

Zad.1.

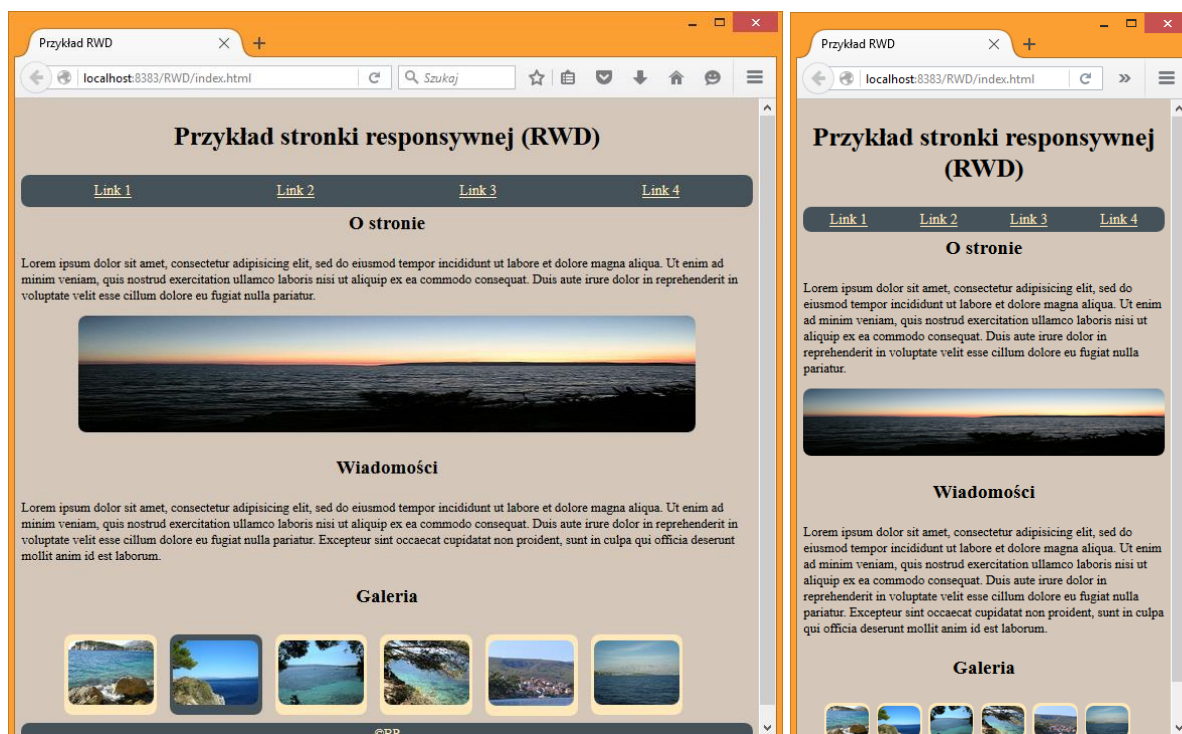
Utwórz dokument HTML5 z zastosowaniem nowych elementów struktury strony dodanych w standardzie HTML5 (*header, footer, section, nav itp.*) postaci jak na Listingu 1.

Listing 1. Przykładowa struktura strony z wykorzystaniem nowych elementów HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <title>Przykład RWD</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <header>
      <h1>Przykład stronki responsywnej (RWD)</h1>
      <nav>
        <a href="index.html">Link 1</a>
        <a href="strona2.html">Link 2</a>
        <a href="strona3.html">Link 3</a>
        <a href="strona4.html">Link 4</a>
      </nav>
    </header>

    <section>
      <h2>O stronie</h2>
      <p>
        Lorem ipsum ...
      </p>
      <p class="srodek">
        
      </p>
    </section>
    <section>
      <h2>Wiadomości</h2>
      <p>
        Lorem ipsum ...
      </p>
    </section>
    <section>
      <div class="galeria">
        <h2>Galeria</h2>
        <div class="slajd">
          
