

Podstawy informatyki

HTML5 / CSS3

Laboratorium 11

Responsywność i Media Queries

Czas trwania: 2 × 45 min**Tryb:** studia stacjonarne**Jednostka:** Katedra Elektrotechniki i Technologii Nadprzewodnikowych**Prowadzący:** dr inż. Michał Łanczont

Cel ćwiczenia

Celem jest przygotowanie układu reagującego na szerokość viewportu za pomocą podejścia mobile-first, elastycznych wymiarów i zapytań medialnych. Student ma rozumieć, że breakpoint powinien wynikać z treści i układu, a nie wyłącznie z nazw urządzeń.

Wstęp

Responsywny interfejs dostosowuje układ do dostępnej przestrzeni i możliwości urządzenia. Nie oznacza to tworzenia osobnej strony dla każdego modelu telefonu; zamiast tego projektuje się elastyczne komponenty i dodaje punkty przełomowe tam, gdzie treść faktycznie przestaje mieścić się czytelnie.

Viewport

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

Jest to standardowa konfiguracja dla współczesnych stron mobilnych. Nie zastępuje ona elastycznych szerokości, Flexboxa, Grid ani media queries.

Jednostki

Jednostka	Typowe znaczenie
%	wartość względem odpowiedniego wymiaru kontenera; przy szerokości często względem szerokości zawierającego bloku
em	względem obliczonego rozmiaru fontu bieżącego elementu (dla font-size względem rodzica)
rem	względem rozmiaru fontu elementu głównego html
vw/vh	procent wymiaru viewportu
px	piksel CSS; nadal użyteczny np. dla cienkich obramowań lub kontrolowanych wartości – nie trzeba go całkowicie unikać

Mobile-first

W podejściu mobile-first bazowe style odpowiadają małym szerokościom, a rozszerzenia wprowadza się najczęściej przez **min-width**.

```
.kolumna {  
  width: 100%;  
}
```

```
@media (min-width: 48rem) {  
  .kolumna {  
    width: 50%;  
  }  
}
```

Mobile-first ułatwia stopniowe rozszerzanie interfejsu, ale samo w sobie nie powoduje, że przeglądarka “pobiera tylko mały CSS” – cały arkusz nadal może zostać pobrany. Wydajność zależy od organizacji i dostarczania zasobów.

Składnia media query

Typ medium jest opcjonalny; bez niego przyjmowane jest **all**. W aktualnej praktyce podstawowe typy to **all**, **screen** i **print**. Starsze typy urządzeń z dawnych specyfikacji nie powinny być podstawą współczesnego projektu.

```
@media (min-width: 768px) {  
  .main-layout {  
    flex-direction: row;  
  }  
}  
  
@media (min-width: 768px) and (orientation: landscape) {  
  .panel {  
    max-width: 70rem;  
  }  
}  
  
@media print {  
  nav,  
  .element-dekoracyjny {  
    display: none;  
  }  
}
```

Media features obejmują **m.in. width/min-width/max-width, orientation, resolution**, a także cechy interakcji i preferencji użytkownika. W dalszej nauce warto poznać **prefers-reduced-motion** i **prefers-color-scheme**.

Operatory logiczne

```
@media screen and (min-width: 768px) and (orientation: landscape) {  
  /* wszystkie warunki muszą być spełnione */  
}  
  
@media (max-width: 480px), print {  
  /* co najmniej jedno zapytanie z listy */  
}
```

Breakpointy

Wartości 576, 768, 992 czy 1200px są popularnymi punktami spotykanymi w frameworkach, ale nie są nazwami konkretnych urządzeń ani normą. W projekcie należy wybrać punkt, w którym układ zaczyna wyglądać źle i wtedy dodać regułę.

Przykład układu trzykolumnowego

```
.main-layout {
  display: flex;
  flex-direction: column;
  gap: 1rem;
}

.kolumna {
  width: 100%;
  padding: 1rem;
}

@media (min-width: 768px) {
  .main-layout {
    flex-direction: row;
  }

  .kolumna {
    flex: 1;
    width: auto;
  }
}
```

Użycie `flex: 1` może być wygodniejsze niż ręczne 33.33%, ponieważ uwzględnia również `gap`.

