

01. Ogólne

CSS	OPIS	PRZYKŁAD
*	Uniwersalny selektor, odnosi się do wszystkich elementów na stronie.	<code><div></code> <code><p></code> <code><h1></code> <code></code>
<code>div</code> <small>(lub inny tag HTML)</small>	Odnosi się do wszystkich tagów danego typu.	<code><div></code> <code><p></code> <code><div></code> <code></code>
<code>.a</code>	Odnosi się do wszystkich elementów, które posiadają klasę "a".	<code>.a</code> <code>.b</code> <code>.c</code> <code>.a</code>
<code>#a</code>	Odnosi się do elementu, który posiada ID "a".	<code>#a</code> <code>#b</code> <code>#c</code> <code>#d</code>

Wskazówki

- **ID** musi być unikalne! Nie możemy użyć tego samego **ID** na więcej niż jednym elemencie.
- Do stylowania lepiej używać **klas**. **ID** zostawmy do konkretnych zadań realizowanych np. w JavaScript.
- Za pomocą * zerujemy paddingi i marginesy, ustawiamy box-sizing.
Nie przesadzajmy z ilością właściwości CSS umieszczonych w uniwersalnym selektorze, ponieważ wpłyną one dosłownie na każdy element strony.

02. Kombinacje

CSS	OPIS	PRZYKŁAD
<code>div p</code>	Odnosi się do wszystkich <code><p></code> wewnątrz tagów <code><div></code> .	<code><div></code> <code></code> <code><p></code> <code><p></code> <code><h3></code>
<code>div, p, a</code>	Odnosi się do wszystkich wymienionych tagów, a więc zarówno do <code><p></code> , <code><a></code> jak i do <code><div></code> .	<code><div></code> <code><p></code> <code><a></code> <code></code>
<code>p.a</code>	Odnosi się do wszystkich <code><p></code> z klasą "a".	<code><p.a></code> <code><p.b></code> <code><p.a></code> <code></code>
<code>span#b</code>	Odnosi się do wszystkich tagów <code></code> z ID "b". ¹	<code><span#b></code> <code><span#z></code>
<code>div > p</code>	Odnosi się do bezpośrednich dzieci. W pierwszym przykładzie, tylko ostatni <code><p></code> jest bezpośrednim dzieckiem.	<code><div></code> <code></code> <code><p></code> <code><p></code> <code><p></code> <code><div></code> <code><p></code> <code><p></code> <code><p></code> <code><p></code>
<code>div + p</code>	Odnosi się do bezpośredniego rodzeństwa. W naszym przykładzie – do pierwszego <code><p></code> pod <code><div></code> .	<code><div></code> <code><p></code> <code><p></code> <code><p></code> <code><div></code> <code><p></code> <code><div></code> <code><p></code>
<code>div ~ p</code>	Odnosi się do rodzeństwa pod elementem. W naszym przykładzie – do <code><p></code> pod <code><div></code> .	<code><div></code> <code><p></code> <code><p></code> <code><p></code> <code><p></code> <code><p></code> <code><div></code> <code><p></code>

Wskazówki

- Nie warto zapędzać się w zbyt duże zagnieżdżenia np. `body main div p.test {}`.
Zamiast tego, lepiej nadać unikalną klasę danemu elementowi (w naszym przykładzie mowa o paragrafie), dzięki czemu łatwiej będzie ten element edytować, jeśli zajdzie taka potrzeba (mowa tu o specyficzności).

¹ Mam nadzieję, że wychwyciliście błąd. :)

Tagów – liczba mnoga. Jak wiemy, **ID** musi być unikalne, więc **tylko jeden** `` mógłby mieć ID "b".

03. Pseudo klasy

CSS	OPIS	PRZYKŁAD
<code>li:first-child</code>	Odnosi się do pierwszego bezpośredniego dziecka. W naszym przykładzie – do pierwszego <code></code> w <code></code> .	<div><code></code><div><code></code><code></code><code></code><code></code></div></div> <div><code></code><div><code><a></code><code></code><code></code><code></code></div></div>
<code>li:last-child</code>	Odnosi się do ostatniego bezpośredniego dziecka. W naszym przykładzie – do ostatniego <code></code> w <code></code> .	<div><code></code><div><code></code><code></code><code></code><code><a></code></div></div> <div><code></code><div><code></code><code></code><code></code><code></code></div></div>
<code>p:nth-child(Xn)</code>	Odnosi się do X*n elementów, gdzie X to dowolna podana przez nas liczba całkowita, a n to liczba od 1 do +∞. Zapis na górze: <code>p:nth-child(2n)</code> Zapis na dole: <code>p:nth-child(3n)</code>	<div><code><p></code><code><p></code><code><p></code><code><p></code></div> <div><code><p></code><code><p></code><code><p></code><code><p></code></div>
<code>a:first-of-type</code>	Odnosi się do pierwszego elementu danego typu. W naszym przykładzie – do pierwszego <code><a></code> w <code><div></code> .	<div><code><div></code><div><code></code><code><p></code><code><a></code><code><a></code></div></div>
<code>p:nth-of-type(2n)</code>	Połączenie <code>nth-child</code> oraz <code>first-of-type</code> . Odnosi się do n-tego elementu danego typu. Zapis na górze: <code>p:nth-of-type(2n)</code> Zapis na dole: <code>p:nth-of-type(3)</code>	<div><code><div></code><div><code><p></code><code><p></code><code><p></code><code><p></code></div></div> <div><code><div></code><div><code></code><code><p></code><code><p></code><code><p></code></div></div>
<code>p:only-child</code>	Odnosi się do elementów, które są jedynymi dziećmi.	<div><code><div></code><div><code><p></code></div></div> <div><code><div></code><div><code><p></code><code><p></code></div></div>
<code>p:only-of-type</code>	Odnosi się do elementów, które są jedynym takim typem w rodzicu.	<div><code><div></code><div><code></code><code></code><code></code><code><p></code></div></div> <div><code><div></code><div><code></code><code><p></code><code></code><code><p></code></div></div>
<code>p:not(.a)</code>	Odnosi się do elementów, które nie pasują do warunku podanego w nawiasie. W naszym przykładzie – <code><p></code> bez klasy <code>"a"</code> .	<code><a.a></code> <code><p.a></code> <code><p.b></code> <code><li.a></code>

Wskazówki

- Mamy również: **`nth-last-child`**, **`last-of-type`**, **`nth-last-of-type`**.
Działają one na tej samej zasadzie co **`nth-child`**, **`first-of-type`** oraz **`nth-first-of-type`**, ale zaczynają liczyć od tyłu.
- Zamiast **`p:nth-child(Xn)`** możemy dać np. **`p:nth-child(3)`** – pomijamy więc mnożnik `n`.
Wtedy odniesiemy się tylko i wyłącznie do danego elementu (w naszym przypadku elementu nr 3)
i operacja nie będzie powtarzana na innych elementach.
- Zamiast liczb, możemy wpisać **`odd`** lub **`even`**. Wtedy będziemy odnosić się do parzystych lub nieparzystych elementów.
Np. **`p:nth-child(even)`** lub **`li:nth-of-type(odd)`**.