

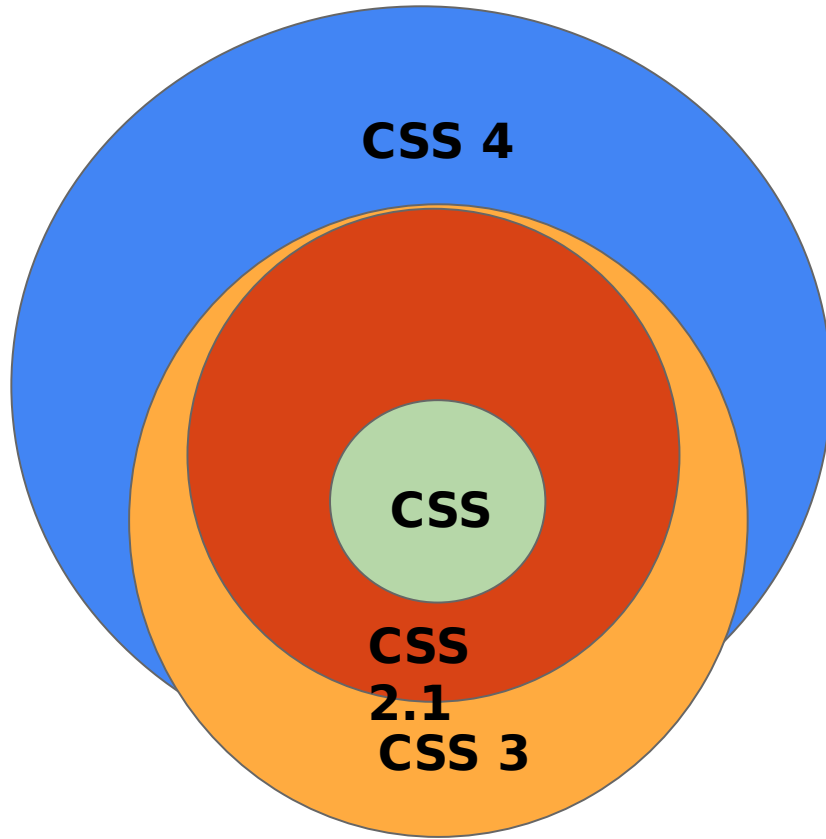
CSS - Kaskadowe arkusze stylów przekrój funkcjonalności



Kamil Cecherz

Wprowadzenie

- **CSS1 (1996) – ok. 50 właściwości**
 - Czcionki (font), Kolor i tło, Tekst, Listy, Marginesy (margin), Odstępy (padding), Obramowania (border), Wymiary (box)
- **CSS2.1 (2004) – ok. 120-130 właściwości**
 - Model pudełkowy (box model), Pozycjonowanie (positioning), Selektory, Media types, Pseudo-klasy i pseudoelementy, Jednostki względne i absolutne, Wyświetlanie (display), Tabele, Wartości dziedziczone (inheritance), Druk i paginacja, Wartości domyślne
- **CSS3 i moduły współczesne (2025) – około 560-600 właściwości zdefiniowanych w specyfikacjach W3C**
 - Tło i gradienty, Typografia, Układ / Layout (flexbox, grid), Animacje i przejścia, Kolory i przezroczystość, Media Queries, Pseudoelementy i generowanie treści, Filtry i efekty wizualne, Transformacje 2D i 3D



Inaczej niż **CSS 2**, który jest jedną wielką specyfikacją definiującą różne właściwości, **CSS 3** podzielony jest na niezależne dokumenty zwane modułami. Każdy moduł zawiera nowe możliwości i rozszerza elementy zdefiniowane w **CSS 2**, tak, żeby zachowywały kompatybilność z wersjami wcześniejszymi.

CSS 4 nie istnieje już jako jedna specyfikacja. Odkąd **CSS 3** został podzielony na moduły, każdy z nich rozwija się niezależnie. Większość z nich jest na **poziomie 3**, który bazuje na wcześniejszym **CSS 2.1**. Kilka z nich osiągnęło **poziom 4** np. własności grafiki (*image values*), tła i obramowania (*backgrounds & borders*)¹ czy selektory¹, które oparto na sprawdzonych modułach **poziomu 3**.

