Kurs Front-End Developer

CSS3

SELEKTORY

Selektor służy do wyszukania elementu bądź grupy elementów w danym dokumencie HTML, które mają otrzymać dany styl CSS, czyli dane właściwości CSS, które odpowiadają za zmianę wyglądu odszukanych elementów HTML.

Istnieją różne rodzaje selektorów takie jak:

- selektor uniwersalny;
- selektor elementu;
- selektor identyfikatora czy klasy;
- selektor potomka, dziecka, brata, braci;
- selektor atrybutu;
- pseudo-selektor czyli selektory pseudo-elementów oraz pseudo-klas.

```
div.background {
    background-color: red;
}
```

selektor div.background odszuka wszystkie elementy div z klasą background i zmieni kolor tła na czerwony

SELEKTORY

selektor uniwersalny

Tworzymy go za pomocą gwiazdki. Selektor ten odwołuje się do każdego typu elementu w dokumencie HTML.

```
* {
    color: #313131;
}
```

wszystkie elementy będą miały kolor czcionki #313131

SELEKTORY (I-***)

selektor pseudo-elementów

Pseudo-elementy to są elementy, które nie istnieją w drzewie dokumentu.

Pseudo-element składa się z dwóch dwukropków (::), za którymi znajduje się jego nazwa.

SELEKTORY (I-***)

PSEUDO-ELEMENT	OPIS	PRZYKŁAD	WYJAŚNIENIE

::before	Służy do dodania dodatkowej treści na początku zawartości selektora HTML.	div::before {
		content: "*";
		color: #444;
		}
			Na początku zawartości każdego elementu div dodaj * w kolorze #444.

::after	Służy do dodania dodatkowej treści na końcu zawartości selektora HTML.	div::after {
		content: "*";
		color: #444;
		}
			Na końcu zawartości każdego elementu div dodaj * w kolorze #444.

::first-letter	Służy do zmiany wyglądu pierwszej litery tekstu, który znajduje się w podanym elemencie.	p::first-letter {
		color: #444;
		}
			Spraw, aby pierwsza litera tekstu w każdym elemencie p, była koloru #444.

::first-line	Służy do zmiany wyglądu pierwszego wiersza tekstu, który znajduje się w podanym elemencie.	div::first-line {
		color: #444;
		}
			Spraw, aby pierwszy wiersz tekstu w każdym elemencie div, była koloru #444.

SELEKTORY (2-***)

selektor pseudo-klas

Pseudo-klasy wykorzystywane są do nadawania specjalnych właściwości elementom języka HTML. Właściwości te są dodawane do elementów po wykonaniu określonej akcji użytkownika lub gdy element ma określone położenie w strukturze dokumentu.

Pseudo-klasa składa się z dwukropka (:) i nazwy, za którymi może być podana wartość w nawiasie.

SELEKTORY (2-***)

pseudo-klasy łączy

Są przeznaczone dla elementów a.

PSEUDO-KLASY	OPIS	PRZYKŁAD	WYJAŚNIENIE

:link	Określa wygląd nieodwiedzonych linków.	a:link {
		color: #444;
		}
			Nieodwiedzone linki w kolorze #444.

:visited	Określa wygląd odwiedzonych linków.	a:visited {
		color: #666;
		}
			Odwiedzone linki w kolorze #666.

SELEKTORY (2-***)

pseudo-klasy dotyczące działań użytkownika

PSEUDO-KLASY	OPIS	PRZYKŁAD	WYJAŚNIENIE

:hover	Określa wygląd elementu, na który najedziemy kursorem myszy.
	
	WAŻNE: Urządzenia mobilne mogą różnie interpretować ten selektor lub wcale.
		div:hover {
		color: red;
		}
			Spraw, aby w momencie najechania kursorem myszy na element div jego kolor tekstu stał się czerwony.

:active	Określa wygląd elementu od momentu kliknięcia na niego lewym przyciskiem myszy, do momentu gdy ten przycisk zostanie puszczony.	div:active{
		color: yellow;
		}
			Spraw, aby podczas kliknięcia na element div jego kolor tekstu stał się żółty.

:focus	Używane jest do aktualnie aktywnego elementu. Np. Pole input w formularzu, w którym obecnie wpisujemy tekst	input:focus{
		color: red;
		}
			Spraw, aby element input, w którym się aktualnie znajdujemy (chcemy wpisać tekst), zmienił kolor tekstu na czerwony.

SELEKTORY (2-***)

inne pseudo-klasy

PSEUDO-KLASY	OPIS	PRZYKŁAD	WYJAŚNIENIE

:first-child	Reprezentuje element będący pierwszym zagnieżdżonym elementem.	ol li:first-child {
		color: red;
		}
			Spraw, aby kolor tekstu pierwszego elementu listy był czerwony.

:last-child	Reprezentuje element będący ostatnim zagnieżdżonym elementem HTML.	ol li:last-child {
		color: red;
		}
			Spraw, aby kolor tekstu ostatniego elementu listy był czerwony.