Przykład kompletny – karta responsywna bez nazw urządzeń

```
.cards {
  display: grid;
  grid-template-columns: 1fr;
  gap: 1rem;
}

@media (min-width: 42rem) {
  .cards {
    grid-template-columns: repeat(2, 1fr);
  }
}

@media (min-width: 70rem) {
  .cards {
    grid-template-columns: repeat(3, 1fr);
  }
}
```

Wartości 42rem i 70rem opisują moment zmiany układu w tym konkretnym projekcie. Nie muszą odpowiadać nazwanym modelom telefonu czy tabletu.

Typowe błędy responsywności

- brak viewportu i testowanie wyłącznie na desktopie,
- stałe szerokości większe niż viewport,
- breakpointy kopiowane z frameworka bez sprawdzenia własnej treści,
- zbyt wiele bardzo podobnych media queries utrudniających utrzymanie CSS,
- ukrywanie ważnej treści tylko dlatego, że ekran jest mały,

- założenie, że mobile-first automatycznie zmniejsza rozmiar pobieranego CSS.

Przebieg ćwiczenia

Zadanie 1. Układ mobilny

Do wykonania

Utwórz `responsive.html` i `responsive.css`. Dodaj viewport. Zbuduj `.main-layout` z trzema kolumnami. Bazowo ustaw `flex-direction: column`, szerokość kolumn 100%, padding i gap. Sprawdź widok w trybie urządzenia mobilnego DevTools.

Zadanie 2. Pierwszy breakpoint

Do wykonania

Dodaj `@media (min-width: 768px)`. Przełącz kontener na row i nadaj kolumnom `flex: 1`. Porównaj ten wariant z ręcznym `width: 33.33%` przy włączonym gap.

Zadanie 3. Drugi breakpoint i widoczność treści

Do wykonania

Dodaj element `.tylko-desktop`, bazowo ukryty. Przy wybranym drugim breakpointie (np. 1200px) pokaż go, zwiększ rozmiar pisma i zmień tło. Zapisz, czy 1200px wynika z potrzeby treści czy tylko z przyjętego przykładu.

Zadanie 4. Orientacja i druk

Do wykonania

Dodaj jedną regułę wykorzystującą `orientation: landscape`. Następnie utwórz `@media print`, które ukrywa nawigację i dekoracje, ustawia czytelny kolor tekstu oraz usuwa zbędne cienie.

Zadanie 5. Responsywne obrazy

Do wykonania

Dodaj obraz o `max-width: 100%; height: auto`. Umieść go w kolumnie i sprawdź, że nie wypycha kontenera. Następnie ustaw na chwilę stałą szerokość większą od viewportu i porównaj efekt.

Zadanie 6. Dobór breakpointu na podstawie treści

Do wykonania

Usuń na chwilę ustaloną wartość 768px. Zwążaj i rozszerzaj stronę, obserwując moment, w którym trzy kolumny przestają być czytelne. Zapisz własny breakpoint i uzasadnij wybór jednym zdaniem w komentarzu CSS.

Zadanie końcowe

Integracja

Przebuduj wybraną wcześniejszą stronę na układ mobile-first. Musi mieć co najmniej dwa breakpoints, elastyczne media, sensowny układ dla druku oraz brak poziomego przewijania całej strony przy typowych małych szerokościach.

Kryteria zaliczenia

- poprawny viewport,
- układ mobile-first bez fałszywego założenia “same px są złe”,
- breakpoint dobrany do zachowania treści,
- poprawne jednostki em/rem,
- responsywne obrazy i test wydruku/małego ekranu.

Sprawozdanie

Do odbioru należy przygotować pliki źródłowe utworzone podczas ćwiczenia. Jeżeli prowadzący wymaga sprawozdania, powinno ono zawierać: krótki opis rozwiązania, najważniejsze fragmenty kodu, zrzut ekranu wyniku oraz wnioski dotyczące napotkanych problemów. Pliki HTML i CSS należy zachować w strukturze katalogów zgodnej z instrukcją.

Materiały pomocnicze

- HTML Living Standard: <https://html.spec.whatwg.org/>
- MDN Web Docs – HTML: <https://developer.mozilla.org/docs/Web/HTML>
- MDN Web Docs – CSS: <https://developer.mozilla.org/docs/Web/CSS>
- Walidator HTML: <https://validator.w3.org/>