Język CSS w pigułce

Aspekt	Opis	
Selektory	Określają, które elementy HTML mają być stylowane. Typowe: <code>.klasa</code> , <code>#id</code> , <code>div > p</code> , <code>a:hover</code> , <code>:not()</code> , <code>:has()</code> .	
Dziedziczenie i kaskadowość	Reguły stylu mogą się dziedziczyć; w razie konfliktu decyduje priorytet (specificity) i kolejność.	
Jednostki	<code>px</code> , <code>%</code> , <code>em</code> , <code>rem</code> , <code>vw</code> , <code>vh</code> , <code>fr</code> , <code>ch</code> , <code>lh</code> , <code>min()</code> , <code>max()</code> , <code>clamp()</code> . Umożliwiają tworzenie responsywnych układów.	
Wymiary (Dimensions)	<code>width</code> , <code>height</code> , <code>min/max-width</code> , <code>min/max-height</code> . Określają rozmiar elementów.	
Marginesy (Margin)	Odstęp na zewnątrz elementu: <code>margin</code> , <code>margin-top/right/bottom/left</code> , <code>auto</code> do centrowania.	
Wypełnienie (Padding)	Odstęp między zawartością a krawędzią elementu.	
Obramowanie (Border)	<code>border-width</code> , <code>border-style</code> , <code>border-color</code> , <code>border-radius</code> , <code>border-image</code> .	

Model pudełkowy (Box Model)	Każdy element to prostokąt składający się z: content + padding + border + margin. Kontrolowany przez <code>box-sizing</code> .
Pozycjonowanie (Positioning)	<code>position: static, relative, absolute, fixed, sticky</code> . Ustawia elementy względem innych lub okna.
Układ (Layout)	<code>display (block , inline , flex , grid , table , none)</code> .
Flexbox	Jednowymiarowy układ. Kontrola osi i wyrównania: <code>flex-direction , justify-content , align-items , flex-grow</code> .
Grid Layout	Dwuwymiarowa siatka: <code>grid-template-columns/rows , grid-gap , grid-area , justify-items</code> .
Float / Clear	Starszy sposób układu w poziomie, głównie historyczny.
Overflow	Jak zachować się, gdy zawartość wychodzi poza kontener (<code>visible , hidden , scroll , auto</code>).
Z-index	Kolejność nakładania się elementów na siebie.

Kolory i tła	<code>color</code> , <code>background-color</code> , <code>background-image</code> , <code>background-repeat</code> , <code>background-size</code> , <code>background-position</code> .
Gradienty	<code>linear-gradient()</code> , <code>radial-gradient()</code> , <code>conic-gradient()</code> – generują płynne przejścia kolorów.
Cienie (Shadows)	<code>box-shadow</code> (dla elementów), <code>text-shadow</code> (dla tekstu).
Nieprzezroczystość (Opacity)	<code>opacity: 0-1</code> – przezroczystość całego elementu.
Tekst i typografia	<code>font-family</code> , <code>font-size</code> , <code>font-weight</code> , <code>line-height</code> , <code>text-align</code> , <code>text-decoration</code> , <code>letter-spacing</code> .
@font-face	Pozwala wczytać niestandardową czcionkę z zewnętrznego źródła.
Zawijanie tekstu	<code>white-space</code> , <code>word-wrap</code> , <code>overflow-wrap</code> , <code>hyphens</code> .
Dekoracje tekstu	<code>text-shadow</code> , <code>text-transform</code> , <code>text-underline-offset</code> , <code>text-emphasis</code> .

Tabele	<code>border-collapse</code> , <code>caption-side</code> , <code>table-layout</code> .
Kursor i wskaźnik	<code>cursor: pointer</code> , <code>move</code> , <code>wait...</code> — wygląd kursora myszy.
Obrazy i multimedia	<code>object-fit</code> , <code>object-position</code> , <code>clip-path</code> , <code>mask</code> , <code>filter</code> .
Przejścia (Transitions)	Płynne zmiany właściwości: <code>transition-property</code> , <code>transition-duration</code> , <code>transition-timing-function</code> .
Animacje (Animations)	Ruchome efekty z <code>@keyframes</code> . Parametry: <code>animation-name</code> , <code>animation-duration</code> , <code>animation-iteration-count</code> .
Transformacje (Transforms)	Obrót, przesunięcie, skalowanie: <code>transform: rotate()</code> <code>scale()</code> <code>translate()</code> <code>skew()</code> .
Filtry (Filters)	Efekty graficzne: <code>blur()</code> , <code>brightness()</code> , <code>contrast()</code> , <code>grayscale()</code> , <code>sepia()</code> .
Pseudo-klasy	Stan elementu: <code>:hover</code> , <code>:focus</code> , <code>:nth-child()</code> , <code>:not()</code> , <code>:checked</code> .