:only-child	Służy do wskazania każdego elementu, który jest jedynym dzieckiem swojego rodzica.	span:only-child {
		color: red;
		}
			Element span, który jest jedynym dzieckiem swojego rodzica ma kolor tekstu czerwony.

:empty	Reprezentuje element, który nie posiada zawartości (są puste).	div:empty {
		height: 30px;
		background-color: red;
		}
			Spraw aby element div, który jest pusty miał wysokość 30px i kolor tła czerwony.

SELEKTORY (2-***)

:nth-child() oraz :nth-last-child()

Służy do odszukania elementów (według określonego wzoru) będących dziećmi swojego rodzica. Selektor :nth-child() rozpoczyna liczenie dzieci od pierwszego dziecka do ostatniego dziecka, natomiast selektor :nth-last-child() rozpoczyna liczenie od ostatniego do pierwszego.

Wartości jakie możemy podać wraz z selektorami:

- **even** - selektor odszuka parzyste dzieci;
- **odd** - selektor odszuka nieparzyste dzieci;
- **liczba** - odszuka konkretne dziecko rodzica, np. po podaniu liczby 5, selektor wskaże piąte dziecko;
- **an+b** - wartości a i b muszą być liczbami całkowitymi (dodatnimi, ujemnymi lub równe zero) - możemy podać własny wzór według jakiego selektor ma odszukiwać kolejne dzieci. Liczba **a** – oznacza co jaką liczbę selektor ma wskazywać kolejne dzieci, a liczba **b** oznacza - od którego dziecka ma zacząć się odszukiwanie – czyli offset.

SELEKTORY (3-***)

selektory atrybutów

selector[att] - Reprezentuje element mający atrybut att o dowolnej wartości.

```
selector[title] {
      color: #313131;
}
```

Wszystkie elementy, które posiadają atrybut title mają kolor tekstu #313131.

selector[att="val"] - Reprezentuje element mający atrybut att o wartości „val”.

```
selector[title="opis"] {
      color: #313131;
}
```

Wszystkie elementy, które posiadają atrybut title o wartości „opis” mają kolor tekstu #313131.

SELEKTORY (3-***)

selector[att~="val"] - Reprezentuje element mający atrybut att, którego wartość zawiera w sobie „val” (musi być on oddzielony spacją od pozostałych ciągów znaków).

```
selector[title~="opis"] {
    color: #313131;
}
```

Wszystkie elementy, które posiadają atrybut title, którego wartość zawiera w sobie wyraz „opis” mają kolor tekstu #313131.

selector[att*="val"] - Reprezentuje element mający atrybut att, którego wartość zawiera przynajmniej jedno wystąpienie łańcucha „val”.

```
selector[title*="opis"] {
    color: #313131;
}
```

Wszystkie elementy, które posiadają atrybut title, którego wartość zawiera w sobie ciąg znaków „opis” mają kolor tekstu #313131.

SELEKTORY (3-***)

selector[att^="val"] - Reprezentuje element mający atrybut att, którego wartość zaczyna się od łańcucha „val”.

```
selector[title^="opis"] {
        color: #313131;
}
```

Wszystkie elementy, które posiadają atrybut title, którego wartość zaczyna się od łańcucha znaków „opis” mają kolor tekstu #313131.

selector[att$="val"] - Reprezentuje element mający atrybut att, którego wartość kończy się łańcuchem „val”.

```
selector[title$="opis"] {
        color: #313131;
}
```

Wszystkie elementy, które posiadają atrybut title, którego wartość kończy się ciągiem znaków „opis” mają kolor tekstu #313131.

SELEKTORY (3-***)

Stylowanie pól formularza

Selektory atrybutów najczęściej wykorzystuje się do nadania jednolitego wyglądu polom formularzy w naszych projektach.

Do ostylowaniania pól typu **input** zastosujemy odwołanie do selektora atrybutu

input[type=”text”] – Taki selektor nada jednolity wygląd wszystkim polom do wprowadzania tekstu w naszym formularzu

WARSZTATY ☺
STYLOWANIE PÓL FORMULARZA

Wykorzystując dotychczasową wiedzę wykonaj zadanie:
Nadaj stworzonemu przez siebie w ramach zadnia domowego formularzowi następujący wgląd

https://akademia108.pl/kurs-front-end/form-css.png

Kody kolorów:

- tło: #213b50
- pole nieaktywne: #2b4a64
- pole aktywne: #4e708c
- ramka aktywnego pola i kolor tekstu: #ffffff

WYŚWIETLANIE ELEMENTÓW (4-***)

ELEMENTY BLOKOWE (KRÓTKA POWTÓRKA)

Elementy blokowe to takie elementy, które domyślnie wyświetlane są jeden pod drugim. Do tych elementów należą np. div, p czy nagłówki od h1 do h2. Przestrzeń jaką zajmuje element blokowy jest uzależniona od szerokości jego elementu rodzica, domyślnie jest to 100% szerokości rodzica.

