

Zestaw Zadań: Weryfikacja i utrwalenie wiedzy o CSS

Informacje Ogólne

Materiał źródłowy: Kompleksowy przegląd CSS - od podstaw (sposoby dodawania stylów, selektory, kaskadowość) przez typografię, Box Model, pozycjonowanie, animacje, aż po Flexbox, Grid i responsywność

Poziom: Mieszany (podstawowy → zaawansowany)

Szacowany czas: 5-6 godzin (wszystkie zadania)

Typ zestawu: Powiązane - każde zadanie rozwija struktury HTML stworzone w poprzednim zestawie

Wymagania wstępne: Ukończone zadania z zestawu HTML lub równoważna wiedza

Zadanie 1: Trzy sposoby na CSS

Poziom trudności: 1 

Szacowany czas: 25 minut

Punktacja: 15 punktów

Typ: Teoretyczne/Praktyczne

Powiązanie: Rozszerza Zadanie 1 z HTML (Anatomia dokumentu HTML)

Cel dydaktyczny

Zrozumienie trzech sposobów dodawania CSS do dokumentu HTML, poznanie zasad kaskadowości i priorytetów stylów.

Polecenie

Wykorzystaj szablon dokumentu HTML5 stworzony w Zadaniu 1 (HTML) i wykonaj następujące czynności:

- Dodaj style inline do elementów w <body>:**
 - Ustaw kolor tekstu na niebieski dla jednego elementu
 - Zmień rozmiar czcionki na 20px dla innego elementu
- Dodaj sekcję <style> w <head> zawierającą:**
 - Styl dla body (font-family, background-color)
 - Minimum 3 różne selektory (element, klasa, id)
- Stwórz zewnętrzny plik style.css zawierający:**
 - Te same style co w <style>, ale z innymi wartościami
 - Podłącz plik do HTML za pomocą <link>
- Napisz komentarz (150-200 słów) wyjaśniający:**
 - Kolejność priorytetów (który styl wygrywa i dlaczego?)
 - Kiedy stosować każdy sposób?
 - Zalety i wady każdej metody

Materiały/Zasoby potrzebne

- Szablon HTML z Zadania 1 (HTML)
- Edytor tekstu lub IDE
- Przeglądarka z DevTools

Kryteria oceny

- Poprawne użycie stylów inline (3 pkt)
- Poprawna sekcja <style> z różnymi selektorami (4 pkt)
- Poprawny zewnętrzny plik CSS i jego podłączenie (4 pkt)
- Jakość wyjaśnienia teoretycznego (4 pkt)

Wskazówki

- Użyj DevTools (F12) → Zakładka Elements, aby zobaczyć które style są nadpisane
- Priorytet: inline > internal > external (przy tej samej specyficzności)
- Pamiętaj o komentarzach CSS: /* komentarz */

Zadanie 2: Typografia i stylizacja tekstu

Poziom trudności: 2 🟡

Szacowany czas: 45 minut

Punktacja: 20 punktów

Typ: Praktyczne

Powiązanie: Rozszerza Zadanie 2 z HTML (Mistrz formatowania tekstu)

Cel dydaktyczny

Opanowanie właściwości CSS związanych z typografią: fonty, kolory, wyrównanie, dekoracje tekstu i efekty wizualne.

Polecenie

Weź artykuł HTML stworzony w Zadaniu 2 (HTML) i dodaj do niego zewnętrzny arkusz stylów zawierający:

Część A - Fonty (7 pkt)

1. Zaimportuj i zastosuj font Google Fonts dla nagłówków
2. Zaimportuj i zastosuj inny font Google Fonts dla tekstu głównego
3. Ustaw font stack (font-family) z fallback fonts
4. Zastosuj różne font-weight dla nagłówków (h1, h2, h3)
5. Użyj font-variant: small-caps dla jednego elementu

Część B - Kolory (5 pkt)

6. Użyj minimum 3 różnych formatów kolorów: HEX, RGB/RGBA, HSL/HSLA
7. Stwórz spójną paletę kolorystyczną (max 4-5 kolorów)
8. Zastosuj przezroczystość (opacity lub rgba/hsla) dla co najmniej jednego elementu

Część C - Właściwości tekstu (8 pkt)

9. Różne text-align dla różnych sekcji (left, center, right, justify)
10. text-decoration dla linków (hover i normalne)
11. text-transform dla nagłówków (uppercase, capitalize)

