



## 전체 선택자(Universal Selector)

- 전체 선택자(Universal Selector)는 모든 요소를 선택하는 방법

선택자	의미
*	모든 요소

SAMPLE

```
* {  
  margin: 0;  
  padding: 0;  
}
```

## 요소 선택자(Type Selector)

- HTML 요소를 선택하는 방법
- "h1", "img", "ul", "li", "a", "p", "div", "span" 등의 요소를 선택자로 지정

선택자	의미
E	E (Element)

SAMPLE

```
div {  
  color: red;  
}
```

## 클래스 선택자(Class Selector)

중요

- 클래스 선택자(Class Selector)는 HTML 요소의 class 속성 값을 참조하여 선택하는 방법
- class 속성 값은 하나의 HTML 요소에 여러 개를 지정할 수 있으므로, 다중 class도 선택자로 지정 가능

선택자	의미
E.box	class 속성 값이 box 인 E
E.box.con	class 속성 값이 box 와 con 을 모두 가진 E

SAMPLE

```
div.box { background-color: gold; }
div.box.con { background-color: crimson; }
```

## 아이디 선택자(ID Selector)

중요

- 아이디 선택자(ID Selector)는 HTML 요소의 id 속성 값을 참조하여 선택하는 방법
- id 속성 값은 하나의 HTML 문서에 한 번만 사용할 수 있으므로, 유일한 요소 선택시 사용

선택자	의미
E#wrap	id 속성 값이 wrap 인 E

SAMPLE

```
#wrap { width: 1200px; position: relative; }
```

## 후손 선택자(Descendant Combinator)

중요

- 선택자A와 선택자B를 공란(빈칸)으로 선언하며, 선택자A 의 하위(자손/후손) 요소 중 선택자B 에 해당하는 요소를 선택하는 방법

선택자	의미
선택자A 선택자B	"선택자B" 가 반드시 "선택자A" 안에 포함되어 있을 경우에만 선택

SAMPLE

```
#wrap div { background-color: gold; }
```

## 자식 선택자(Child Combinator)

중요

- 부모 요소 선택자A 하위에 포함된 자식 요소 선택자B 를 선택하는 방법
- 부모 선택자와 자식 선택자는 ">"로 구분하여 선언

선택자	의미
선택자A > 선택자B	"선택자B" 가 부모 요소로 "선택자A" 를 가지고 있는 경우에만 선택

SAMPLE

```
#wrap > div { background-color: gold; }
```

중요

## 형제 선택자(Sibling Combinators)

- 기본 형제 선택자(General Sibling Combinators)
  - 선택자A 와 선택자B 를 "+" 로 구분하여 선언
- 인접 형제 선택자(Adjacent Sibling Combinators)
  - 선택자A 와 선택자B 를 "~" 로 구분하여 선언

선택자	의미
선택자A + 선택자B	"선택자A" 바로 뒤(다음) "선택자B" 만 선택
선택자A ~ 선택자B	"선택자A" 뒤(다음)에 인접한 모든 "선택자B" 선택

SAMPLE

```
#box + p { color: crimson; }
#box ~ p { background-color: gold; }
```

## 가상 클래스 선택자(Pseudo-classes Selector)

- 동적 가상 클래스(Dynamic pseudo-classes)
  - :link, :visited, :active, **:hover**, :focus
- 타겟 가상 클래스(Target pseudo-classes)
  - :target
- 언어 가상 클래스(Language pseudo-classes)
  - :lang

## 가상 클래스 선택자(Pseudo-classes Selector)

- UI 요소 상태 가상 클래스(UI element states pseudo-classes)
  - :enabled, :disabled, **:checked**
- 구조 가상 클래스(Structural pseudo-classes)
  - :root, **:nth-child( )**, :nth-last-child( ), :nth-of-type( ), :nth-last-of-type( ), **:first-child**, **:last-child**, :first-of-type( ), :last-of-type( ), :only-child, :only-of-type( ), :empty
- 부정 가상 클래스(Negation pseudo-classes)
  - :not( )

## 가상 클래스 선택자(Pseudo-classes Selector)

선택자	의미	
E:hover	마우스 포인터가 올라간 E 선택	중요
E:link	아직 방문하지 않은 하이퍼링크를 가진 E 선택	
E:visited	이미 방문한 하이퍼링크를 가진 E 선택	
E:active	현재 활성화된 E 선택	
E:focus	키보드 포커스를 받은 E 선택	중요
E:target	하이퍼링크 타겟이 되는 경우에 선택	
E:enabled	사용 가능 상태인 E 선택	
E:disabled	사용 불가 상태인 E 선택	
E:root	문서 최상의 요소 선택	
E:checked	선택(체크)된 E 선택	중요

