

CSS — selektory i kaskada

Czas: 2 × 45 min

praca indywidualna

Cel: Dołączenie zewnętrznego arkusza CSS i stosowanie różnych rodzajów selektorów oraz zrozumienie kaskady i specyficzności na podstawie przewidywań sprawdzanych w narzędziach F12.

Po zajęciach potrafisz:

- dołącza zewnętrzny arkusz stylów elementem link
- stosuje selektory elementu, klasy, id, potomka, dziecka, sąsiada, atrybutu i pseudoklasy
- oblicza specyficzność selektora i przewiduje, która reguła wygra
- wyjaśnia działanie kolejności reguł, stylu inline i !important
- odczytuje w F12 → Styles zastosowane i przekreślone deklaracje

Sprzęt i materiały

- komputer z Windows 11
- folder `r1_09_start` z Pliki_do_zadan (możesz też użyć własnej strony z zadania 2, ale wtedy dopisz brakujące elementy: tabelę, klasy `uwaga`, `wazne`, `wyrozniione`, odnośnik zewnętrzny)

Oprogramowanie

- **Visual Studio Code** — rozszerzenie Live Server (autor: Ritwick Dey) — instalacja z panelu Rozszerzenia (Ctrl+Shift+X) code.visualstudio.com
- **Przeglądarka Chrome, Edge lub Firefox** — narzędzia deweloperskie — klawisz F12
- **Walidator CSS W3C** — sprawdzanie pliku style.css jigsaw.w3.org/css-validator

Przygotowanie przed zajęciami

- Powtórz z wykładu 1.07 sposób liczenia specyficzności (id, klasa/atribut/pseudoklasa, element).

Przebieg zadania

1 Przewidź kolor — specyficzność

Specyficzność zapisujemy jako trójkę (id, klasy, elementy), np. `div .info` = (0,1,1). Przy remisie wygrywa reguła zapisana później. Styl inline i `!important` mają pierwszeństwo.

1. Otwórz `specyficznosc.html` w VS Code (nie w przeglądarce!). Dla każdego akapitu P1–P8 wypisz pasujące reguły z ich specyficznością i przewidź kolor. Wpisz do tabeli w sprawozdaniu.
2. Dopiero teraz otwórz plik w przeglądarce, sprawdź kolory i dla każdego akapitu zajrzyj do F12 → Elements → Styles: która deklaracja wygrała, które są przekreślone? Zaznacz w tabeli trafione przewidywania.
3. Dla P6 i P7 wyjaśnij, co zdecydowało o kolorze. Dla P8 wyjaśnij, dlaczego nie działa reguła z `:nth-child(2)`.

2 Zewnętrzny arkusz stylów

Każdą regułę poprzedź komentarzem z numerem, np. `/* S5. Selektor potomka */`.

```
zad09/  
├─ index.html  
├─ specyficzosc.html  
└─ css/  
    └─ style.css
```

1. Skopiuj pliki startowe do `zad09`, utwórz `css/style.css` i dołącz go w `head` strony `index.html` (zastąp komentarz). Wpisz w stopce swoje dane.
2. S1. `body`: krój bezszeryfowy z listą zapasowych krojów, kolor tekstu `#222222`, tło `#f4f6fb`.
3. S2. Element o `id="naglowek"`: tło `#14213d`, biały tekst, `padding: 16px`.
4. S3. Jedną regułą (grupowanie) nadaj `h2` i `h3` kolor `#1d4e89`.
5. S4. Tylko akapit będący bezpośrednim dzieckiem `header`: kursywa. S5. Odnośniki wewnątrz `nav`: pogrubione, bez podkreślenia. S6. Po najechaniu kursorem: kolor `#e76f51` i podkreślenie.
6. S7. Klasa `uwaga`: kolor `#9b2226`, pogrubienie. S8. Tylko `section` z klasą `wazne`: lewa ramka 6px w kolorze `#fca311` i `padding-left: 10px`.
7. S9. Co drugi wiersz `tbody` (parzyste) — tło `#dde7f5`; dodaj też krawędzie komórek tabeli. S10. Odnośniki, których `href` zaczyna się od `https` — kolor `#2a9d8f`.
8. S11. Akapit leżący bezpośrednio po `h3` (data wpisu): rozmiar 0.9em, kolor `#555555`. S12. Pierwszy odnośnik menu (`:first-child`): wielkie litery (`text-transform`). S13. `article` z klasą `wyrozniome`: tło `#fff3c4`, `padding: 8px`.

3 Sprawdzenie w F12 i walidacja

1. Zbadaj (F12) podtytuł strony i datę dowolnego wpisu — zapisz, które reguły z Twojego arkusza na nie działają i jakie właściwości są dziedziczone z `body` (sekcja Inherited from).
2. W zakładce Styles wyłącz chwilowo (checkbox) regułę S9 i włącz ją z powrotem. Zmień w F12 kolor z S7 i sprawdź, czy zmiana zapisła się w pliku.
3. Sprawdź `style.css` walidatorem CSS W3C, a `index.html` walidatorem HTML. Popraw błędy i zrób zrzut strony.

W sprawozdaniu umieść

- ☐ tabela przewidywań P1–P8 (reguły, specyficzność, przewidziany i rzeczywisty kolor)
- ☐ wyjaśnienie wyników P6, P7 i P8
- ☐ plik `css/style.css` z regułami S1–S13 i zrzut strony
- ☐ zrzut F12 → Styles z przekreśloną deklaracją i informacją o dziedziczeniu
- ☐ wyniki walidacji HTML i CSS

Pytania kontrolne

1. Oblicz specyficzność selektorów: `#menu a`, `.ramka p:nth-child(2)`, `ul li a:hover`, `a[href^="https"]`. Który jest najsilniejszy?
2. Dlaczego nie należy nadużywać `!important` i stylów inline?
3. Czym różni się selektor `header p` od `header > p`?
4. Które właściwości dziedziczą się z rodzica (podaj dwie), a które nie (podaj dwie)?

Kryteria oceny

Kryterium	Punkty
Tabela przewidywań P1–P8 z poprawnie obliczoną specyficznością	4
Wyjaśnienie P6 (!important), P7 (inline) i P8 (nth-child liczy wszystkie dzieci rodzica, a p8 nie jest w .ramka)	2
Arkusz zewnętrzny dołączony, reguły S1–S7 poprawne	4
Reguły S8–S13: klasa z elementem, nth-child, atrybut, sąsiad, first-child	4
Praca z F12 (przekreślenia, dziedziczenie, wyłączanie reguł) i walidacja HTML/CSS bez błędów	3
Odpowiedzi na pytania kontrolne	3
Razem	20

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