

# Responsywny interfejs

---

## 1. Wprowadzenie

Współcześnie użytkownicy Internetu przeglądają strony internetowe przy użyciu różnorodnych urządzeń. Jest to efektem dynamicznego rozwoju rynku smartphonów i tabletów w ostatnich latach. Interfejsy, projektowane z myślą o przeglądaniu stron internetowych na komputerze, nie są wygodne w użyciu na urządzeniach przenośnych. Wynika to z bardzo zróżnicowanych charakterystyk ekranów:

- zewnętrzne monitory: typowo ekrany panoramiczne, format 16:9 lub 16:10, powyżej 19”, rozdzielczości: 1440x900, 1680x1050, 1920x1080 (Full HD) i wyższe,
- laptopy: typowo ekrany panoramiczne, format 16:9 lub 16:10, powyżej 12” (mniejsze urządzenia z reguły zaliczane do kategorii netbooków), rozdzielczości: 1360x768, 1280x800 i wyższe,
- tablety: typowo ekrany 7” – 12”, mogą być używane w orientacji pionowej lub poziomej, rozdzielczości: 800x600, 1024x600, 1024x768, 1280x800 i wyższe,
- smartphony: typowo ekrany 4” – 6”, najczęściej używane w orientacji pionowej, rozdzielczości: 480x800, 800x1280, 768x1360, 1080x1920 i wyższe.

Podstawowym problemem przy dostosowywaniu interfejsu do obsługi różnych urządzeń jest fakt, że smartphony są wykorzystywane w orientacji pionowej (wąski ekran), podczas gdy monitory komputerów są panoramiczne (szeroki ekran, ale mniej miejsca do prezentacji treści w pionie). Specyfikacja stylów CSS obejmuje jednostki względne, których rzeczywisty rozmiar jest uwarunkowany cechami urządzenia, np.: procentowe (względem elementu nadrzędnego lub okna przeglądarki), em (względem rozmiaru czcionki), rem (root em – względem bazowego rozmiaru czcionki). W większości przypadków jednostki te nie są jednak wystarczające, aby obsłużyć tak skrajne różnice, jak pomiędzy smartphonami i ekranami komputerów. Często konieczna jest nie tylko zmiana relatywnych wymiarów poszczególnych części strony, ale przygotowanie zupełnie innego układu elementów w zależności od urządzenia. Przykładowo: na panoramicznym ekranie komputera typowo wykorzystuje się równoległe kolumny (kolumna z nawigacją, kolumna z zasadniczą treścią, kolumna na reklamy), podczas gdy na ekranie smartphona elementy te należy umieścić jeden pod drugim, aby były czytelne dla użytkowników.

Jeśli strona internetowa nie oferuje interfejsu dedykowanego dla urządzeń mobilnych, przeglądarki wyświetlają typowo stronę w pomniejszeniu tak, jak wyglądałaby ona na ekranie komputera. Użytkownik, chcąc nawigować po stronie lub przeczytać jej treść musi ją powiększyć a następnie przesunąć zarówno w pionie jak i w poziomie. Nawigowanie po stronie staje się przez to niewygodne. Pojawiają się również dodatkowe problemy, np.:

- symulowanie zachowań myszki na ekranie dotykowym – wiele stron wykorzystuje selektory/zdarzenia hover, mouseenter, mouseleave, które są związane z zachowaniem kursora myszki na ekranie komputera; niektóre przeglądarki mobilne nie generują takich zdarzeń, a jedynie symulują kliknięcia w chwili dotknięcia ekranu dotykowego palcem; inne przeglądarki w chwili dotknięcia ekranu emitują zarówno zdarzenie hover jak i zdarzenie click/action, co może utrudniać nawigację po stronie,

- brak obsługi text reflow w wielu przeglądarkach (m.in. Chrome mobile, Safari dla systemu iOS) – funkcja określana jako text reflow pozwala na zmianę szerokości szpalty tekstu tak, aby była ona dopasowana na szerokości ekranu urządzenia przy powiększeniu, które użytkownik wybrał w danym momencie.

Prawidłowe obsłużenie opisanych powyżej różnic pomiędzy przeglądarkami na komputerach i na urządzeniach mobilnych z dotykowymi ekranami wymaga często przygotowania osobnych zestawów stylów dla różnych grup urządzeń.