## 가상 클래스 선택자(Pseudo-classes Selector)

선택자	의미	
E:nth-child(n)	앞에서부터 n 번째 자식 E 선택	중요
E:first-child	첫 번째 자식 요소인 E 선택	중요
E:last-child	마지막 자식 요소인 E 선택	중요
E:only-child	부모 요소의 유일한 자식인 E 선택	
E:nth-last-child(n)	뒤에서부터 n 번째 자식 E 선택	
E:nth-of-type(n)	동일한 E 타입 형제 요소 중 앞에서부터 n 번째 E 선택	
E:nth-last-of-type(n)	동일한 E 타입 형제 요소 중 뒤에서부터 n 번째 E 선택	
E:first-of-type()	부모 요소에 대해 첫 번째 자식의 타입이 E 이면 선택	
E:last-of-type()	부모 요소에 대해 마지막 자식의 타입이 E 이면 선택	

## 가상 클래스 선택자(Pseudo-classes Selector)

선택자	의미
E:empty	텍스트를 포함한 어떤 자식 요소도 가지지 않은 E 선택
E:not(S)	S 로 지정된 선택자와 일치하지 않는 E 선택

SAMPLE

```
#gnb a:hover { color: crimson; }
```

## 가상 요소 선택자(Pseudo-element Selector)

- 요소의 첫 글자나 첫 줄 또는 요소 앞이나 뒤 등 가상의 영역을 선택하고자 할 때 사용

선택자	의미	
E::before	E 콘텐츠 앞 영역에 생성된 가상 요소 선택	중요
E::after	E 콘텐츠 뒤 영역에 생성된 가상 요소 선택	중요
E::first-letter	E 요소의 문자열 중 첫 번째 문자 선택	
E::first-line	E 요소의 문자열 중 첫 번째 줄 선택	
E::selection	사용자가 선택한 E 요소의 범위 선택	

SAMPLE

```
p:after {
  content: "after"; display: block;
}
```

## 속성 선택자(Attribute Selector)

- HTML 요소의 속성을 참조하여 선택하는 방법
- 속성의 지정 여부나 속성 값의 일치 여부로 선택 가능

선택자	의미
E[attr]	attr 속성을 가진 E
E[attr="value"]	attr 속성 값이 "value" 인 E
E[attr^="va"]	attr 속성 값이 "va" 로 시작하는 E
E[attr\$="lue"]	attr 속성 값이 "lue" 로 끝나는 E
E[attr*="al"]	attr 속성 값이 "al" 을 포함하는 E
E[attr~="value"]	attr 속성 값 중 공백으로 분리된 "value" 단어가 있는 E
E[attr = "value"]	attr 속성 값이 "value" 이거나 "value-" 로 시작하는 E



## 속성 선택자(Attribute Selector)

SAMPLE

```
a[title] { text-decoration: underline; }
```

## 선택자의 그룹화

중요

- 모든 선택자는 콤마(,)를 사용하여 한 번에 그룹 선언 가능
- 그룹 선언시 선언된 모든 선택자에는 동일한 선언이 적용 됨

선택자	의미
선택자A , 선택자B	"선택자A" 와 "선택자B" 선택
선택자A, 선택자B, 선택자C	"선택자A" 와 "선택자B" 와 "선택자C" 선택

SAMPLE

```
h2, p, #box { color: crimson; }
```

## 선택자 우선순위

- CSS는 중복 선언을 허용함
  - 단, 하나의 요소에 여러 스타일이 중복 지정될 경우, 선택자의 우선순위에 따라 구체성(specificity)이 높은 선택자의 스타일이 적용됨
- ① 속성값 뒤에 !important 를 붙인 속성
  - ② HTML에서 style을 직접 지정한 속성
  - ③ #id 로 지정한 속성
  - ④ .class 로 지정한 속성
  - ⑤ 요소 이름으로 지정한 속성
  - ⑥ 상속된 속성

## 선택자 우선순위 (명시도 계산법)

- 명시도 계산법

!important > id [ 100 ] > class [ 10 ] > tag [ 1 ] > \* [ 0 ]

- ① ID 선택자의 갯수를 세어서 개당 100 으로 계산
  - ② 클래스 선택자의 갯수를 세어서 개당 10 으로 계산
  - ③ 태그 선택자의 갯수를 세어서 개당 1 로 계산
  - ④ 공용 선택자는 모두 0으로 계산
- 가상 엘리먼트는 무시
  - 같은 우선순위에 있는 경우, 나중에 선언된 것이 우선 적용

# 선택자 우선순위

선택자	A X 100	B X 10	C X 1	구체성(합계)	우선순위
li	0	0	1	1	5
ul li	0	0	2	2	4
div.content	0	1	1	11	3
#nav li	1	0	1	101	2
ul#nav li	1	0	2	102	1



Q & A