        </div>
        <div class="slajd">
          
        </div>
        ...
      </div>
    </section>
    <footer>&copy;BP</footer>
  </body>
</html>
```

Naszym zadaniem będzie dostosowanie strony dla urządzeń o różnych rozdzielczościach i szerokościach. Pierwszy widok, do którego dążymy, przedstawiony został na rys. 1.



Rys. 1. Widok strony bez zapytania o media przy różnych szerokościach okna przeglądarki

W widoku mobilnym bardzo istotna jest rola znacznika:

`<meta name="viewport">`

`content="width=device-width, initial-scale=1.0, user-scalable=true">`

który określa, że dany dokument został zoptymalizowany z myślą o urządzeniach mobilnych.

W atrybucie **content** ustawiono dyrektywy:

- **Width=device-width** - określa, że szerokość obszaru wyświetlania powinna odpowiadać szerokości ekranu danego urządzenia;
- **initial-scale** - ustawia początkową wartość skalowania (przybliżenia) dla danej strony (domyślna wartość początkowa zależy od przeglądarki stosowanej na urządzeniu; wartość 1.0 powoduje wyświetlenie dokumentu bez skalowania);
- **user-scalable** - ustala, czy dany użytkownik może przybliżać i oddalać obszar wyświetlania.

Kolejny etap to dostosowanie arkusza stylów odpowiedzialnego za formatowanie strony. Pierwszą rzeczą, którą możemy zrobić, jest sformatowanie obrazka jako **fluid image**, czyli nakazanie przeglądarce płynnego dostosowywania rozmiaru obrazka do podanej rozdzielczości, co realizuje prosta reguła CSS:

```
img { max-width: 100%; height: auto; }
```

Przykładowy arkusz CSS może być postaci jak na Listingu 2.

Listing 2. Pierwszy arkusz CSS

```
body { background:#d4c7b9; }
h1,h2,footer { padding:0.2em; text-align:center; }
.srodek { text-align:center;}
nav{ float:left; width:100%; }
```

```
nav, footer {background: #45525a; color:#ffe9ba; border-radius:10px;}
nav a {
    float:left;
    display:block;
    width:23%;
    padding:1%;
    font-size:110%;
    color:#ffe9ba;
    text-align:center;
}
section { clear:both;}
img { max-width: 100%; height: auto; border-radius:10px; }
.slajd {
    float:left;
    display:block;
    width:13%;
    text-align:center;
    border-radius:10px;
    background:#ffe9ba;
    padding:1% 0.5%;
    margin:1%;
}
.slajd:hover { background:#45525a; }
footer { clear:both; }
.galeria{ margin:auto; width:90%;}
```

Zad.2.

Kolejnym zadaniem jest zadeklarowanie innych reguł formatowania dla poszczególnych rozmiarów okien za pomocą tzw. **zapytania mediów** (media queries).

Zapytanie o media może mieć postać jak na Listingu 3.

Listing 3. Zapytanie o media

```
@media screen and (max-width: 720px) {
    nav a { width: 48%;
            padding: 10px 0;
            font-size:140%;
        }
    .slajd {width: 45%; }
}

