

CSS Grid w praktyce

Czas: 2 × 45 min

praca indywidualna

Cel: Budowanie układu strony siatką CSS Grid z nazwanymi obszarami oraz galerii, która sama dopasowuje liczbę kolumn do szerokości okna; porównanie z rozwiązaniem we Flexboxie.

Po zajęciach potrafisz:

- definiuje siatkę kolumn i wierszy z jednostką `fr`, `repeat` i `minmax`
- nazywa obszary (`grid-area`) i rozmieszcza je właściwością `grid-template-areas`
- buduje galerię `auto-fit/minmax` i rozróżnia `auto-fit` od `auto-fill`
- rozciąga element na kilka komórek (`span`)
- dobiera Grid albo Flexbox do problemu i uzasadnia wybór

Sprzęt i materiały

- komputer z Windows 11 i przeglądarką Chrome, Edge lub Firefox
- folder `r1_14_start` z Pliki_do_zadan (`index.html`, `style.css`, folder `img`)

Oprogramowanie

- **Visual Studio Code z rozszerzeniem Live Server** — podgląd odświeżany po zapisie code.visualstudio.com
- **Grid Garden (opcjonalnie)** — gra w przeglądarce, wersja polska cssgridgarden.com/#pl
- **Narzędzia F12 — nakładka Grid** — plakietka „grid” w Elements; w zakładce Layout włącz „Show area names” developer.chrome.com/docs/devtools/css/grid

Przygotowanie przed zajęciami

- Przypomnij sobie z wykładu 1.11: `fr`, `repeat()`, `minmax()`, `grid-template-areas`.
- Skopiuj folder `r1_14_start` do folderu `nazwisko_imie` jako `zad14`. **Zachowaj wynik** — w zadaniu 15 przerobisz tę stronę na responsywną.

Bezpieczeństwo

- Pracuj na kopii plików; nie zmieniaj pliku `index.html` (wyjątek: plik `galeria-flex.html` w G9).
- Zmiany wprowadzane w F12 (np. usuwanie elementów) działają tylko w podglądzie — nie zapisują się w pliku.

Przebieg zadania

1 Rozgrzewka i rozpoznanie (maks. 10 min)

1. Opcjonalnie przejdź poziomy 1–8 gry Grid Garden; zanotuj ostatni ukończony poziom.
2. Uruchom `zad14/index.html` przez Live Server i zrób zrzut strony przed zmianami. Zwróć uwagę, że wszystkie bloki stoją jeden pod drugim, a ilustracje mają pełną szerokość.

2 Układ strony z obszarami

Docelowy układ (szkic, kontener `.strona`, maks. 1200px):

nagłówek		
menu		
tresc (1fr)	boczny 280px	
stopka		

1. G1. Przypisz nazwy obszarów: `.nagłówek` → `nagłówek`, `.menu` → `menu`, `.tresc` → `tresc`, `.boczny` → `boczny`, `.stopka` → `stopka` (właściwość `grid-area`).
2. G2. Zamień `.strona` w siatkę o kolumnach `1fr 280px` i rozmieść obszary według szkicu właściwością `grid-template-areas` (każdy wiersz w osobnym cudzysłowie).
3. G3. Ustaw wiersze `auto auto 1fr auto` i minimalną wysokość `.strona` równą `100vh` — stopka ma być na dole okna. Ułóż odnośniki menu w poziomie (Flexbox z zadania 13).
4. G4. W F12 zaznacz `.strona`, włącz nakładkę Grid z nazwami obszarów i zrób zrzut. Odczytaj szerokość kolumny `tresc` przy oknie 1280px.

3 Galeria auto-fit / minmax

`repeat(auto-fit, minmax(180px, 1fr))` = tyle kolumn, ile się zmieści, każda co najmniej 180px, a wolne miejsce dzielone równo.

1. G5. Zbuduj galerię `.galeria` jako siatkę z kolumnami `repeat(auto-fit, minmax(180px, 1fr))` i odstępem 12px. Zmieniaj szerokość okna i zapisz liczbę kolumn przy 1280px, 1000px i 800px.
2. G6. Porównaj auto-fit i auto-fill: w F12 usuń z podglądu 6 ostatnich elementów `figure` (prawy przycisk → Delete element), a potem zmieniaj w zakładce Styles `auto-fit` na `auto-fill`. Opisz różnicę. Odśwież stronę, aby przywrócić figury.
3. G7. Wyróżnij pierwszą ilustrację: ma zajmować 2 kolumny i 2 wiersze (`grid-column: span 2; grid-row: span 2`).
4. G8. Sprawdź, czy przy 1280px galeria nie ma pustych dziur. Jeśli są, wypróbuj `grid-auto-flow: dense` i opisz, co się zmieniło (w pliku zostaw wersję, która lepiej wygląda).

4 Porównanie z Flexboxem

Ta sama galeria zbudowana Flexboxem — w osobnych plikach, aby nie psuć wersji Grid.

1. G9. Skopiuj `index.html` jako `galeria-flex.html` i zmień w nim tylko odnośnik do arkusza na `style-flex.css`. W nowym pliku `style-flex.css` w pierwszej linii zaimportuj główny arkusz: `@import url("style.css");`, a poniżej nadpisz galerię: kontener flex z zawijaniem i odstępem 12px, figury `flex: 1 1 180px`, pierwsza figura z podstawą 372px.
2. G10. Porównaj obie wersje przy szerokościach 1280px i 1000px (zrzuty obok siebie). Zwróć uwagę na ostatni rząd i na wyróżnioną ilustrację. Zapisz w sprawozdaniu, kiedy lepszy jest Grid, a kiedy Flexbox.
3. G11. Sprawdź oba arkusze walidatorem CSS W3C. Zrób zrzuty końcowe strony przy 1280px.

W sprawozdaniu umieść

- ☐ tabela G1–G11 z regułami CSS i opisem wyniku
- ☐ zrzut nakładki Grid z nazwami obszarów (G4)
- ☐ liczba kolumn galerii przy 1280px, 1000px i 800px (G5)

- ☐ opis różnicy auto-fit i auto-fill (G6)
- ☐ rzuty galerii Grid i Flexbox obok siebie z wnioskami (G10)
- ☐ folder `zad14` z plikami `index.html`, `style.css`, `galeria-flex.html`, `style-flex.css`
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Co oznacza jednostka `fr` i czym różni się kolumna `1fr` od kolumny `100%`?
2. Kiedy `auto-fit` i `auto-fill` dają ten sam wynik, a kiedy różny?
3. Dlaczego Grid nazywamy układem dwuwymiarowym, a Flexbox jednowymiarowym? Podaj przykład zadania dla każdego.
4. Jak zmienić układ z obszarami, aby panel boczny był po lewej stronie? Ile linii CSS trzeba zmienić?

Kryteria oceny

Kryterium	Punkty
Obszary nazwane i rozmieszczone wg szkicu, kolumny <code>1fr</code> 280px, stopka na dole okna, menu w poziomie (G1–G3)	5
Zrzut nakładki Grid i odczyt szerokości (G4)	1
Galeria <code>auto-fit</code> / <code>minmax</code> z odstępem 12px, liczby kolumn, opis <code>auto-fit</code> vs <code>auto-fill</code> (G5–G6)	4
Pierwsza ilustracja na 2×2 komórki, analiza dziur (G7–G8)	2
Wersja Flexbox w osobnych plikach i porównanie z wnioskami (G9–G10)	4
Walidacja CSS bez błędów, kompletne sprawozdanie	1
Odpowiedzi na pytania kontrolne	3
Razem	20

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik `.docx` o tym samym numerze).