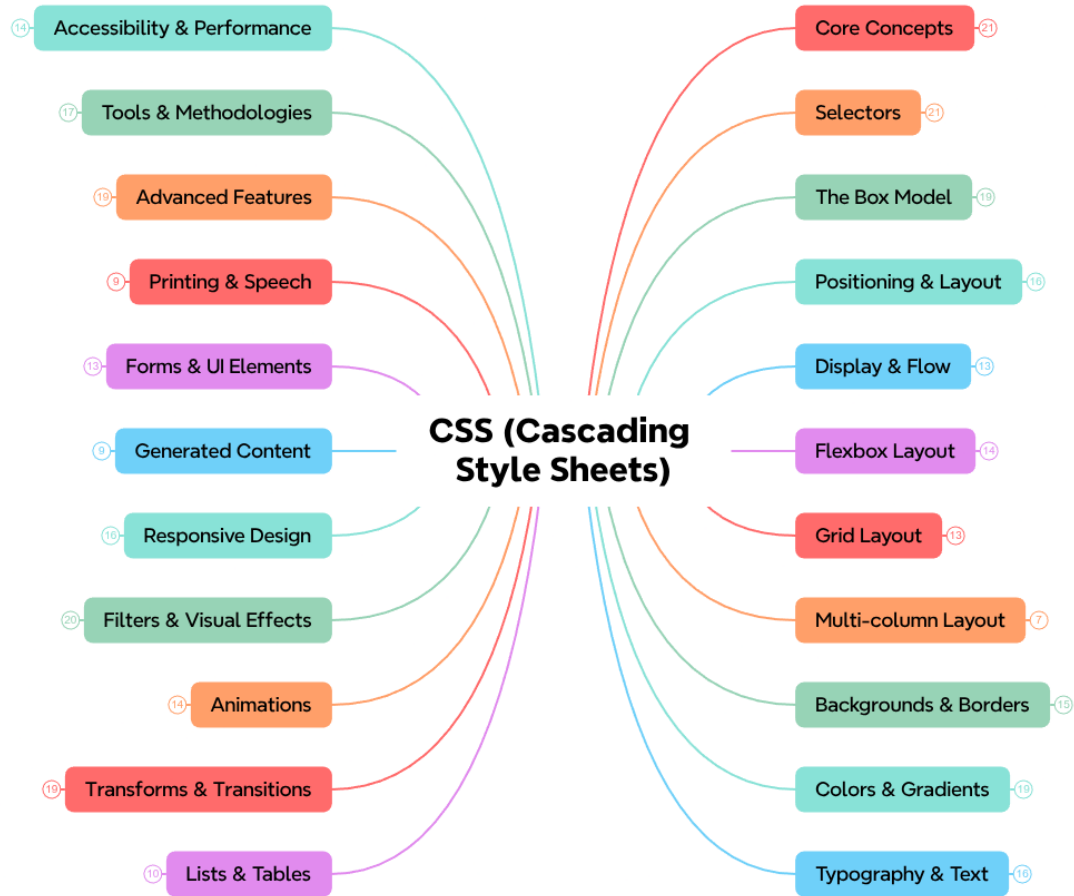
Pseudo-elementy	Wirtualne części: <code>::before</code> , <code>::after</code> , <code>::first-letter</code> , <code>::placeholder</code> .
Media Queries	Reagowanie na rozdzielczość i typ urządzenia: <code>@media (max-width: 600px) { ... }</code> .
Zmienne CSS (Custom Properties)	<code>--nazwa: wartość;</code> używane przez <code>var(--nazwa)</code> . Ułatwiają skalowalność stylów.
Kolorystyka (Color Schemes)	<code>prefers-color-scheme</code> , <code>accent-color</code> , <code>color-scheme: light dark;</code> – wspiera motywy systemowe.
Scroll i przewijanie	<code>scroll-behavior: smooth</code> , <code>scroll-snap-type</code> , <code>scroll-margin</code> .
Widoczność i wyświetlanie	<code>visibility: visible/hidden</code> , <code>display: none</code> , <code>clip-path</code> , <code>backface-visibility</code> .
Zachowanie interfejsu (UI)	<code>resize</code> , <code>pointer-events</code> , <code>user-select</code> , <code>appearance</code> .

Filtry wsteczne (Backdrop filters)

Efekty na tle półprzezroczystych elementów, np. `backdrop-filter: blur(10px)`.

