

## Tworzenie Strony responsywnej

**Strona responsywna** lub ang. **responsible web design (RWD)** – to strona, która dopasowuje swoją

szerokość i układ elementów do rozdzielczości na jakiej jest wyświetlana. Strony responsywne „narodziły” się od wersji CSS3 (arkusza stylów w wersji 3) ponieważ mocno rozbudowano możliwości **Media Queries**. Został on wtedy poszerzony o nowe możliwości *min-width* oraz *maxwidth*, dzięki którym możliwe jest rozpoznawanie rozdzielczości okna użytkownika. W zależności od wykrytych rozdzielczości, wczytujemy różne arkusze stylów, które tę samą strukturę HTML wyświetlają w całkiem inny sposób. Niektóre elementy są przesuwane, niektóre (te mniej ważne) całkowicie znikają.

Niezależnie czy chcesz **zrobić stronę responsywną** od początku, czy chcesz **przerobić stary szablon na szablon responsywny**, musisz wiedzieć o kilku najważniejszych rzeczach.

### Zasady tworzenie stron responsywnych

1. Pierwszym krokiem zawsze jest dodanie odpowiednich *meta-tagów*. Zapewniają one poprawne wyświetlanie strony na urządzeniach mobilnych, m.in. **zapobiegają skalowaniu**.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

Meta-tag *viewport* zapewnia poprawne dostosowanie szerokości oraz brak powiększenia. Jeżeli zapomnisz go umieścić, strona będzie prezentować się źle.

2. Kolejnym krokiem jest **zaplanowanie kompozycji szablonu**. Musisz zdecydować jakie elementy mają być wyświetlane na pełnej rozdzielczości, a jakie na telefonie. W standardowej konfiguracji, na pełnej rozdzielczości wyświetlana jest **część główna strony oraz boczne menu, co daje razem dwie kolumny**. Oczywiście niektóre strony mogą wcale nie mieć menu, lub mieć aż 2 boczne kolumny + środkowa będąca treścią strony.

Musisz mieć świadomość, że **każda strona dopasowana do urządzeń mobilnych musi być budowana na jednej kolumnie**. Jedna kolumna jest jedynym rozwiązaniem, jakie zapewnia nam poprawne skalowanie. Kiedy szablon ma jedną kolumnę, ustawiamy jego szerokość na *width:100%*, dzięki temu szerokość kolumny dopasuje się do ekranu niezależnie od jego wielkości.

### JAK TWORZYĆ STRONĘ RESPONSYWNA - PRZYKŁAD

Przykład implementacji prostego szablonu responsywnego. W pełnej okazałości strona posiada 2 kolumny, po zmianie wielkości okna cała prawa kolumna menu znika. Wiąże się to oczywiście z ukryciem niektórych informacji **ale nie ma innego wyjścia**. Należy przemyśleć, jakie informacje zostaną ukryte dla użytkowników urządzeń mobilnych. Jednym z rozwiązań jest wyświetlanie na telefonie tylko logo + menu + treść strony. Znika ankieta, wtyczka facebooka, linki oraz informacje o autorze. Niestety na ekranie 320×480 **nie zmieścimy** tyle co na 1280×600.

Jeżeli zastanowisz się nad kompozycją, czas wziąć się za **arkusz stylów**.

```
/*najwieksza rozdzielczosc laptopów*/
```

```
@media only screen and (max-width: 1000px) {
```

```
#prawe_menu { display: block; }
```

```
}
```

```
/*średni telefon lub iPad*/
```

```
@media only screen and (max-width: 680px) {
```

```
#prawe_menu { display: none; }
```

```
}
```

Jak widzisz są tu tylko dwie reguły **Media queries**. To ile ich dodasz zależy od Twojego serwisu. Umieszczenie tego kodu w arkuszu stylów powoduje, że dla urządzeń z szerokością ekranu w granicach 1-680px prawe menu znika, natomiast dla urządzeń z rozdzielczością większą niż 680px menu się pojawia. Rozbudowane serwisy posiadają po 5-7 reguł, które **znacząco zmieniają wygląd strony nawet przy nieznacznych** zmianach rozdzielczości.