@media screen and (min-width: 481px) and (max-width: 720px) {
    body { font-size: 13px; }
    h1 { font-size: 23px; }
    h2 { font-size: 18px; }
}
```

Po dodaniu powyższych reguł na koniec do arkusza *style.css*, przy odpowiednio małej szerokości okna przeglądarki – wynik będzie jak na rysunku 2.

UWAGA 1

Na listingu 3, typem medium w obu przypadkach jest **screen**, czyli ekran komputerowy. Istnieją również takie typy, jak syntezytor mowy, materiały drukowane nieprzezroczyste, odbiorniki TV, czy też drukarki Braille. Więcej informacji na temat mediów można znaleźć na stronie:

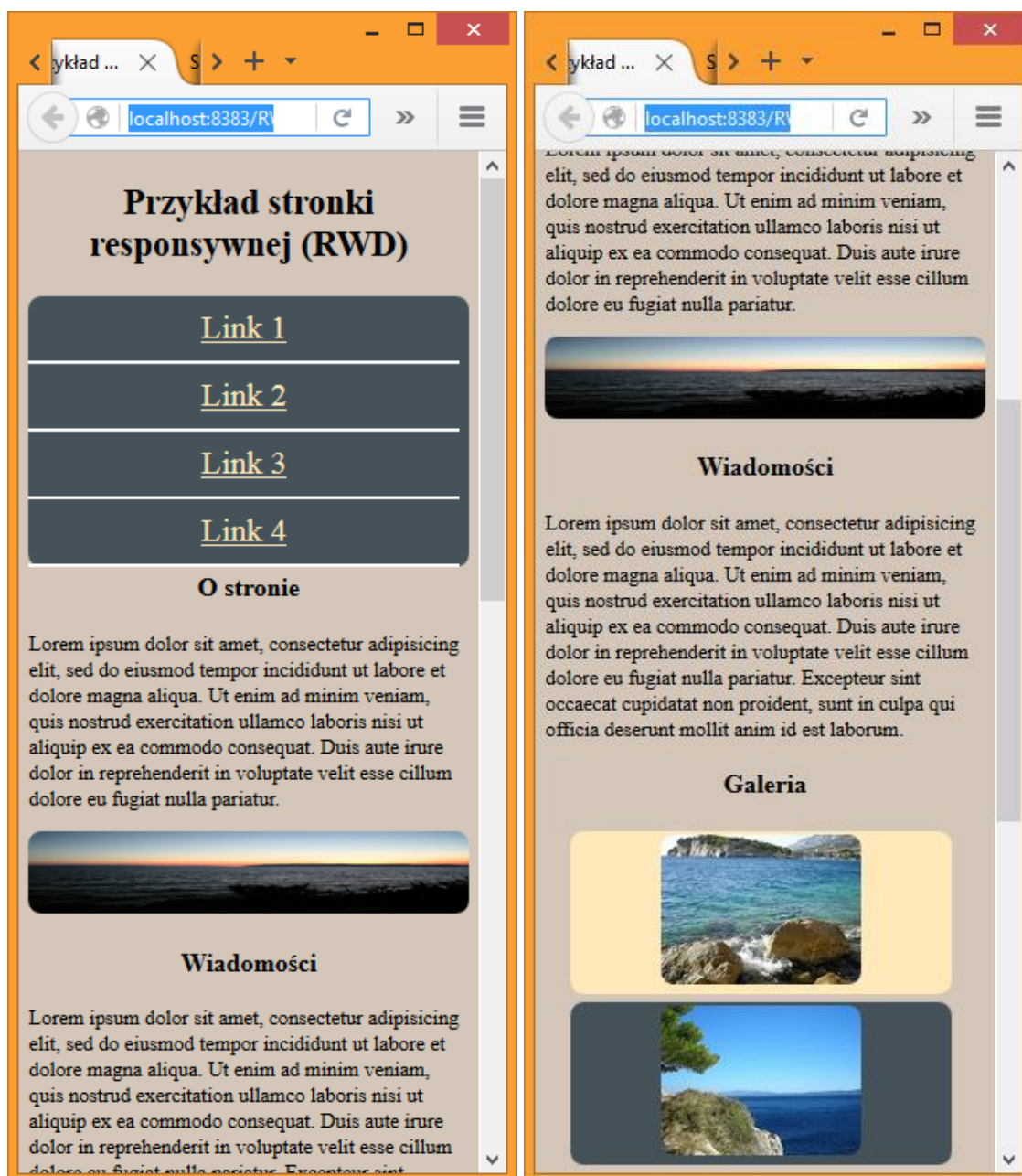
http://www.w3schools.com/css/css_mediatypes.asp



Rys. 2. Widok strony po dodaniu zapytania o media

Zad. 3.

Do reguł CSS dodaj kolejne zapytanie o media tak aby przy rozdzielczościach mniejszych niż **481 px** strona renderowała się podobnie jak na rysunku 3.



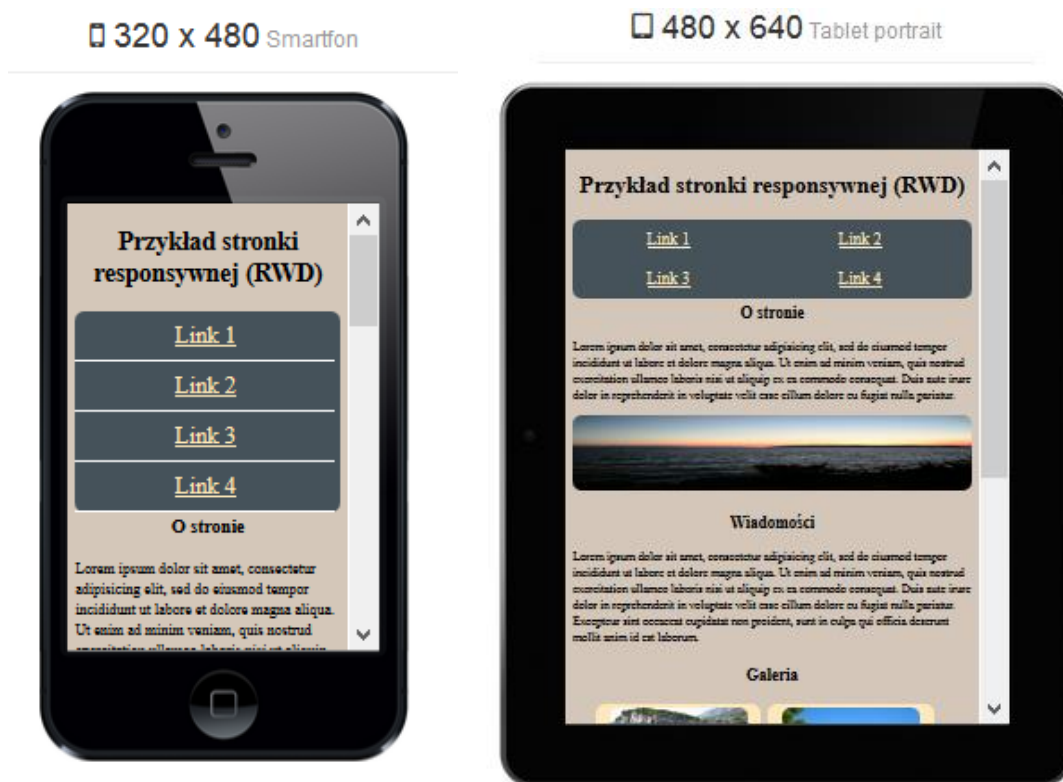