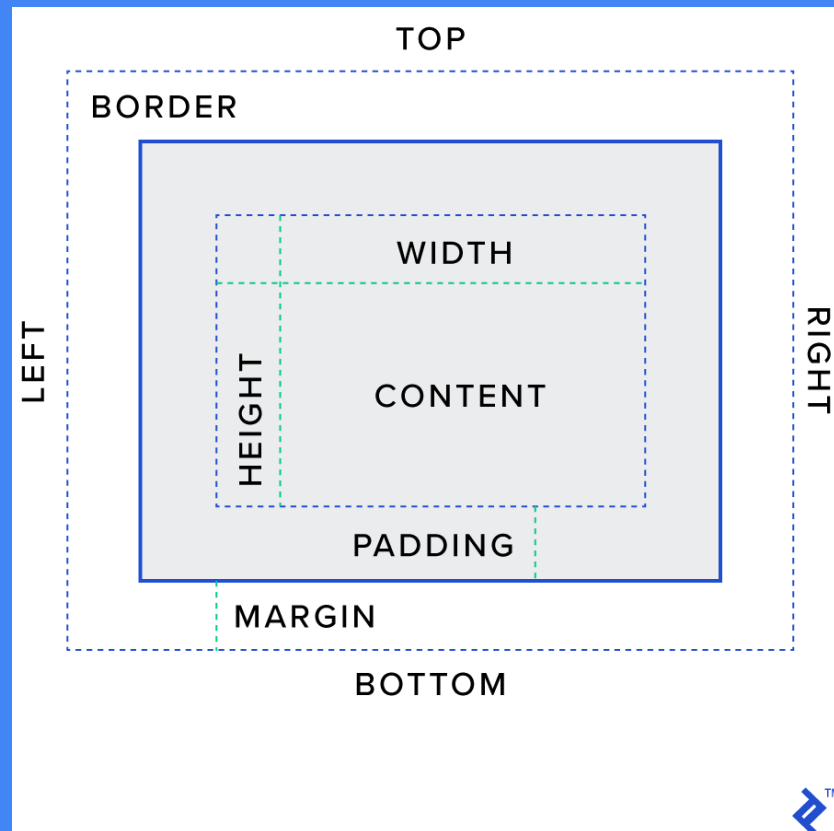
Houdini API

Daje deweloperom dostęp do silnika CSS, by tworzyć własne właściwości i efekty.



Model pudełkowy

Zagnieżdżanie i pozycjonowanie elementów HTML to kluczowy aspekt tworzenia stron WWW.



Przykład 1

```
element_name {  
    height:<value>;  
    width:<value>;  
    border:<value>;  
    padding: <value>;  
    margin:<value>;  
}
```

```
div {  
    height:300px;  
    width:400px;  
    border:1px solid black;  
    padding:30px;  
    margin:20px;  
}
```

Układanie elementów HTML

Struktura drzewiasta dokumentu HTML (dziedziczenie) wpływa na sposób układania elementów względem tagów potomnych i siostrzanych.

- display
- position
- float
- clear
- z-index

● <https://learntheweb.courses/topics/css-layout-cheat-sheet/>

Tworzenie layoutu, struktury strony

- Tworzenie układu strony należy do najważniejszych zadań webdesignera, na szczęście nie trzeba wszystkich właściwości ustawiać ręcznie, gdyż istnieją świetne narzędzia graficzne.
- Pełna znajomość CSS jest potrzebna do zrozumienia działania, a tym samym, pełnej kontroli wyglądu treści. Wszystko znajduje się w dokumentacji na stronie <https://developer.mozilla.org/en-US/docs/Web/CSS>.
- W zasadzie wyłącznie ręczne pisanie stylów jest podatne na błędy, gdyż istniejące rozwiązania są różnymi kombinacjami istniejących właściwości CSS. Nie wystarczy ich zapamiętać, trzeba lat praktyki nad wieloma projektami.

Przydatne narzędzia

- <https://layout.bradwoods.io/>
- <https://cssgrid-generator.netlify.app/>
- <https://cssgenerator.pl/en/flexbox-generator/>
- <https://angrytools.com/css-flex/>

- Children

9
- Grid

*** Flexbox
- Rows

3

number
- Columns

3

number
- Row Height

1

fr
- Column Width

1

fr
- Row Gap

8

px
- Column Gap

8

px



Hide

CSS

Copy

```
.layout {  
  width: 100%;  
  
  display: grid;  
  grid-template-rows: repeat(3, 1fr);  
  grid-template-columns: repeat(3, 1fr);  
  gap: 8px;  
}
```

HTML

Copy

```
<section class="layout">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>  
  <div>5</div>  
  <div>6</div>  
  <div>7</div>  
  <div>8</div>  
  <div>9</div>  
</section>
```



CSS Grid Generator

built with 🍷 by sarah_edo



1fr

1fr

1fr

1fr

1fr

1fr

1fr

1fr

1fr

1fr



Columns

5

Rows

5

Column Gap (in px)

0

Row Gap (in px)

0

Please may I have some code

Reset grid

What does this project do?

