządzenia będzie nie mniejsza niż 800 pikseli.

```
@media all and (max-device-width:800px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (max-device-width:800px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość szerokości rozdzielczości ekranu urządzenia będzie nie większa niż 800 pikseli.

## device-height

`device-height` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy wysokość rozdzielczości ekranu urządzenia będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość. Wartość tą możemy określić za pomocą dowolnej jednostki, która jest prawidłową jednostką dla właściwości `height`.

```
@media all and (device-height:480px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (device-height:480px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość wysokości rozdzielczości ekranu urządzenia będzie dokładnie równa 480 pikseli.

```
@media all and (min-device-height:480px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (min-device-height:480px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość wysokości rozdzielczości ekranu urządzenia będzie nie mniejsza niż 480 pikseli.

```
@media all and (max-device-height:480px) {  
  div {
```

```
background-color:red;
}
}
```

Reguła `@media all and (max-device-height:480px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość wysokości rozdzielczości ekranu urządzenia będzie nie większa niż 480 pikseli.

## color

`color` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy ilość bitów na kolor w danym urządzeniu będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość. Jeżeli ilość bitów na dany kolor (czerwony, zielony, niebieski) nie jest równa, to brana jest pod uwagę najmniejsza wartość.

```
@media all and (color:8) {
  div {
    background-color:red;
  }
}
```

Reguła `@media all and (color:8)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy ilość bitów na kolor w danym urządzeniu jest dokładnie równa 8.

```
@media all and (min-color:8) {
  div {
    background-color:red;
  }
}
```

Reguła `@media all and (min-color:8)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy ilość bitów na kolor w danym urządzeniu nie jest mniejsza niż 8.

```
@media all and (max-color:8) {
  div {
    background-color:red;
  }
}
```

Reguła `@media all and (max-color:8)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy ilość bitów na kolor w danym urządzeniu nie jest większa niż 8.

## color-index

`color-index` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy głębokość kolorów w danym urządzeniu będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość.

```
@media all and (color-index:256) {
  div {
    background-color:red;
  }
}
```

```
}  
}
```

Reguła `@media all and (color-index:256)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość głębi kolorów w danym urządzeniu jest dokładnie równa 256.

```
@media all and (min-color-index:256) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (min-color-index:256)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość głębi kolorów w danym urządzeniu nie jest mniejsza niż 256.

```
@media all and (max-color-index:256) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (max-color-index:256)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość głębi kolorów w danym urządzeniu nie jest większa niż 256.

## aspect-ratio

`aspect-ratio` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy proporcja szerokości do wysokości okna przeglądarki internetowej będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość.

Wartość tą podajemy jako ułamek. Kreskę ułamkową stanowi znak `/`.

Przykładowa wartość `4/3` pasuje między innymi do takich rozmiarów jak: `640×480`, `800×600`, `1024×768`, ponieważ, np.  $640 / 480 = 1.(3)$ , czyli `4/3`.

```
@media all and (aspect-ratio:4/3) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (aspect-ratio:4/3)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość proporcji szerokości do wysokości okna przeglądarki internetowej będzie dokładnie równa `4/3`.

```
@media all and (min-aspect-ratio:4/3) {  
  div {  
    background-color:red;  
  }  
}
```



```
}
```

Reguła `@media all and (min-aspect-ratio:4/3)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość proporcji szerokości do wysokości okna przeglądarki internetowej będzie nie mniejsza niż  $4/3$ .

```
@media all and (max-aspect-ratio:4/3) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (max-aspect-ratio:4/3)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość proporcji szerokości do wysokości okna przeglądarki internetowej będzie nie większa niż  $4/3$ .

## device-aspect-ratio

`device-aspect-ratio` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy proporcja szerokości do wysokości rozdzielczości ekranu urządzenia będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość.

Wartość tą podajemy jako ułamek. Kreskę ułamkową stanowi znak `/`.

Przykładowa wartość  $4/3$  pasuje między innymi do takich rozmiarów jak:  $640 \times 480$ ,  $800 \times 600$ ,  $1024 \times 768$ , ponieważ, np.  $800 / 600 = 1.(3)$ , czyli  $4/3$ .

```
@media all and (device-aspect-ratio:4/3) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (device-aspect-ratio:4/3)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość proporcji szerokości do wysokości rozdzielczości ekranu urządzenia będzie dokładnie równa  $4/3$ .

```
@media all and (min-device-aspect-ratio:4/3) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (min-device-aspect-ratio:4/3)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość proporcji szerokości do wysokości rozdzielczości ekranu urządzenia będzie nie mniejsza niż  $4/3$ .

```
@media all and (max-device-aspect-ratio:4/3) {  
  div {  
    background-color:red;  
  }  
}
```

```
}
```

Reguła `@media all and (max-device-aspect-ratio:4/3)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość proporcji szerokości do wysokości rozdzielczości ekranu urządzenia będzie nie większa niż 4/3.

## grid

`grid` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy nasza strona internetowa została wyświetlona w urządzeniu z ograniczonymi możliwościami wyświetlania (np. terminale, starsze telefony komórkowe ze stałą wartością wysokości czcionki, urządzenia korzystające z bitmapy).