Istnieje ważna zasada podczas planowania kompozycji. Możesz a nawet **musisz ukrywać odpowiednie elementy na mniejszych rozdzielczościach** używając *display: none*. **Nie wolno natomiast utworzyć dwóch osobnych menu**, jednego dla telefonów drugiego dla komputerów. Nie można pokazywać ich i ukrywać w zależności od wykrytej rozdzielczości. Jest to powielanie treści, przez co strona może być nieciekawa dla wyszukiwarki, a w ekstremalnym wypadku grozi banem *duplicate-content*. Na stronie muszą znajdować się pojedyncze elementy, a ich położeniem sterujemy za pomocą arkusza stylów.

## Responsywne menu

Budowa **responsywnego menu** nie należy do najłatwiejszych i wymaga dużo pracy. Menu responsywne charakteryzuje się tym, że zawsze jest poziome i rozwija się dopiero po kliknięciu. Należy wykonać je w CSS oraz JQuery. Przykład wykonania w dołączonym szablonie strony responsywanej.

## Obrazki

Ważnym aspektem, na który należy zwrócić szczególną uwagę są obrazki. W **szablonie responsywnym** obrazki muszą się automatycznie skalować wraz z szerokością strony. Aby osiągnąć taki efekt, można dodać do arkusza stylów takie globalne ustawienie dla obrazków:

```
img{ max-width: 100%;
```

```
height:auto;
```

```
width:auto;}
```

Oczywiście reguła sprawdzi się tylko dla obrazków będących statyczną częścią treści strony np. zdjęć w artykule strony.

## Elementy graficzne szablonu

Tak jak w przypadku obrazków, szerokość elementów graficznych (np. logo lub tytuł menu) powinna być podana w wartościach procentowych. Jeżeli nie jest możliwe ustalenie wartości procentowych, należy ustawić wartości procentowe **dla pojemnika będącego rodzicem**, a następnie podmieniać element graficzny (np. logo) na mniejsze gdy braknie na nie miejsca – w zależności od wykrytej rozdzielczości.

Tworząc szablon responsywny możemy ustalać wysokość elementów graficznych stosując bezwzględną jednostkę *px*. Wynika to z faktu, że strona zbudowana na jednej kolumnie, musi dostosować się do urządzenia poziomo, pionowo nigdy miejsca nam nie braknie.

## Strony ze skomplikowaną szatą graficzną

Jeżeli jesteś właścicielem jakiejś firmy, ze skomplikowanym motywem graficznym (np. restauracje), niezbędna podczas przeróbki może być pomoc grafika, który wykonywał szablon strony. W takim wypadku przeważnie dla większości elementów graficznych nie można ustalić szerokości procentowej, ponieważ w przypadku zmiany rozmiaru okna grafika będzie niewyraźna (skalowanie).

Niestety, w takim konkretnym przypadku, wszystkie elementy graficzne trzeba ukrywać i zastępować mniejszymi odpowiednikami.

## Rozmiary elementów i czcionek

W szablonie responsywnym powinniśmy używać jednostek procentowych. **Priorytetem jest nadanie procentowej szerokości**, aby mogła się zmieniać wraz ze zmianą rozdzielczości. Nie powinniśmy ustalać sztywnych rozmiarów wszelkich obrazków, one także powinny skalować się razem z szablonem.

Wartości stałe powinniśmy natomiast nadawać czcionkom, trudno przewidzieć jak będzie wyglądać czcionka, której rozmiar zależny jest od innych elementów. Czcionki dla komputerów PC są średnio o 1-2*px* większe niż czcionki dla urządzeń mobilnych.

## Testowanie strony responsywnej

Bardzo dobry dodatek do testowania naszego szablonu udostępnia najnowsza wersja przeglądarki Firefox. Pozwala ona z opcji włączyć *Narzędzia dla twórców witryn > Widok responsywny*. Dzięki temu możemy podglądać stronę w różnych zestawieniach rozdzielczości.