CSS flexbox

display: flex

flex-direction: row

flex-wrap: wrap

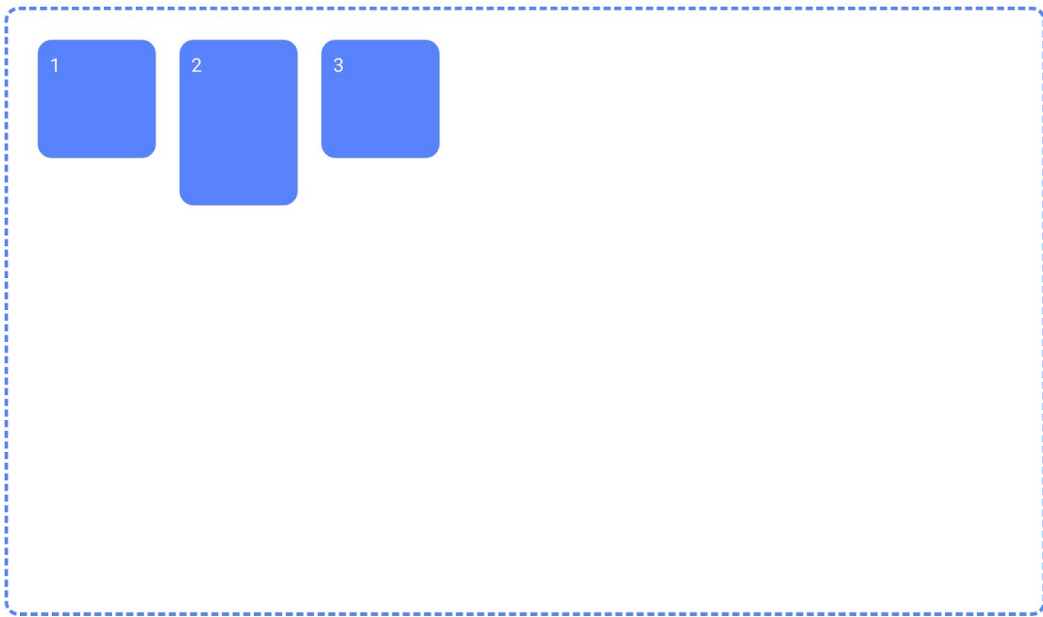
justify-content: flex-start

align-items: flex-start

align-content: flex-start

+ Add

- Remove



CSS Code:

Copy

```
display: flex;  
flex-direction: row;  
flex-wrap: wrap;  
justify-content: flex-start;  
align-items: flex-start;  
align-content: flex-start;
```

Selektory

Definiowanie zbiorów właściwości
dla wybranych znaczników

- Selektor uniwersalny *** {}**
 - Selektor typu **h1, h2, h3 {}**
 - Identyfikator **#id {}**
 - Klasa **.class {}**
 - Selektor dziecka **ul > li {}**
 - Selektor potomka **p a {}**
 - Selektor rodzeństwa **h1 ~ p {}**
 - Selektor sąsiadującego rodzeństwa **h1 + p {}**
 - Selektor atrybutu **div[attributes="Wartość"] {}**
-

Zapytania o media

Kluczowy aspekt działania dla responsywnej strony, to jej umiejętność przystosowania się do różnych urządzeń i ich ekranów.

CSS Media queries to sposób na ukierunkowanie przeglądarki na określone cechy, cechy i preferencje użytkownika, a następnie zastosowanie stylów lub uruchomienie innego kodu w oparciu o te rzeczy. Być może najczęstszymi zapytaniami multimedialnymi na świecie są te, które są ukierunkowane na określone zakresy widoków i stosują niestandardowe style, co zrodziło całą ideę responsywnego projektowania.

Oprócz szerokości obszaru widoku możemy skupić się na wielu innych rzeczach. Mogą to być rozdzielczość ekranu, orientacja urządzenia, preferencje systemu operacyjnego lub nawet inne spośród wielu rzeczy, które możemy przeszukiwać i wykorzystywać do stylizowania treści.