```
@media handheld and (grid) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media handheld and (grid)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa została wyświetlona w przenośnym urządzeniu z ograniczonymi możliwościami wyświetlania.

## monochrome

`monochrome` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy ilość bitów na kolor w urządzeniu, które wyświetla naszą stronę internetową w trybie monochromatycznym, będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość. Tryb monochromatyczny to nie tylko kolory czarny i biały, ponieważ w różnych trybach monochromatycznych mogą się zdarzyć również inne sekwencje prostych kolorów.

```
@media all and (monochrome:8) {  
  div {  
    background-color:white;  
  }  
}
```

Reguła `@media all and (monochrome:8)` mówi przeglądarce internetowej, że biały kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy ilość bitów na kolor w danym urządzeniu monochromatycznym jest dokładnie równa 8.

```
@media all and (min-monochrome:8) {  
  div {  
    background-color:white;  
  }  
}
```

Reguła `@media all and (min-monochrome:8)` mówi przeglądarce internetowej, że biały kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy ilość bitów na kolor w danym urządzeniu monochromatycznym nie jest mniejsza niż 8.

```
@media all and (max-monochrome:8) {  
  div {
```

```
background-color:white;
}
}
```

Reguła `@media all and (max-monochrome:8)` mówi przeglądarce internetowej, że biały kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy ilość bitów na kolor w danym urządzeniu monochromatycznym nie jest większa niż 8.

## orientation

`orientation` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy szerokość okna przeglądarki internetowej będzie miała wartość mniejszą lub większą niż wysokość okna przeglądarki internetowej.

Cecha `orientation` pomaga nam wykryć w jakiej orientacji, pionowej (`portrait`) czy poziomej (`landscape`), została wyświetlona nasza strona internetowa, w przeglądarce danego urządzenia.

```
@media all and (orientation:portrait) {
  div {
    background-color:red;
  }
}
```

Reguła `@media all and (orientation:portrait)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w przeglądarce internetowej, której wartość szerokości okna jest mniejsza od wartości wysokości jej okna (orientacja pionowa).

```
@media all and (orientation:landscape) {
  div {
    background-color:red;
  }
}
```

Reguła `@media all and (orientation:landscape)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w przeglądarce internetowej, której wartość szerokości okna jest większa od wartości wysokości jej okna (orientacja pozioma).

## resolution

`resolution` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy gęstość pikseli w danym urządzeniu będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość. Aby określić wartość dla `resolution` możemy posłużyć się jednostką `dpi` (plamki na cal) lub `dpcm` (plamki na centymetr).

```
@media all and (resolution:200dpi) {
  div {
    background-color:red;
  }
}
```

Reguła `@media all and (resolution:200dpi)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość gęstości pikseli w danym urządzeniu jest dokładnie równa 200 plamek na cal.

```
@media all and (min-resolution:200dpi) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (min-resolution:200dpi)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość gęstości pikseli w danym urządzeniu nie jest mniejsza niż 200 plamek na cal.

```
@media all and (max-resolution:200dpi) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (max-resolution:200dpi)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy wartość gęstości pikseli w danym urządzeniu nie jest większa niż 200 plamek na cal.

## scan

`scan` - za pomocą tej właściwości reguły `@media`, możemy określić reguły CSS, które zostaną aktywowane tylko wtedy, gdy tryb skanowania obrazu w telewizorze jest trybem progresywnym (progressive) lub międzyliniowym (interlace).

```
@media tv and (scan:progressive) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (scan:progressive)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w telewizorze, którego obraz jest skanowany w trybie progresywnym.

```
@media tv and (scan:interlace) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (scan:interlace)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w telewizorze, którego obraz jest skanowany w trybie międzyliniowym.

## and

and - operator ten łączy różne właściwości w regule @media, dla danego urządzenia, dzięki czemu możemy tworzyć bardziej precyzyjne reguły @media. Słowo and oznacza "i" lub jak kto woli "oraz".

```
@media all and (min-width:480px) and (max-width:800px) and (orientation:landscape) {  
  div {  
    background-color:red;  
  }  
}
```

Powyższa reguła @media mówi przeglądarce internetowej, że czerwony kolor tła dla elementu div ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w przeglądarce internetowej, w dowolnym urządzeniu (all), którego aktualna wartość szerokości okna przeglądarki internetowej jest nie mniejsza niż 480 pikseli (min-width:480px) ORAZ nie jest większa niż 800 pikseli (max-width:800px), dodatkowo strona musi zostać wyświetlona w orientacji poziomej (orientation:landscape).

## przecinek

przecinek - operator ten umożliwia stworzenie jednego zapisu reguły @media, z różnymi właściwościami, dla różnych typów urządzeń. Przecinek w regule @media pełni rolę słowa "lub".