WYŚWIETLANIE ELEMENTÓW (4-***)

ELEMENTY LINIOWE (KRÓTKA POWTÓRKA)

Elementy liniowe to takie elementy, które domyślnie wyświetlane są w jednej linii (obok siebie). Takimi przykładowymi elementami są: span, i czy a. Domyślna przestrzeń (wysokość i szerokość) jaką zajmuje element wyświetlany w linii jest uzależniona od zawartości jaka się w nim znajdzie.

Elementy liniowe w gruncie rzeczy są traktowane przez przeglądarkę internetową jak tekst. Wyśrodkowanie elementu liniowego jest możliwe, jeżeli do elementu rodzica, który jest elementem blokowym, zostanie dodana właściwość text-align:center

Ponadto niektóre właściwości CSS **nie mają zastosowania** do elementów liniowych. Tymi właściwościami są, np. width oraz height

WYŚWIETLANIE ELEMENTÓW (4-***)

WŁAŚCIWOŚĆ display

Do zmiany sposobu wyświetlania elementów na stronie służy właściwość CSS – display.

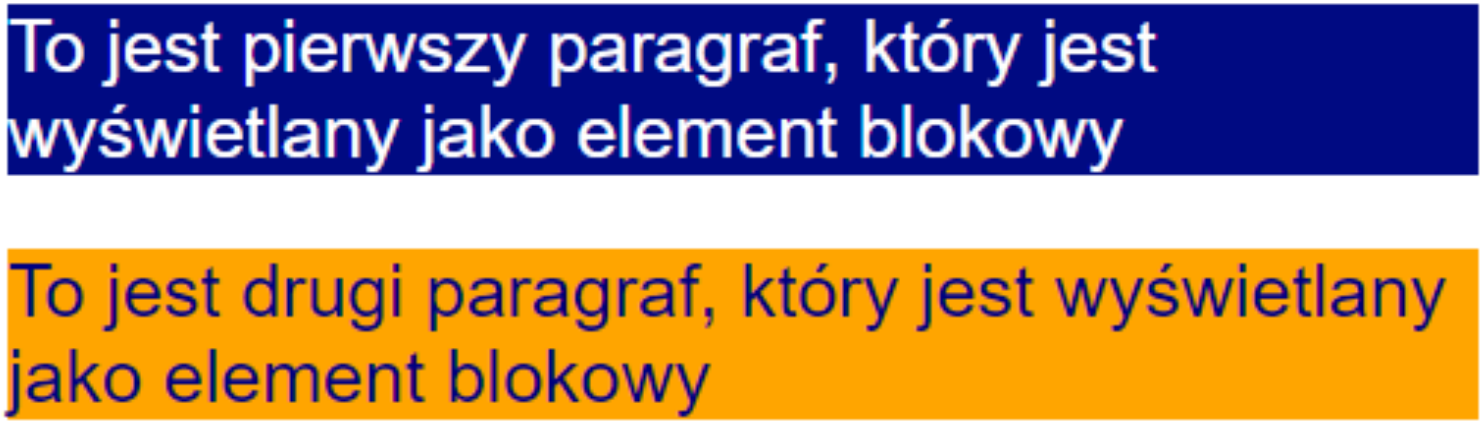
Zmieniając domyślną wartość właściwości display danego elementu HTML sami możemy określić czy dany element HTML ma być "rozumiany" przez przeglądarkę internetową, np. jako element liniowy inline, element blokowy block lub inny typ elementu HTML.

Każdy element HTML posiada domyślną wartość własności display. Najczęściej domyślną wartością właściwości display są wartość inline lub block.

WYŚWIETLANIE ELEMENTÓW (4-***)

display: block;

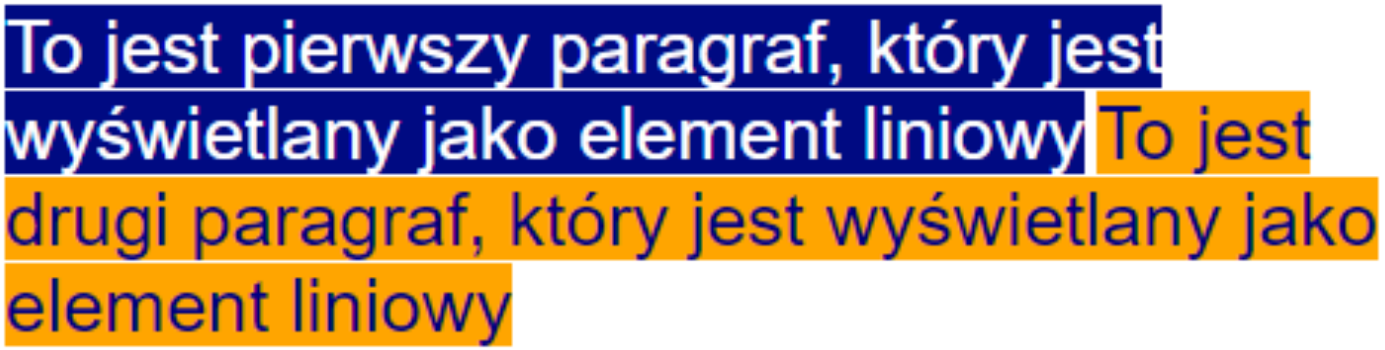
Element HTML będzie wyświetlany w bloku (jeden pod drugim). Element będzie interpretowany przez przeglądarkę internetową jako element blokowy.