## Prosty szablon responsywny

Przykład mało skomplikowanego szkieletu szablonu responsywanego

Jego kod HTML jest następujący:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset=utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<link rel="stylesheet" href="style.css" />

<title>Przykład strony responsywnej</title>

</head>

<body>

  <header >

    <div id="logo">

      <h1>Logo strony</h1>

    </div>

  </header>

  <div id="wrapper">

    <main id="content">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh augue, suscipit a, scelerisque sed, lacinia in, mi. Cras vel lorem. Etiam pellentesque aliquet tellus. Phasellus pharetra nulla ac diam. Quisque semper justo at risus. Donec venenatis, turpis vel hendrerit interdum, dui ligula ultricies purus, sed posuere libero dui id orci. Nam congue, pede vitae dapibus aliquet, elit magna vulputate arcu, vel tempus metus leo non est.<br><br> Etiam sit amet lectus quis est congue mollis. Phasellus congue lacus eget neque. Phasellus ornare, ante vitae consectetur consequat, purus sapien ultricies dolor, et mollis pede metus eget nisi. Praesent sodales velit quis augue. Cras suscipit, urna at aliquam rhoncus, urna quam viverra nisi, in interdum massa nibh nec erat.</main>

    <nav id="menu">

      <h5>Menu główne</h5>

      <ul>

        <li><a href="#">Strona główna</a></li>

        <li><a href="#">Link 1</a></li>

        <li><a href="#">Link 2</a></li>

      </ul>

    </nav>

  </div>

</body>

</html>
```

```

        <li><a href="#">Link 3</a></li>

    </ul>

</nav>

</div>

</body>

</html>

```

Posiada też kod CSS:

```

/* ogólne ----- */

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

body { font: 16px Verdana; line-height:1.6; background: #EDE6DE; }

#wrapper { width:90%; overflow:hidden; margin: 0 auto; }

a {color: #354458;}

/* logo ----- */

header { height:100px; background: #354458; margin: 0 0 10px 0; }

header h1 {padding: 20px; color: #E9E0D6;}

/* tresc ----- */

#content { width:75%; background: #EB7260; float:left; padding:1%; overflow:hidden; }

/* menu ----- */

#menu { background: #548BC2; width:20%; float:right; padding:1%; overflow:hidden; }

#menu h5 {padding:3px 3px 15px 3px; font-size:14px;}

#menu ul li {list-style-type:none; padding:0 0 0 10px;}

```

Szablon na wszystkich rozdzielczościach wygląda tak samo. Jedyną jego zaletą jest szerokość wyrażona w procentach, więc dostosowuje się do szerokości okna przeglądarki.

Chcemy **aby szablon był responsywny** i aby prawe menu przesunęło się na górę i zamieniło w menu poziome. W tym celu musimy dodać do arkusza stylów znacznik **media queries** zmieniający kod CSS poniżej określonej rozdzielczości. Ja użyję rozdzielczości progowej *900px*. **Aby menu przeskoczyło na górę**, trzeba nadać mu atrybut *position: absolute*. To przez to, że w kodzie HTML menu jest w kodzie jako drugie. Skoro będzie to menu poziome, należy ustawić jego szerokość na *width: 100%*. Należy

także zwiększyć szerokość treści strony, ponieważ obecnie zajmuje ona tylko 75%. Po przeskoczeniu menu powinna zajmować *width: 98%*.

Aby **linki w menu układały się poziomo** wystarczy zmienić im atrybut *display* i ustawić go na *inline*. To już wszystko. Ostatnim zabiegiem kosmetycznym jest ukrycie napisu *Menu główne* oraz dodanie poziomych kresek po każdym elemencie menu.

Cały opisany kod wygląda następująco:

```
@media only screen and (max-width: 900px) {  
  
#menu {width:88%; position:absolute; height:30px;}  
  
#menu h5 {display: none;}  
  
#menu ul li {display: inline; }  
  
#menu ul li:after {content: " |";}  
  
#content {width: 98%;position:relative; top: 30px;margin-top:5%;}  
  
}
```