

### Selektory proste:

selektor	Opis
.klasa	wszystkie znaczniki należące do klasy
.klasa1.klasa2	wszystkie znaczniki należące do obu klas
#id	znacznik o określonym id
*	wszystkie znaczniki
tag	wszystkie znaczniki o danym tagu
tag.klasa	znaczniki o danym tagu należące do danej klasy
tag1 , tag2	wszystkie znaczniki o tagu tag1 i tag2

### Selektory kombinacyjne:

selektor	Opis
spacja	potomek
>	dziecko
+	bezpośredni (bezpośrednio za) węzeł siostrzany za danym elementem
~	węzeł siostrzany PO bieżącym (wszystkie spełniające kryteria)

### Selektory atrybutów:

selektor	Opis
[atrybut]	wszystkie znaczniki zawierające określony atrybut
[atrybut=wartość]	wszystkie znaczniki zawierające daną wartość atrybutu
[atrybut~=słowo]	wszystkie znaczniki zawierające wyraz <b>słowo</b> (przydatne np. w klasach)
[atrybut =word]	wszystkie znaczniki, których dany atrybut <b>rozpoczyna się</b> od wyrazu word (całe słowo lub słowo i "-" )
[atrybut^=wart]	wszystkie znaczniki, których wartość atrybutu rozpoczyna się od ciągu znaków
[atrybut\$=wart]	-  - kończy się ciągiem znaków
[atrybut*=rtos]	-  - zawiera ciąg znaków

pseudo-klasy:	
selektor	Opis
:first-child	element o bieżącej definicji, będący pierwszym dzieckiem rodzica
:first-of-type	wszystkie elementy o bieżącej definicji, będące pierwszym elementem danego typu w dowolnym rodzicu
:last-child	element o bieżącej definicji, będący ostatnim dzieckiem rodzica
:last-of-type	-  - typie
:not (SELEKTOR)	wszystkie elementy niezgodne z definicją w nawiasie
:nth-child(n)	element o bieżącej definicji, będący n'tym dzieckiem rodzica
:nth-last-child(n)	-  - od końca
:nth-of-type(n)	-  - o typie
:only-child	-  - które są jedynym dzieckiem rodzica

selektor	Opis
:has ( . . . )	pseudoklasa umożliwia wyznaczenie elementu poprzedniego w definicji na podstawie posiadanych przez niego własności (np. istnienia elementu wewnątrz niego)