WYŚWIETLANIE ELEMENTÓW (4-***)

display: inline;

Element HTML będzie wyświetlany w linii (jeden obok drugiego). Element będzie interpretowany przez przeglądarkę internetową jako element liniowy.



WYŚWIETLANIE ELEMENTÓW (4-***)

display: inline-block;

Element HTML będzie interpretowany przez przeglądarkę internetową jako element liniowy, lecz z zachowaniem pewnych cech elementu blokowego.

Element z daną wartością będzie interpretowany przez swojego rodzica jako jego element liniowy, dlatego wyśrodkowanie go będzie możliwe za pomocą
text-align: center;

Element z display: inline-block będzie miał pewne cechy elementu blokowego , dzięki czemu takie właściwości jak: padding, margin, width czy height będą na niego oddziaływały jak na normalny element blokowy.

WYŚWIETLANIE ELEMENTÓW (4-***)

```
<ul>
    <li>1</li>
    <li>2</li>
    <li>3</li>
    <li>4</li>
    <li>5</li>
    <li>6</li>
</ul>
```

elementy **ul->li**, które zostały zaprezentowane w tym przykładzie, mają domyślnie cechy elementu blokowego. Po zmianie właściwości **display** elementow li na wartość **inline-block** wyświetlają się one w jednej linii jak elementy liniowe inline, równocześnie zachowując właściwości elementów blokowych block np. width i heigth

WYŚWIETLANIE ELEMENTÓW (4-***)

display: none;

Element nie będzie wyświetlany.

OPŁYWANIE ELEMENTÓW - FLOATING (5-***)

Właściwość float określa czy element powinien być opływany i z której strony
Właściwość clear używamy do kontroli opływania elementów HTML

Przykład:

```
img { float: left; }
```

Bootcamp to coś więcej niż czysta teoria kodowania. To kurs, na którym pokażemy Ci jak

szybko przyswajać niezbędne umiejętności oraz wyrobić w sobie nawyki ułatwiające programowanie.Przybliżymy Ci narzędzia, które znacznie ułatwią pracę, a także wytłumaczymy, na jakiej zasadzie działają. Kładziemy nacisk przede wszystkim na praktyczne umiejętności i przekładamy je ponad suchą akademicką wiedzę. Pokażemy Ci jak uczyć się szybko i mieć przy tym niezłą frajdę.

OPŁYWANIE ELEMENTÓW - FLOATING (5-***)

```
img { float: right; }
```

Bootcamp to coś więcej niż czysta teoria kodowania. To kurs, na którym pokażemy Ci jak szybko przyswajać niezbędne umiejętności oraz wyrobić w sobie nawyki ułatwiające programowanie.Przybliżymy Ci narzędzia, które znacznie ułatwią pracę, a także wytłumaczymy, na jakiej zasadzie działają. Kładziemy nacisk przede wszystkim na praktyczne umiejętności i przekładamy je ponad suchą akademicką wiedzę. Pokażemy Ci jak uczyć się szybko i mieć przy tym niezłą frajdę.

```
img { float: none; }
```

Bootcamp to coś więcej niż czysta teoria kodowania. To kurs, na którym pokażemy Ci

jak szybko przyswajać niezbędne umiejętności oraz wyrobić w sobie nawyki ułatwiające programowanie.Przybliżymy Ci narzędzia, które znacznie ułatwią pracę, a także wytłumaczymy, na jakiej zasadzie działają. Kładziemy nacisk przede wszystkim na praktyczne umiejętności i przekładamy je ponad suchą akademicką wiedzę. Pokażemy Ci jak uczyć się szybko i mieć przy tym niezłą frajdę.

OPŁYWANIE ELEMENTÓW - FLOATING (5-***)

```
img { float: left; }
p:last-child { clear: both; }
```

Bootcamp to coś więcej niż czysta teoria kodowania.

To kurs, na którym pokażemy Ci jak szybko przyswajać niezbędne umiejętności oraz wyrobić w sobie nawyki ułatwiające programowanie. Przybliżymy Ci narzędzia, które znacznie ułatwią pracę, a także wytłumaczymy, na jakiej zasadzie działają. Kładziemy nacisk przede wszystkim na praktyczne umiejętności i przekładamy je ponad suchą akademicką wiedzę. Pokażemy Ci jak uczyć się szybko i mieć przy tym niezłą frajdę.