12. line-height dla czytelności paragrafów (1.5-1.8)
13. letter-spacing i word-spacing dla nagłówka głównego
14. text-shadow dla tytułu artykułu
15. text-indent dla pierwszego paragrafu każdej sekcji

Materiały/Zasoby potrzebne

- Artykuł HTML z Zadania 2
- Google Fonts: fonts.google.com
- Narzędzie do doboru kolorów (np. coolors.co)

Kryteria oceny

- Poprawny import i użycie Google Fonts (4 pkt)
- Font stack z fallback fonts (3 pkt)
- Różnorodność formatów kolorów (3 pkt)
- Spójna paleta kolorystyczna (2 pkt)
- Użycie wszystkich wymaganych właściwości tekstu (6 pkt)
- Estetyka i czytelność (2 pkt)

Wskazówki

- Łącz fonty kontrastowe: serif dla nagłówków + sans-serif dla tekstu (lub odwrotnie)
 - HSL jest najłatwiejszy do tworzenia wariantów kolorów (zmień tylko L dla jasności)
 - Unikaj text-justify dla krótkich linii - powoduje brzydkie przerwy
-

Zadanie 3: Box Model w praktyce

Poziom trudności: 2 🟡

Szacowany czas: 50 minut

Punktacja: 25 punktów

Typ: Praktyczne

Powiązanie: Rozszerza Zadanie 3 z HTML (Listy i tabele w praktyce)

Cel dydaktyczny

Opanowanie Box Model - margin, padding, border, width/height, oraz różnic między box-sizing: content-box a border-box.

Polecenie

Weź stronę z listami i tabelą z Zadania 3 (HTML) i dodaj kompleksową stylizację:

Część A - Stylizacja list (10 pkt)

1. **Lista nieuporządkowana:**
 - Własny marker (list-style-type lub list-style-image)
 - Padding i margin dla elementów
 - Tło dla co drugiego elementu (nth-child)
2. **Lista uporządkowana:**
 - Własny styl numeracji (list-style-type: upper-roman, lower-alpha, etc.)
 - Border-left dla całej listy
3. **Lista definicji:**
 - Wyróżnienie <dt> (pogrubienie, większy font, tło)
 - Wcięcie dla <dd>
 - Separator między definicjami (border-bottom)

Część B - Stylizacja tabeli (10 pkt)

4. Struktura tabeli:
 - border-collapse: collapse
 - Szerokość 100% z max-width
 - box-shadow dla całej tabeli
5. Nagłówki (thead/th):
 - Wyróżniający kolor tła
 - Padding minimum 12px
 - border-bottom wyrazisty
6. Komórki (td):
 - Padding dla czytelności
 - Alternujące tło wierszy (zebra stripes)
 - Efekt hover na wierszu

Część C - Box Model Demo (5 pkt)

Stwórz sekcję demonstracyjną pokazującą różnicę między box-sizing: content-box a border-box:

- Dwa identyczne divy obok siebie
- Oba z width: 300px, padding: 20px, border: 5px solid
- Jeden z content-box, drugi z border-box
- Podpisy wyjaśniające różnicę w rzeczywistej szerokości

Materiały/Zasoby potrzebne

- Strona HTML z Zadania 3
- DevTools do inspekcji Box Model

Kryteria oceny

- Stylizacja list (wszystkie 3 typy) (6 pkt)
- Użycie nth-child dla list (2 pkt)
- Kompletna stylizacja tabeli (6 pkt)
- Zebra stripes + hover (2 pkt)
- Box-shadow (2 pkt)
- Demonstracja box-sizing (5 pkt)
- Ogólna estetyka i spójność (2 pkt)

Wskazówki

- Użyj * { box-sizing: border-box; } jako reset na początku CSS
- tr:nth-child(even) dla zebra stripes
- Delikatne cienie: box-shadow: 0 2px 5px rgba(0,0,0,0.1)

Zadanie 4: Tła i pozycjonowanie - Galeria

Poziom trudności: 3 ●

Szacowany czas: 60 minut

Punktacja: 25 punktów

Typ: Praktyczne

Powiązanie: Rozszerza Zadanie 4 z HTML (Galeria multimedialna)

Cel dydaktyczny

Opanowanie właściwości background, różnych schematów pozycjonowania (relative, absolute, fixed, sticky) oraz tworzenia overlayów i efektów wizualnych.