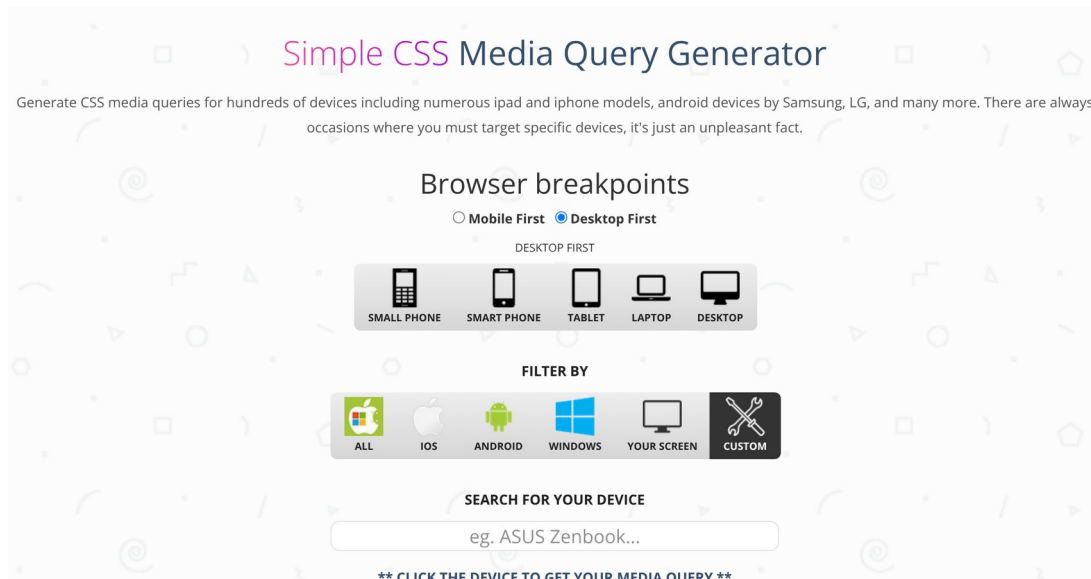
Przykład 2

```
@media only screen and (min-device-width: 320px) and (max-device-width: 480px) {  
  .card {  
    background: #bada55;  
  }  
}
```

@media	screen	(min-width: 320px)	and	(max-width: 768px)
AT-RULE	MEDIA TYPE	MEDIA FEATURE	OPERATOR	MEDIA FEATURE

Narzędzia i ściągowki

- W przypadku tego modułu także istnieje wiele narzędzi w postaci ściągawek, przykładowo: <https://simplecss.eu/>



One Single Page

Strona internetowa, która składa się tylko ze strony głównej i nie zawiera podstron.

Strony internetowe one page składają się z kilku sekcji:

- nagłówek z logo i nawigacją
 - intro, w formie baneru, slidera, czy też wideo w tle oraz wprowadzenia tekstowego
 - sekcje informacyjne (np. o firmie, oferta, galeria)
 - sekcja kontaktowa (z formularzem kontaktowym)
 - stopka
-

Nawigacja na stronie One Single Page

- Strony mają charakter wertykalny, potrafią być bardzo długie. Wygodną nawigację na nich zapewniają **kotwice (anchor)**.
- Aby wykorzystać ten mechanizm korzystamy ze znaczników **a (hiperłącze)** oraz **identyfikatorów (id)**.
- Do dowolnego elementu strony zawierającego atrybut id, możemy odwołać się wykorzystując łącze znajdujące w tym samym dokumencie **HTML** bez konieczności ponownego jego przeładowywania. To oszczędza czas.
- Podejście to jest kluczowe podczas tworzenia nowoczesnych stron WWW.
- Link może przenosić do kotwicy znajdującej się także w innym dokumencie HTML.

Przykład 3

```
<nav>
  <a href="#pomidor">Czerwone warzywo</a>
  <a href="https://adresstrony.pl/podstrona#ogórek">Zielone
warzywo</a>
</nav>

<section id="pomidor">
  <h1>Pomidory</h1>
  <p>Treść artykułu na temat pomidorów</p>
</section>

<section id="ogórek">
  <h1>Ogórki</h1>
  <p>Treść artykułu na temat ogórków</p>
</section>
```

Ta sekcja znajduje się pod
wskazaniem:

<https://adresstrony.pl/podstrona>

Typografia

Reguły **@font-face**, **@import** pozwalają na osadzanie na stronie wybranych krojów czcionek.

- Wadą rozwiązania jest konieczność zapewnienia wielu formatów czcionek na różnych przeglądarkach.
- Bezpieczną opcją wydaje się pobranie krojów czcionek i umieszczenie ich wraz z plikami HTML, nie można polegać na zewnętrznych, które mogą się zmienić.
- Dodanie dużej ilości fontów zwiększa pojemność strony, może spowolnić jej działanie.