## 2. Reguły @media

Możliwość definiowania różnych stylów dla różnych urządzeń została wprowadzona wraz ze specyfikacją CSS 2.0, w której pojawiły się reguły @media. Możliwości te zostały rozszerzone w wersji CSS 3.0, która poza predefiniowanymi grupami urządzeń (screen, print, handheld itd.), wprowadza dodatkowo selektory cech, np.: min-width, max-width, aspect-ratio, resolution, orientation. Spośród zdefiniowanych selektorów cech najczęściej wykorzystuje się te związane z szerokością wyświetlacza: min-width i max-width. Pozwalają one na rozróżnienie, czy mamy do czynienia z panoramicznym ekranem, czy też z wąskim ekranem używanego w orientacji pionowej smartphona.

Reguły @media mogą być umieszczane wewnątrz plików z rozszerzeniem .css, przykładowo:

```
@media (max-width: 768px) {  
    div#main-container{  
        width: 100%;  
        margin: 0px;  
    }  
}
```

Lub w obrębie atrybutu media elementu link, który jest wykorzystywany do dołączania arkusza stylów do dokumentu HTML:

```
<link rel="stylesheet" media="(max-width: 768px)"  
      href="max-768px.css">
```

Typowo konieczna jest modyfikacja pozycjonowania głównych sekcji strony (nagłówek, menu/nawigacja, sekcja zawierająca treść podstrony, stopka) w zależności urządzenia użytkownika. Często jednak konieczna będzie również modyfikacja stylów dla elementów konkretnych podstron (np. zmiana układu formularza dla smartphonów). Aby uniknąć tworzenia zestawów wielu specyficznych reguł pozycjonowania dla poszczególnych elementów i podstron witryny, konieczne jest opracowanie ustandaryzowanego i systematycznego rozwiązania dla problemu responsywności interfejsu.

## 3. Gridowy model responsywnego interfejsu

Najczęściej wykorzystywanym obecnie podejściem do tworzenia responsywnych interfejsów jest wykorzystanie modelu opartego o siatkę elementów, o określonej liczbie kolumn. Poszczególne elementy witryny mogą zajmować różną liczbę kolumn w zależności od urządzenia, na którym wyświetlana jest strona. Najczęściej

wykorzystuje się układ 12 kolumn. Jest to najmniejsza liczba, która dzieli się przez 2, 3 i 4. Dysponując 12 kolumnami można przykładowo tworzyć następujące układy:

- 6+6 – podział strony pół na pół,
- 4+4+4 – podział strony na 3 kolumny,
- 3+3+3+3 – podział strony na 4 kolumny,
- 3+9 – podział  $\frac{1}{4} + \frac{3}{4}$  – lewa kolumna o szerokości 25% strony, prawa przeznaczona na zasadniczą treść o szerokości 75%,
- 4+8 – podobnie jak powyżej, ale z podziałem 33%+66%,
- 3+6+3 – środkowa kolumna o szerokości 50% i dwie boczne o szerokości 25% każda,
- 2+8+2 – podobnie jak wyżej, ale środkowa kolumna szersza kosztem kolumn bocznych.

Dla poszczególnych elementów strony definiujemy ile kolumn powinny one zajmować na urządzeniach różnego typu. Aktualnie najczęściej wykorzystuje się 4 grupy urządzeń:

- małe urządzenia, smartphony – poniżej 768px,
- tablety – powyżej 768px,
- netbooki, laptopy o małych ekranach – powyżej 992px,
- laptopy, komputery stacjonarne – powyżej 1200px.

Definiując liczbę kolumn zajmowanych przez poszczególne elementy zazwyczaj przyjmuje się następujące zasady:

- jeśli element zajmuje X kolumn na mniejszym urządzeniu, to na większych również zajmuje X kolumn, o ile jawnie nie zdefiniowano inaczej (kolokwialnie mówiąc: jeśli coś mieści się na 6 kolumnach na małym ekranie, to na większym ekranie tym bardziej zmieści się na 6 kolumnach),
- jeśli na większym urządzeniu coś zajmuje X kolumn, to na mniejszym domyślnie zajmuje całą szerokość 12 kolumn (jeśli coś się mieści na większym ekranie na 6 kolumnach nie oznacza to, że na mniejszym ekranie również się zmieści).

Dzięki tym domyślnym założeniom nie jest konieczne definiowanie liczby kolumn dla każdej grupy urządzeń w każdym elemencie. Podejście takie określane jest jako *mobile first* (najpierw urządzenia mobilne) – domyślne style są zdefiniowane dla urządzeń mobilnych (najmniejszych) i są one promowane dla urządzeń większych o ile jawnie nie zostaną zdefiniowane inne style.

Istnieje wiele gotowych bibliotek, które oferują gridowy model interfejsu. Do najpopularniejszych należą: biblioteka Bootstrap (<http://getbootstrap.com>) oraz biblioteka Foundation (<http://foundation.zurb.com/>).