Rys. 3. Widok strony po dodaniu zapytania o media <481px

UWAGA 2

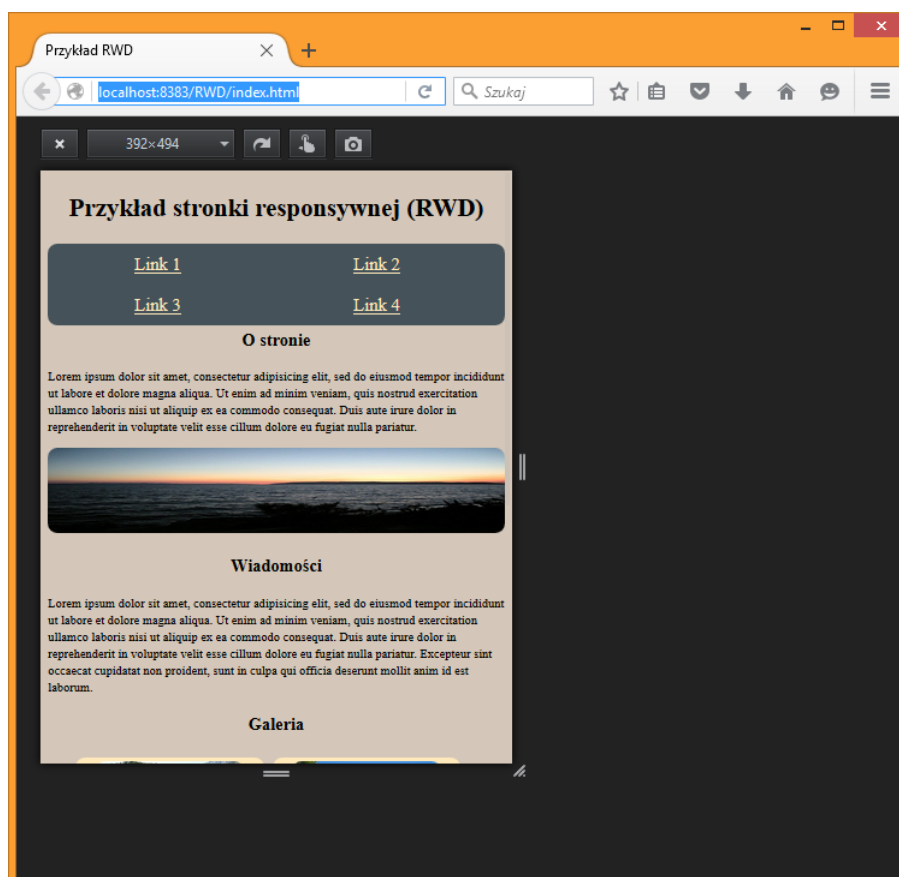
W celu symulacji renderowania strony na urządzeniach o różnych rozdzielczościach można skorzystać ze strony:

<http://testrwd.pl/>

Przykładowe widoki stron symulowane na powyższej stronie przedstawia rysunek 4. Widok responsywny oferuje również narzędzie dostępne w przeglądarce Firefox (Narzędzia -> Widok responsywny) – rysunek 5.



Rys. 4. Widok strony na symulatorze testrwd



Rys. 5. Widok responsywny w Firefox

Korzystając z media queries można również ustalić oddzielne reguły dla urządzeń trzymanyh w pozycji wertykalnej (**portrait**) i horyzontalnej (**landscape**). Przykład wykorzystujący orientację został pokazany na Listingu 4.

Listing 4. Przykład reguł z uwzględnieniem orientacji urządzenia

```
@media only screen and (min-width : 768px)
  and (max-width : 1024px)
  and (orientation : landscape)
{
  /* Reguły CSS poszczególnych elementów strony */
}
```