EFEKTY PRZEJSCIA W CSS (6-***)

Właściwość **transition** pozwala utworzyć pewien efekt animacji.

Wspomniana animacja zmieni wartości właściwości CSS dla elementu HTML. Prościej mówiąc na ekranie monitora zaobserwujemy przejście jednych wartości właściwości CSS w drugie – efekt przejścia.

Na przykład po najechaniu myszką na element div (pseudoklasa HOVER), tło elementu div zmieni (przejdzie) z białego na czerwony. Wspomniany efekt możemy uzyskać za pomocą pseudoklasy :hover

Zmiana ta dokona się skokowo, jednak dzięki właściwościom pochodzącym z grupy transition możemy uzyskać efekt animacji.

ZAPIS JEDNOLINIJKOWY:

transition: **wlasciwosc_CSS duration timing_function delay**;
transition: color 5s ease-in 2s;
transition: color 5s ease-in 2s, width 3s linear 3s ;

EFEKTY PRZEJSCIA W CSS (6-***)

właściwości efektu transition

transition-property:

- określa, jaka własność stylu ma podlegać efektowi przejścia
- dozwolone wartości:
  - ✓ all - wszystkie właściwości CSS mogą brać udział w efekcie przejścia – domyślna wartość
  - ✓none - żadna z właściwości CSS nie może brać udziału w efekcie przejścia – wyłącza efekt przejścia
  - ✓ wybrana właściwość / wybrane właściwości
- przykład

```
transition-property: background-color;
```

EFEKTY PRZEJSCIA W CSS (6-***)

transition-duration:

- określa, czas trwania efektu przejścia
- dozwolone wartości:
  - ✓ jednostki czasu – sekundy (s) oraz milisekundy (ms)
- przykład

```
transition-duration: 2s;
```

transition-delay:

- określa, czas opóźnienie efektu przejścia
- dozwolone wartości:
  - ✓ jednostki czasu – sekundy (s) oraz milisekundy (ms)
  - ✓ może przyjmować wartości dodatnie (opóźnienie), ujemne (przyśpieszenie) oraz zero (brak opóźnienia, przyśpieszenia)
- przykład

```
transition-delay: 0.5s;
```

EFEKTY PRZEJSCIA W CSS (6-***)

transition-timing-function:

- określa, tempo efektu przejścia
- dozwolone wartości:
  - ✓ linear - stałe tempo efektu przejścia
  - ✓ ease-in - wolniejsze tempo efektu przejścia na początku
  - ✓ ease-out - wolniejsze tempo efektu przejścia na końcu
  - ✓ ease-in-out - wolniejsze tempo efektu przejścia na początku oraz na końcu
  - ✓ steps() - ilość klatek animacji efektu przejścia określona za pomocą funkcji steps()
  - ✓ step-start - stałe tempo efektu przejścia z pominięciem pierwszej klatki
  - ✓ step-end - stałe tempo efektu przejścia z pominięciem ostatniej klatki
  - ✓ ease - efekt przejścia będzie miał wolniejszy start, lecz później przyspieszy, a następnie zwolni tempo przed zakończeniem swojego czasu wykonywania się - wartość domyślna
- przykład

```
transition-timing-function: linear;
```

EFEKTY PRZEJSCIA W CSS (6-***)

transition:

- służy do zapisania wszystkich zaprezentowanych właściwości efektu transition za pomocą jednej reguły
- skrótowy zapis wszystkich właściwości efektu transition, w kolejności: property, duration, timing function, delay
- przykład

```
transition-property: color;
transition-duration: 5s;
transition-timing-function: ease-in;
transition-delay: 2s;

transition: color 5s ease-in 2s;
```

Zarówno własności transition-* jak i skrótowa własność transition mogą przyjmować wiele wartości równolegle

```
transition: color 5s ease-in 2s, height 5s ease-in 2s;
transition: all 5s ease-in 2s;
```

WARSZTATY ☺ EFEKT TRANSITION

Wykorzystując dotychczasową wiedzę wykonaj następujące zadanie:
Zrób animację boxów z wykorzystaniem właściwości transition na HOVER

1. Stwórz trzy boxy w rozmiarze 150px na 150px
2. Pierwszy z nich ma się zmieniać szerokość do 300px z wykorzystaniem efektu transition
3. Drugi z nich ma się zmieniać wysokość do 300px z wykorzystaniem efektu transition
4. Trzeci z nich ma się zmieniać wysokość i szerokość do 300px z wykorzystaniem efektu transition

Podgląd:

https://akademia108.pl/kurs-front-end/transition/

WŁAŚCIWOŚCI TŁA (7-***)

Za pomocą CSS możemy określić rodzaj tła dla konkretnego elementu tzn. czy ma być to kolor czy obrazek.

Jeśli zdecydujemy się na grafikę w tle możemy zdecydować czy tło ma być powielane czy nie lub czy ma być statyczne czy się przewijać. Dodatkowo możliwe jest określenie pozycji tła w elemencie.