Polecenie

Weź galerię multimedialną z Zadania 4 (HTML) i przekształć ją w atrakcyjną wizualnie stronę:

Część A - Tła i gradienty (8 pkt)

1. Tło strony:
 - Gradient liniowy lub radialny jako background
 - Użyj background-attachment: fixed dla efektu parallax
2. Nagłówek strony:
 - Obrazek jako tło (background-image)
 - background-size: cover
 - Overlay z gradientem dla czytelności tekstu
3. Karty galerii:
 - Białe/jasne tło z border-radius
 - Box-shadow dla efektu uniesienia

Część B - Pozycjonowanie (12 pkt)

4. **Position: relative + absolute:**

- Badge/etykieta w rogu każdego obrazka (np. 'NEW', kategoria)
- Overlay z opisem pojawiający się na hover

5. **Position: fixed:**

- Przycisk 'Powrót do góry' w prawym dolnym rogu
- Pasek social media na boku strony

6. **Position: sticky:**

- Nagłówek sekcji (np. 'Obrazy', 'Wideo') przyklejający się przy scrollowaniu

Część C - Z-index i warstwy (5 pkt)

- Stwórz efekt lightbox: kliknięcie w obraz pokazuje overlay z powiększonym obrazem
- Overlay musi być nad wszystkimi elementami (z-index)
- Przycisk zamknięcia w rogu overlay

(Uwaga: sam overlay może być stworzony w HTML + CSS bez JavaScript, z użyciem :target)

Materiały/Zasoby potrzebne

- Galeria HTML z Zadania 4
- Obrazy stockowe (unsplash.com, pexels.com)
- CSS Gradient Generator (cssgradient.io)

Kryteria oceny

- Gradienty i tła (4 pkt)
- Background z obrazem i overlay (4 pkt)
- Position relative/absolute - badge i hover overlay (5 pkt)
- Position fixed - elementy stałe (3 pkt)
- Position sticky (2 pkt)
- Lightbox z z-index (5 pkt)
- Estetyka i spójność (2 pkt)

Wskazówki

- Dla overlay: position: absolute; top: 0; left: 0; right: 0; bottom: 0;
 - Lightbox CSS-only: użyj i #img1:target { display: block; }
 - Pamiętaj: sticky wymaga określenia top/bottom i rodzica bez overflow: hidden
-

Zadanie 5: Stylizacja formularzy i pseudo-klasy

Poziom trudności: 3 

Szacowany czas: 60 minut

Punktacja: 25 punktów

Typ: Praktyczne

Powiązanie: Rozszerza Zadanie 5 z HTML (Kompleksowy formularz kontaktowy)

Cel dydaktyczny

Opanowanie stylizacji formularzy, pseudo-klas (:hover, :focus, :valid, :invalid, :checked, etc.) oraz pseudo-elementów (::before, ::after, ::placeholder).

Polecenie

Weź formularz z Zadania 5 (HTML) i stwórz profesjonalną stylizację:

Część A - Podstawowa stylizacja formularza (8 pkt)

1. Pola tekstowe (input, textarea):
 - Spójny styl: border, border-radius, padding
 - Stylizacja ::placeholder
 - Szerokość 100% z max-width
2. Etykiety (label):
 - Wyraźny font-weight
 - Odpowiednie odstępy (margin-bottom)
3. Przyciski:
 - Wyrazisty kolor tła
 - cursor: pointer
 - Padding dla klikalności
4. Fieldset i legend:
 - Stylowe obramowanie fieldset
 - Legend jako elegancki nagłówek sekcji

Część B - Pseudo-klasy stanów (10 pkt)

5. :hover - dla wszystkich interaktywnych elementów
6. :focus - wyraźne oznaczenie aktywnego pola (outline lub box-shadow)
7. :focus-within - podświetlenie całej grupy pól
8. :valid i :invalid - wizualna walidacja (zielona/czerwona ramka)
9. :checked - stylizacja wybranych checkboxów i radio
10. :disabled - wyszarzone pola nieaktywne (dodaj atrybut disabled do jednego pola)
11. :required - oznaczenie pól wymaganych (np. gwiazdka przez ::after)

Część C - Pseudo-elementy (5 pkt)

12. ::before dla gwiazdki przy polach required
13. ::after dla ikony walidacji (✓ lub ✗)
14. ::first-letter dla pierwszej litery legend
15. Własny wygląd checkboxa/radio (ukrycie domyślnego, stworzenie custom)

Część D - Bonus: Custom range slider (2 pkt)

Ostyluj input type='range' z użyciem pseudo-elementów ::-webkit-slider-thumb i ::-webkit-slider-runnable-track.