Narzędzia

- Świetnym narzędziem pozwalającym pobrać niezbędne pliki wraz z regułami stylów CSS **@font-face** jest <https://www.fontsquirrel.com/> oraz <https://www.fontsquirrel.com/tools/webfont-generator>
- Nieco inne podejście proponuje <https://fonts.google.com/> gdzie fonty osadzamy na stronie z pomocą tagu **link** oraz reguły **@import**. Strona generuje nam gotowe reguły prowadzące do linków zewnętrznych. Możemy pobrać kroje i zastąpić je linkami lokalnymi.
- Google Fonts to bogata baza możemy korzystać z plików używając także reguły **@font-face** po odpowiedniej modyfikacji kodu.

Jednostki i kolory w CSS

Rodzaj	Jednostka	Opis	Przykład
Bezwzględne	px	Piksele — najmniejsza standardowa jednostka ekranu.	<code>width: 200px;</code>
	cm , mm	Centymetry i milimetry — użyteczne głównie w druku, nie na ekranie.	<code>width: 10cm;</code>
	in	Cal — 1in = 96px.	<code>width: 1in;</code>
	pt	Punkt — 1pt = 1/72 cala.	<code>font-size: 12pt;</code>
	pc	Pica — 1pc = 12pt.	<code>margin: 2pc;</code>

Rodzaj	Jednostka	Opis	Przykład
Względne	%	Procent wartości elementu nadrzędnego.	<code>width: 50%;</code>
	em	Odnosi się do rozmiaru fontu elementu nadrzędnego.	<code>font-size: 1.5em;</code>
	rem	Odnosi się do font-size elementu <code>html</code> .	<code>font-size: 2rem;</code>
	vw , vh	Procent szerokości (<code>vw</code>) lub wysokości (<code>vh</code>) okna przeglądarki.	<code>width: 50vw; height: 50vh;</code>
	vmin , vmax	Część mniejszego (<code>vmin</code>) lub większego (<code>vmax</code>) wymiaru widoku.	<code>font-size: 5vmin;</code>
	ex	Wysokość litery „x” w danym foncie.	<code>line-height: 2ex;</code>
	ch	Szerokość cyfry „0” w aktualnym foncie.	<code>width: 60ch;</code>

Sposób	Przykład	Opis
Nazwy kolorów	<code>color: red;</code>	147 nazw predefiniowanych w CSS (np. <code>blue</code> , <code>teal</code> , <code>tomato</code>).
RGB	<code>color: rgb(255, 0, 0);</code>	Składowe czerwony, zielony, niebieski (0–255).
RGBA	<code>color: rgba(255, 0, 0, 0.5);</code>	RGB + kanał przezroczystości (0–1).
HEX	<code>color: #ff0000;</code> lub skrót <code>#f00;</code>	Kolory zapisane szesnastkowo.
HEX z alfa	<code>color: #ff000080;</code>	Jak wyżej, ale z kanałem przezroczystości (ostatnie 2 cyfry).
HSL	<code>color: hsl(0, 100%, 50%);</code>	Barwa, nasycenie, jasność (Hue, Saturation, Lightness).
HSLA	<code>color: hsla(0, 100%, 50%, 0.5);</code>	HSL + kanał przezroczystości.
LCH / Lab (CSS Color Level 4)	<code>color: lch(60% 70 40);</code>	Nowoczesne modele koloru — bardziej precyzyjne, lepsze dla dostępności.

Pseudoklasy

Odpowiedniki funkcji i zdarzeń na
stronie WWW

- pseudoklasy akcji
- pseudoklasy formularzy
- pseudoklasy strukturalne
- pseudoklasy ogólnego zastosowania

Pseudo classes

<code>p::after</code>	Add content after p
<code>p::before</code>	Add content before p
<code>p::first-letter</code>	First letter in p
<code>p::first-line</code>	First line in p
<code>::selection</code>	Selected by user
<code>::placeholder</code>	Placeholder attribute
<code>:root</code>	Documents root element
<code>:target</code>	Highlight active anchor
<code>div:empty</code>	Element with no children
<code>p:lang(en)</code>	P with en language attribute
<code>:not(span)</code>	Element that's not a span

Input pseudo classes

<code>input:checked</code>	Checked inputs
<code>input:disabled</code>	Disabled inputs
<code>input:enabled</code>	Enabled inputs
<code>input:focus</code>	Input has focus
<code>input:in-range</code>	Value in range
<code>input:out-of-range</code>	Input value out of range
<code>input:valid</code>	Input with valid value
<code>input:invalid</code>	Input with invalid value
<code>input:optional</code>	No required attribute
<code>input:required</code>	Input with required attribute
<code>input:read-only</code>	With readonly attribute
<code>input:read-write</code>	No readonly attribute
<code>input:indeterminate</code>	With indeterminate state