WŁAŚCIWOŚCI TŁA (7-***)

background-color: kolor | transparent | initial | inherit;

Kolor tła danego elementu/elementów.

Dozwolone wartości:

- kolor – określenie własnej wartości koloru tła elementu (np. hex, rgb)
- transparent – przezroczyste tło wartość domyślna
- initial – ustawienie wartości domyślnej przeglądarki
- inherit – dziedzicz po rodzicu

Przykład

background-color: red;

WŁAŚCIWOŚCI TŁA (7-***)

background-image: none | url() | initial | inherit | funkcje gradientowe;

Wypełnienie tła elementu obrazkiem w formacie np. .jpg, .png, .gif.

Za pomocą tej właściwości możemy wypełnić tło gradientem.

Dozwolone wartości:

- none – żadna grafika nie jest wyświetlana
  wartość domyślna
- url() – ścieżka do grafiki tła
- initial – ustawienie wartości domyślnej
- inherit – dziedzicz po rodzicu
- funkcje gradientowe – np.
  linear-gradient(red, yellow);

Przykład

background-image: url('/img/obrazek.png');

WŁAŚCIWOŚCI TŁA (7-***)

background-repeat: repeat | repeat-x | repeat-y | no-repeat | initial | inherit;

Określa sposób powtarzania się obrazka w tle elementu.

Dozwolone wartości:

- repeat – grafika powtarzana w pionie oraz poziomie
- repeat-x – grafika powtarzana w poziomie
- repeat-y – grafika powtarzana w pionie
- no-repeat – grafika nie jest powtarzane
- initial – ustawienie wartości domyślnej
- inherit – dziedzicz po rodzicu

background-repeat: repeat-y;

WŁAŚCIWOŚCI TŁA (7-***)

background-position: left top | left center | left bottom | center top | center center | center bottom | right top | right center | right bottom | x% y% | x y | initial | inherit;

Określa pozycję obrazka w tle elementu.

Dozwolone wartości:

- left/center/right top/center/bottom – wyrównanie według wskazanych parametrów – jeśli podana jest tylko jedna wartość druga ustawiana jest na center
- x% y% – pozycja tła w procentach – wartość domyślna 0% 0%
- x y – pozycja tła w jednostkach długości
- no-repeat – grafika nie jest powtarzane
- initial – przywrócenie wartości domyślnej przeglądarki
- inherit – dziedzicz po rodzicu

background-position: 180px 10px;

WŁAŚCIWOŚCI TŁA (7-***)

background-attachment: scroll | fixed | local | initial | inherit;

Określa w jaki sposób ma zachowywać się tło obrazkowe w danym elemencie, w momencie gdy będziemy przewijać zawartość strony za pomocą suwaków przeglądarki internetowej lub w momencie gdy będziemy przewijać zawartość danego elementu za pomocą suwaków.

Dozwolone wartości:

- scroll – tło elementu będzie się przewijać wraz z tym elementem – wartość domyślna
- fixed – grafika tła nieruchoma względem okna przeglądarki
- local – grafika tła przewija się wraz z zawartością elementu
- initial – przywrócenie wartości domyślnej przeglądarki
- inherit – dziedzicz po rodzicu

WŁAŚCIWOŚCI TŁA (7-***)

background-size: auto | szer wys | szer% wys% | cover | contain | initial | inherit;;

Określa rozmiar obrazka w tle elementu.

Dozwolone wartości:

- auto – rzeczywiste wymiary obrazka – wartość domyślna
- szer wys – szerokość i wysokość tła obrazka w jednostkach miary (np. px), jeśli ustawiona jest tylko jedna wartość druga ustawiona jest na auto
- szer% wys% – szerokość i wysokość tła obrazka w procentach rozmiaru elementu nadrzędnego, jeśli ustawiona jest tylko jedna wartość druga ustawiona jest na auto
- cover – skalowanie grafiki tła elementu aby cały element został wypełniony grafiką. Zostają zachowane proporcje grafiki. Część grafiki nie będzie widoczna, jeśli szerokość i wysokość elementu nie są proporcjonalne do rozmiarów grafiki
- contain – skalowanie grafiki tła elementu aby cała grafika była widoczna. Zostają zachowane proporcje grafiki. Część elementu nie będzie wypełniona grafiką jeśli rozmiar tła i grafiki się nie pokrywają
- initial – przywrócenie wartości domyślnej przeglądarki
- inherit – dziedzicz po rodzicu

WŁAŚCIWOŚCI TŁA (7-***)

background:

- służy do zapisania wszystkich zaprezentowanych właściwości tła za pomocą jednej reguły
- skrótowy zapis wszystkich właściwości tła

- przykład

```
background-color: white;
background-image: url('/img/obrazek.png');
background-position: center top;
background-size: 50% 75%;
background-repeat: no-repeat;
background-attachment: scroll;

background: white url('/img/obrazek.png') center top no-repeat scroll;
```

PRZEPEŁNIENIE ELEMENTU – SCROLL (8-***)

Jeżeli zawartość elementu nie mieści się w jego rozmiarach możliwe jest ukrycie nadmiarowej zawartość, pokazanie wszystkiego poprzez powiększenie rozmiarów elementu (bez względu na parametry width i height) albo utworzenie suwaków do przewijania przepełnionej treści. Służy do tego właściwość CSS – overflow, overflow-x oraz overflow-y.