Materiały/Zasoby potrzebne

- Formularz HTML z Zadania 5
- Dokumentacja MDN - CSS Pseudo-classes

Kryteria oceny

- Podstawowa stylizacja pól i przycisków (5 pkt)
- Stylizacja fieldset/legend (3 pkt)
- Pseudo-klasy stanów - wszystkie wymagane (7 pkt)
- :valid/:invalid walidacja wizualna (3 pkt)
- Pseudo-elementy (4 pkt)
- Estetyka i UX (3 pkt)

Wskazówki

- Nie usuwaj domyślnego outline bez zastąpienia go innym wskaźnikiem focus (accessibility!)
- Custom checkbox: ukryj domyślny (opacity: 0) i styluj label::before
- :valid/:invalid działają dopiero po interakcji - użyj :not(:placeholder-shown) dla lepszego UX

Zadanie 6: Animacje i transformacje

Poziom trudności: 3 

Szacowany czas: 50 minut

Punktacja: 20 punktów

Typ: Praktyczne/Twórcze

Powiązanie: Może być dodane do dowolnego z poprzednich zadań

Cel dydaktyczny

Opanowanie CSS transitions, transforms (2D i 3D) oraz keyframe animations do tworzenia płynnych efektów wizualnych.

Polecenie

Stwórz stronę demonstracyjną 'CSS Animation Showcase' prezentującą różne efekty:

Część A - Transitions (6 pkt)

Stwórz sekcję z przyciskami/kartami demonstrującą:

1. Płynna zmiana koloru tła na hover
2. Płynne powiększenie (transform: scale)
3. Różne timing functions: ease, ease-in-out, cubic-bezier
4. Transition delay dla efektu kaskadowego

Część B - Transforms 2D (6 pkt)

Stwórz elementy demonstrujące:

5. translate() - przesunięcie elementu
6. rotate() - obrót
7. scale() - skalowanie
8. skew() - pochylenie
9. Kombinacja transformacji w jednym elemencie

Część C - Keyframe Animations (8 pkt)

Stwórz minimum 3 różne animacje:

10. Pulsujący element (zmiana skali i/lub opacity)
11. Obracający się loader/spinner
12. Animacja wejścia tekstu (slideDown, fadeIn, lub własna)
13. Użyj animation-iteration-count: infinite dla co najmniej jednej
14. Użyj animation-direction: alternate dla jednej animacji
15. Animacja wieloetapowa (więcej niż 0% i 100%)

Bonus - Transform 3D (3 pkt)

Stwórz obracającą się kartę (card flip) z użyciem:

- perspective
- rotateY() lub rotateX()
- backface-visibility: hidden

Materiały/Zasoby potrzebne

- Edytor HTML/CSS
- cubic-bezier.com - generator krzywych animacji
- animista.net - inspiracje animacji

Kryteria oceny

- Transitions z różnymi timing functions (4 pkt)
- Transform 2D - wszystkie typy (4 pkt)
- Kombinacja transformacji (2 pkt)
- Keyframe animations - różnorodność (6 pkt)
- Właściwe użycie animation properties (2 pkt)
- Estetyka i płynność animacji (2 pkt)

Wskazówki

- Subtelne animacje są lepsze - unikaj ekstremalnych wartości
 - transform-origin zmienia punkt odniesienia transformacji
 - Dla wydajności preferuj transform i opacity nad innymi właściwościami
 - Testuj prefers-reduced-motion dla dostępności
-

Zadanie 7: Projekt kompleksowy - Responsywna strona

Poziom trudności: 4 ●

Szacowany czas: 150 minut

Punktacja: 50 punktów

Typ: Syntetyczne/Twórcze

Powiązanie: Stylizacja kompletnej strony z Zadania 7 HTML

Cel dydaktyczny

Połączenie wszystkich elementów CSS w kompleksowy projekt wykorzystujący Flexbox, Grid, CSS Variables i Media Queries do stworzenia w pełni responsywnej strony.