Operator ten przydaje się nam w przypadku gdy chcemy określić te same reguły CSS dla kilku różnych urządzeń, możemy (lecz nie musimy) przy tym określić różne właściwości dla reguły @media.

```
@media print, tv, handheld {  
  div {  
    background-color:red;  
  }  
}
```

Reguła @media print, tv, handheld mówi przeglądarce internetowej, że czerwony kolor tła dla elementu div ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w podglądzie wydruku (print) LUB w telewizorze (tv) LUB w bezprzewodowym urządzeniu ręcznym (handheld).

```
@media tv and (scan:progressive), handheld and (monochrome) and (orientation:portrait) {  
  div {  
    background-color:red;  
  }  
}
```

Powyższa reguła @media mówi przeglądarce internetowej, że czerwony kolor tła dla elementu div ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w telewizorze (tv) z progresywnym skanowaniem obrazu (scan:progressive) LUB gdy nasza strona zostanie wyświetlona w bezprzewodowym urządzeniu ręcznym (handheld), które jest monochromatyczne (monochrome) ORAZ które posiada orientację pionową (orientation:portrait).

## not

not - operator ten umożliwia zanegowanie zapisu reguły @media, czyli zaprzeczenia danego zapisu.

```
@media not all and (color) {  
  div {  
    background-color:red;  
  }  
}
```

```
}  
}
```

Reguła `@media not all and (color)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w przeglądarce internetowej, w dowolnym urządzeniu tylko nie w urządzeniu z kolorowym ekranem.

Uwaga Operator `not` dotyczy się tylko tej grupy urządzeń i wartości, w którym został on użyty. Gdy dodamy do naszej reguły `@media` więcej grup urządzeń i właściwości, po przecinku, to w każdej takiej grupie musimy użyć operator `not`, oczywiście jeżeli chcemy zaprzeczyć daną grupę operatorem `not`.

```
@media not all and (color), screen {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media not all and (color), screen` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w przeglądarce internetowej, w dowolnym urządzeniu tylko nie w urządzeniu z kolorowym ekranem. Jednak czerwony kolor tła dla elementu `div` może zostać aktywowany w komputerze z kolorowym ekranem, ponieważ po przecinku użyliśmy zapisu `screen` bez operatora `not`.

### only

`only` - operator przeznaczony dla starszych przeglądarek internetowych. Operator ten umożliwia zablokowanie reguł `@media` dla starszych przeglądarek internetowych (np. starszych wersji przeglądarki Internet Explorer).

```
@media only all {  
  div {  
    background-color:red;  
  }  
}
```

Gdyby nasza reguła nie zawierała operatora `only` to przeglądarka internetowa Internet Explorer w wersji wcześniejszej niż 9, która obsługuje proste reguły `@media` (tylko te, które określają typ urządzenia) dodałaby czerwony kolor tła do elementu `div`, a tak dzięki operatorowi `only` tak się nie dzieje.

### Dodatkowe informacje

1) Gdy chcemy określić jakieś kryteria w regule `@media`, według których reguły CSS mają być dodane do naszej strony tylko w określonej sytuacji, jednak dla wszystkich typów urządzeń, to możemy pominąć zapis `all and`.

```
@media (min-width:480px) and (max-width:800px) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media (min-width:480px) and (max-width:800px)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w przeglądarce internetowej, w dowolnym urządzeniu, którego aktualna wartość szerokości okna przeglądarki internetowej jest nie mniejsza niż 480 pikseli (`min-width:480px`) oraz nie jest większa niż 800 pikseli (`max-width:800px`).

2) Każdą właściwość `@media`, jaką poznaliśmy, możemy użyć bez podawania konkretnej wartości, dzięki czemu nasz zapis zyska dodatkowe cechy.

```
@media all and (color) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (color)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w przeglądarce internetowej, w dowolnym urządzeniu, które posiada kolorowy wyświetlacz.

```
@media all and (monochrome) {  
  div {  
    background-color:red;  
  }  
}
```

Reguła `@media all and (monochrome)` mówi przeglądarce internetowej, że czerwony kolor tła dla elementu `div` ma zostać aktywowany tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w przeglądarce internetowej, w dowolnym urządzeniu monochromatycznym.

3) Wszystkie poznane zapisy w tej części kursu CSS możemy również wykorzystywać w regule `@import`, która odpowiada za dołączenie dodatkowego arkusza CSS do danego pliku CSS.

```
@import url("dodatkowy_arkusz.css") screen and (min-width:480px), tv;
```

Powyższa reguła `@import` mówi przeglądarce internetowej, że plik CSS o nazwie "dodatkowy\_arkusz" ma zostać dołączony do danego pliku CSS tylko w momencie gdy nasza strona internetowa zostanie wyświetlona w telewizorze (tv) lub w oknie przeglądarki internetowej komputera z kolorowym ekranem (screen), dodatkowo szerokość tej przeglądarki internetowej musi być nie mniejsza niż 480 pikseli (`min-width:480px`).