Właściwości te informują przeglądarkę internetową co zrobić z zawartością elementu jeśli nie mieści się ona w tym elemencie.

Właściwości te działają tylko dla **elementów blokowych** z określoną wysokością.

Dozwolone wartości:

- scroll – dodanie suwaków do elementu, nawet jeśli zawartość mieści się w elemencie
- hidden – ukrycie niemieszczącej się zawartości elementu
- auto – dodanie suwaków do elementu gdy zawartość nie będzie się mieścić
- visible – niemieszcząca się zawartość elementu pozostanie widoczna – wartość domyślna
- initial – przywrócenie wartości domyślnej przeglądarki
- inherit – dziedzicz po rodzicu

PRZEPEŁNIENIE ELEMENTU – SCROLL (8-***)

overflow: visible;

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.

overflow: hidden;

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.

overflow: scroll;

Sed ut perspiciatis unde omnis iste natus error sit voluptatem

overflow: auto;

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium

PRZEPEŁNIENIE ELEMENTU – SCROLL (8-***)

overflow

Ustala jak ma zachować się element gdy jego zawartość nie będzie się mieścić w granicach jego rozmiaru (wysokość i szerokość).

overflow-x

Ustala jak ma zachować się element gdy jego zawartość nie będzie się mieściła w granicy jego szerokości.

overflow-y

Ustala jak ma zachować się element gdy jego zawartość nie będzie się mieściła w granicy jego wysokości.

ROZMIAR ELEMENTU (9-***)

Aby określić konkretną szerokość elementu HTML używa się właściwości CSS - width, a do określenia konkretnej wysokości elementu HTML - właściwości CSS height.

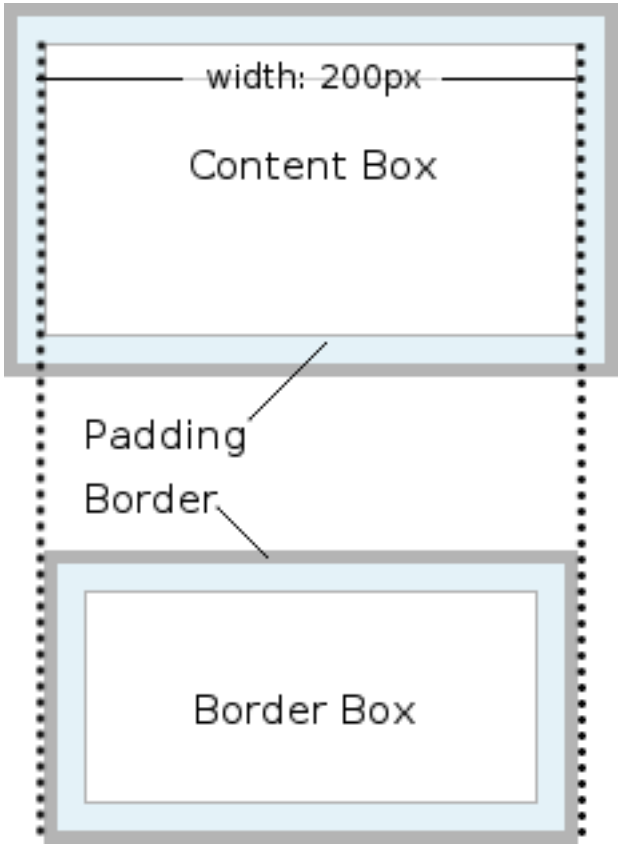
Niektóre właściwości CSS mają wpływ na końcowy rozmiar elementu (szerokość oraz wysokość), obliczany przez przeglądarkę internetową. Właściwości te to: padding, margin, border, line-height.

Za pomocą właściwości **box-sizing** możemy ustalić czy właściwości border oraz padding mają wpływać na całkowitą szerokość i wysokość elementu, które są obliczane przez przeglądarkę internetową.