Polecenie

Weź wielostronicową witrynę z Zadania 7 (HTML) i stwórz kompletny system stylów CSS:

Część A - CSS Variables i architektura (8 pkt)

1. **Zdefiniuj zmienne CSS w :root:**
 - Kolory główne (primary, secondary, accent)
 - Kolory tekstu i tła
 - Fonty (--font-heading, --font-body)
 - Rozmiary (--spacing-sm, --spacing-md, --spacing-lg)
 - Border-radius, shadows
2. **Organizacja CSS:**
 - Reset/Normalize
 - Zmienne
 - Typografia
 - Komponenty
 - Layout
 - Media queries

Część B - Layout z Flexbox (10 pkt)

3. Nawigacja (Flexbox):
 - Logo po lewej, menu po prawej (justify-content: space-between)
 - Elementy menu w linii (display: flex)
 - Gap między elementami
4. Karty/Sekcje (Flexbox):
 - flex-wrap: wrap dla responsywności
 - flex-grow/flex-shrink dla elastyczności
5. Footer (Flexbox):
 - Wielokolumnowy layout
 - align-items dla wyrównania pionowego

Część C - Layout z Grid (10 pkt)

6. Główny layout strony:
 - grid-template-areas dla czytelnej struktury
 - Header, main, sidebar, footer
7. Galeria/Portfolio:
 - grid-template-columns: repeat(auto-fit, minmax(250px, 1fr))
 - Gap dla odstępów

8. Span dla wyróżnionych elementów:
- grid-column: span 2 lub grid-row: span 2

Część D - Responsywność (12 pkt)

9. **Minimum 3 breakpointy:**
- Mobile: < 576px
 - Tablet: 576px - 992px
 - Desktop: > 992px
10. **Elementy responsywne:**
- Nawigacja: hamburger menu na mobile
 - Grid: zmiana liczby kolumn
 - Sidebar: ukryty lub na dole na mobile
 - Typografia responsywna (clamp()) lub media queries
 - Obrazy: max-width: 100%
11. **Mobile-first approach:**
- Bazowe style dla mobile
 - min-width media queries dla większych ekranów

Część E - Dodatkowe wymagania (10 pkt)

12. Dark mode (użyj prefers-color-scheme lub toggle)
13. Animacje/transitions dla interakcji
14. Print styles (@media print)
15. Spójność na wszystkich podstronach
16. Komentarze organizacyjne w CSS

Materiały/Zasoby potrzebne

- Strona HTML z Zadania 7
- DevTools z widokiem responsywnym
- CSS Grid Generator (cssgrid-generator.netlify.app)
- Flexbox Froggy / Grid Garden - dla praktyki

Kryteria oceny

- CSS Variables i organizacja (6 pkt)
- Flexbox - nawigacja i komponenty (8 pkt)
- Grid - główny layout i galeria (8 pkt)
- Responsywność - wszystkie breakpointy (10 pkt)
- Mobile-first approach (4 pkt)
- Dark mode (4 pkt)
- Print styles (2 pkt)
- Spójność i estetyka (5 pkt)
- Komentarze i czytelność kodu (3 pkt)

Wskazówki

- Zaczynij od mobile-first: najpierw ostyluj wersję mobilną
 - Testuj na prawdziwych urządzeniach lub w DevTools
 - Używaj clamp() dla responsywnej typografii: clamp(1rem, 2vw + 1rem, 2rem)
 - Dark mode: zmień tylko wartości zmiennych CSS
 - Hamburger menu CSS-only: użyj checkbox hack
-

Dodatkowe Informacje

Sugerowana kolejność wykonywania

Zadania są zaprojektowane sekwencyjnie - każde następne buduje na poprzednich:

- Zadanie 1 → Zadanie 2 → Zadanie 3 → ... → Zadanie 7
- Zadanie 6 (Animacje) może być wykonane niezależnie lub dodane do dowolnego wcześniejszego zadania

Opcjonalne zadania rozszerzające

- Stwórz system Design Tokens eksportowany do JSON
- Zaimplementuj Container Queries zamiast Media Queries
- Stwórz własny CSS Framework/Library na bazie projektu
- Dodaj @layer dla lepszej organizacji kaskady
- Zaimplementuj CSS Scroll Snap dla galerii

Pomocne zasoby

Dokumentacja i referencje

MDN CSS Reference: developer.mozilla.org/en-US/docs/Web/CSS

CSS-Tricks: css-tricks.com

W3Schools CSS: w3schools.com/css

Narzędzia do nauki

Flexbox Froggy: flexboxfroggy.com

Grid Garden: cssgridgarden.com

CSS Diner (selektory): flukeout.github.io

Generatory i narzędzia

Google Fonts: fonts.google.com

CSS Gradient: cssgradient.io

Coolors (palety kolorów): coolors.co

Box Shadow Generator: cssmatic.com/box-shadow

Animista: animista.net

Can I Use: caniuse.com

Walidacja

W3C CSS Validator: jigsaw.w3.org/css-validator