Structural pseudo classes

<code>p:first-child</code>	First child
<code>p:last-child</code>	Last child
<code>p:first-of-type</code>	First of some type
<code>p:last-of-type</code>	Last of some type
<code>p:nth-child(2)</code>	Second child of its parent
<code>p:nth-child(3n+2)</code>	Nth-child (an + b) formula
<code>p:nth-last-child(2)</code>	Second child from behind
<code>p:nth-of-type(2)</code>	Second p of its parent
<code>p:nth-last-of-type(2)</code>	...from behind
<code>p:only-of-type</code>	Unique of its parent
<code>p:only-child</code>	Only child of its parent

User action pseudo classes

<code>a:link</code>	Link in normal state
<code>a:active</code>	Link in clicked state
<code>a:hover</code>	Link with mouse over it
<code>a:visited</code>	Visited link

Animacje i przejścia

Stanowią ukoronowanie pracy webdesignera, zapewniając dynamikę i efektowny wygląd eksponowanej treści dokumentu HTML.

- Efekty uzyskane przy pomocy CSS można wykonać wykorzystując skrypty JavaScript. CSS upraszcza robienie interaktywnych zachowań ale też znacznie ogranicza ich możliwości.
 - Warto pamiętać, że przeładowanie strony wodotryskami nie wpływa dobrze na jakość jej użytkowania. Czytelność, klarowność uzyskuje się poprzez kolorystykę, układ treści i czystą typografię.
-

Właściwości CSS dla efektów specjalnych

- animation
 - @keyframes
 - transition
 - transform
- **animation: bounce 300ms linear infinite;**
 - @keyframes mymove {
 from {top: 0px;}
 50% {background: blue !important;}
 to {top: 200px;}
}
 - **transition: margin-right 2s ease-in-out;**
 - div {
 width: 150px;
 height: 80px;
 background-color: yellow;
 transform: rotate(20deg);
}

Potęga synergii

- Aby uzyskać pełnię możliwości efektów jakie dają animacje, przejścia i transformacje należy je połączyć. Dodatkowy efekt pozwalają uzyskać funkcje wbudowane w CSS. Na szczęście nie trzeba pamiętać wszystkiego gdyż istnieją świetne generatory takie tak: <https://animista.net/>
- Przegląd ciekawych, gotowych animowanych komponentów na strony znajdziesz setki, na <https://prismic.io/blog/css-animation-examples> jak i innych stronach w sieci.
- Ogranicza Cię tylko wyobraźnia i znajomość podstaw bez których nie wykorzystasz ani nie zmienisz gotowych szablonów.

Wykorzystuj gotowe komponenty

- W Internecie istnieje gotowych rozwiązań w postaci zgrupowanych właściwości **CSS**; fragmentów stron: slider, formularz, animacji jak i gotowych już pełnych layoutów stron i bibliotek.
- Klucza jest znajomość podstaw **HTML**, **CSS** oraz **JavaScript** by umieć samemu skomponować stronę z wielu elementów.
- Czym lepiej znasz **CSS** tym łatwiej poradysz sobie w poszukiwaniu informacji i stosowaniu gotowych rozwiązań ale nie trzeba wszystkiego znać na pamięć.
- Na stronie: <https://templatemo.com/> znajdziesz dziesiątki gotowych stron do pobrania i edycji.

Linki - dokumentacja

- <https://devdocs.io/css/>
- <https://developer.mozilla.org/en-US/docs/Web/CSS>

Linki - kursy CSS

- <https://www.w3schools.com/css/default.asp>
- <https://web.dev/learn/css>

Linki - cheat sheet CSS

- <https://www3.cs.stonybrook.edu/~pramod.ganapathi/doc/CSE102/CSE102-CheatSheetCSSLong.pdf>
- <https://admm.co/css-cheat-sheet/>
- <https://htmlcheatsheet.com/css/>
- <https://quickref.me/css3>
- <https://www.codewithharry.com/blogpost/css-cheatsheet>
- <https://css3clickchart.com/>
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://css-tricks.com/almanac/>

Linki - grafiki cheat sheet

- <https://makeawebsitehub.com/wp-content/uploads/2015/10/css3-mega-cheat-sheet.jpg>
- https://i0.wp.com/javaconceptoftheday.com/wp-content/uploads/2022/07/CSS_Cheat_Sheet.png?ssl=1
- <https://www.reveillere.fr/M2WEB/sheets/css.pdf>

Koniec