Dostępne wartości:

- border-box – właściwości padding oraz border nie mają wpływu na ustalony rozmiar elementu
- content-box – właściwości padding oraz border zmieniają ustalony rozmiar elementu – wartość domyślna
- initial – przywrócenie wartości domyślnej przeglądarki
- inherit – dziedzicz po rodzicu

ROZMIAR ELEMENTU (9-***)



box-sizing: border-box;

box-sizing: content-box;

RESPONSYWNOŚĆ ZA POMOCĄ CSS (10-***)

W celu uzyskania efektu automatycznego skalowania i odpowiedniej szerokości obrazu na urządzeniach mobilnych używamy metatagu viewport, który umieszczamy w nagłówku strony <head>

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

Wartość atrybutu content definiuje:

- Wyjściową szerokość strony - width=device-width – szerokość wyświetlacza urządzenia
- Wyjściowe powiększenie - initial-scale=1.0 – domyślne powiększenie

RESPONSYWNOŚĆ ZA POMOCĄ CSS (10-***)

Aby stworzyć **responsywną stronę internetową (RWD)** czyli stronę, która będzie dobrze wyglądać zarówno na laptopach i na urządzeniach mobilnych, możemy użyć gotowych frameworków CSS np. Bootstrapa, ale również możemy stworzyć sami taką stronę za pomocą CSS, a dokładniej reguł **@media queries**.

Reguły @media queries służą do definiowania różnych właściwości CSS dla różnych typów mediów/urządzeń/rozdzielczości.

Za pomocą tych reguł możemy sprawdzić np.:

- typ urządzenia np. screen – komputery z kolorowym ekranem, all – dla wszystkich urządzeń
- wielkość okna przeglądarki
- rozdzielczość urządzenia
- orientację pionową lub poziomą urządzenia

RESPONSYWNOŚĆ ZA POMOCĄ CSS (10-***)

width / **height**
min-width / **min-height**
max-width / **max-height**

Określa reguły CSS, które zostaną aktywowane tylko wtedy, gdy **szerokość/ wysokość okna przeglądarki internetowej** będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość.

```
@media (min-width:800px) and (max-width:1200px) {
    div {
        color: red;
    }
}
```

RESPONSYWNOŚĆ ZA POMOCĄ CSS (10-***)

device-width / **device-height**
min-device-width / **min-device-height**
max-device-width / **max-device-height**

Określa reguły CSS, które zostaną aktywowane tylko wtedy, gdy **szerokość/ wysokość rozdzielczości ekranu urządzenia** będzie miała określoną wartość, lub gdy będzie większa niż podana wartość, lub gdy będzie mniejsza niż podana wartość.

```
@media (max-device-width:800px) {
        div {
                color: red;
        }
}
```

RESPONSYWNOŚĆ ZA POMOCĄ CSS (10-***)

orientation

Określa reguły CSS, które zostaną aktywowane tylko wtedy, gdy **szerokość okna przeglądarki internetowej** będzie miała wartość mniejszą lub większą niż wysokość okna przeglądarki internetowej.

Cecha orientation pomaga nam wykryć w jakiej orientacji, pionowej (portrait) czy poziomej (landscape), została wyświetlona nasza strona internetowa.

```
@media (orientation:landscape) {
        div {
                color: red;
        }
}
```

RESPONSYWNOŚĆ ZA POMOCĄ CSS (10-***)

wybrane oznaczenia typów mediów

all – dla wszystkich mediów
screen – dla komputerów, tabletów, smartfonów itp.
print – dla drukarek
speech – dla czytników ekranu

```
@media screen, tv {
        div {
                color: red;
        }
}
```

RESPONSYWNOŚĆ ZA POMOCĄ CSS (10-***)

Tabela przedstawia operatory logiczne dla reguł @media queries

OPERATOR	PRZEZNACZENIE	PRZYKŁAD

and	operator „i”
	
	operator ten łączy różne właściwości w regule @media
		@media all and (max-width:800px) {
		…
		}

not	operator negacji
	
	operator ten umożliwia zaprzeczenia danego zapisu
		@media not screen {
		…
		}

przecinek	operator „lub”
	
	operator ten umożliwia stworzenie jednego zapisu reguły @media, z różnym właściwościami dla różnych typów urządzeń
		@media all, screen, tv {
		…
		}

WARSZTATY ☺ RWD MEDIA QUERIES

Wykorzystując dotychczasową wiedzę wykonaj następujące zadanie:
Dostosuj galerię do urządzeń mobilnych

1. Pobierz paczkę z plikami https://akademia108.pl/kurs-front-end/rwd.zip
2. Galeria jest nieresponsywna i ma bloki nierelatywnej wielkości
3. Stwórz plik responsive.css gdzie nadpisujemy wielkości wyrażone w pikselach na jednostki relatywne
4. Dodaj odpowiednie warunki w media queries, które sprawią, że obrazki w galerii będą w trzech kolumnach dla urządzeń z dużym wyświetlaczem (powyżej 992px), w dwóch dla urządzeń ze średnim wyświetlaczem (powyżej 768px do 992px) i w jednej kolumnie dla urządzeń z małym wyświetlaczem (dla 768px i poniżej)
5. Dla urządzeń z bardzo małym wyświetlaczem (poniżej 480px) główna sekcja oraz obrazki w galerii mają mieć całe 100% szerokości ekranu

Podgląd: https://akademia108.pl/kurs-front-end/rwd/

Akademia 108

https://akademia108.pl