

CSS 기본 문법

선택자 { 속성: 값; }

스타일(CSS)을 적용할 대상 (Selector)

선택자 {속성: 값; }

스타일(CSS)의 종류 (Property)

선택자 { 속성: 값; }

스타일(CSS)의 값 (Value)

선택자 { 속성: 값; }

속성은 값이다

선택자 {속성: 값; 속성: 값; }

스타일 범위의 시작

스타일 범위의 끝

글자색

빨강


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

The diagram illustrates the assignment of the color 'red' to the 'color' property in a CSS rule. The text 'div { color: red; }' is shown. A light gray rectangular box highlights the 'color:' part, with a curved arrow pointing from the Korean label '글자색' (text color) above it. Another light gray rectangular box highlights the 'red;' part, with a curved arrow pointing from a red square button labeled '빨강' (red) above it.

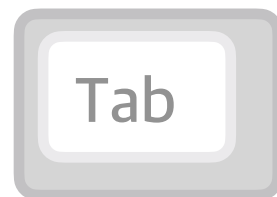
요소 외부 여백

20픽셀

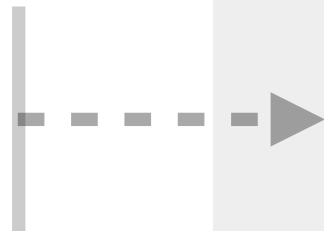
```
div { color: red; margin: 20px; }
```



div {



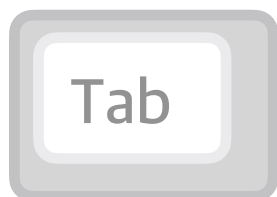
들여쓰기
(Indent)



color: red;
margin: 20px;

VS

내어쓰기
(Outdent)



}

```
/* 설명 작성 */  
div {  
    color: red;  
    margin: 20px;  
}
```

주석 시작

주석 끝

`/* 설명 작성 */`

브라우저는 이 범위를 해석하지 않음



혹은



`div {`

`color: red;`

`margin: 20px;`

`}`

CSS 선언 방식

내장 방식

링크 방식

인라인 방식

@import 방식

```
<style>
  div {
    color: ■ red;
    margin: 20px;
  }
</style>
```

내장 방식

<style></style>의 내용(Contents)으로
스타일을 작성하는 방식

인라인 방식

요소의 style 속성에 직접 스타일을 작성하는 방식
(선택자 없음)

```
<div style="color: ■ red; margin: 20px;"></div>
```

```
<link rel="stylesheet" href="./css/main.css">
```

링크 방식

<link /> 로 외부 CSS 문서를
가져와서 연결하는 방식

main.css

```
div {  
  color: ■ red;  
  margin: 20px;  
}
```

```
<link rel="stylesheet" href="./css/main.css">
```

main.css

box.css

```
@import url("./box.css");
```

```
div {  
  color: ■ red;  
  margin: 20px;  
}
```

```
.box {  
  background-color: ■ red;  
  padding: 20px;  
}
```

@import 방식

CSS의 @import 규칙으로 CSS 문서 안에서
또 다른 CSS 문서를 가져와 연결하는 방식

CSS 선택자

기본

복합

가상 클래스

가상 요소

속성

*

기본

전체 선택자 (Universal Selector)

모든 요소를 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li>오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span>오렌지</span>
</div>
```

선택

*

```
{
  color: ■ red;
}
```

ABC

기본

태그 선택자 (Type Selector)

태그 이름이 ABC인 요소 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li>오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span>오렌지</span>
</div>
```

선택

li {

color: ■ red;

}

.ABC

기본

클래스 선택자 (Class Selector)

HTML class 속성의 값이 ABC인 요소 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
```

선택

```
.orange {
  color: ■ red;
}
```


#ABC

기본


아이디 선택자 (ID Selector)

HTML id 속성의 값이 ABC인 요소 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li id="orange" class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
```

선택

```
#orange {
  color: ■ red;
}
```

A curved arrow points from the '#orange' selector in the CSS block to the '<li id="orange" class="orange">오렌지' element in the HTML block. A small grey box with the Korean word '선택' (select) is placed on the arrow.

ABCXYZ

복합

일치 선택자 (Basic Combinator)

선택자 ABC와 XYZ를 동시에 만족하는 요소 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
```

선택

```
span.orange {
  color: ■ red;
}
```

ABC > XYZ

복합

자식 선택자 (Child Combinator)

선택자 ABC의 자식 요소 XYZ 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
```

선택

```
ul > .orange {
  color: ■ red;
}
```


ABC XYZ

복합

하위(후손) 선택자 (Descendant Combinator)

선택자 ABC의 하위 요소 XYZ 선택.
'띄어쓰기'가 선택자의 기호!

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
<span class="orange">오렌지</span>
```

선택

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

ABC + XYZ

복합

인접 형제 선택자 (Adjacent Sibling Combinator)

선택자 ABC의 다음 형제 요소 XYZ 하나를 선택.

```
.orange + li {  
  color: ■ red;  
}
```

선택

```
<ul>  
  <li>딸기</li>  
  <li>수박</li>  
  <li class="orange">오렌지</li>  
  <li>망고</li>  
  <li>사과</li>  
</ul>
```


ABC ~ XYZ

복합

일반 형제 선택자 (General Sibling Combinator)

선택자 ABC의 다음 형제 요소 XYZ 모두를 선택.

```
.orange ~ li {  
  color: ■ red;  
}
```

선택

```
<ul>  
  <li>딸기</li>  
  <li>수박</li>  
  <li class="orange">오렌지</li>  
  <li>망고</li>  
  <li>사과</li>  
</ul>
```

ABC: hover

가상 클래스 선택자 (Pseudo-Classes)

HOVER

선택자 ABC 요소에 마우스 커서가 올라가 있는 동안 선택.

화면에 출력!

NAVER

NAVER

선택

```
a:hover {  
  color: ■ red;  
}
```

```
<a href="https://www.naver.com">NAVER</a>
```

ABC:active

가상 클래스 선택자 (Pseudo-Classes)

ACTIVE

선택자 ABC 요소에 마우스를 클릭하고 있는 동안 선택.

화면에 출력!

NAVER

NAVER

선택

```
a:active {  
  color: ■ red;  
}
```

```
<a href="https://www.naver.com">NAVER</a>
```

ABC:focus

가상 클래스 선택자 (Pseudo-Classes)

FOCUS

선택자 ABC 요소가 포커스되면 선택.

화면에 출력!

선택

```
input:focus {  
  background-color: orange;  
}
```

```
<input type="text" />
```

ABC:first-child

가상 클래스 선택자 (Pseudo-Classes)

FIRST CHILD

선택자 ABC가 형제 요소 중 첫째라면 선택.

선택

```
.fruits span:first-child {  
  color: ■ red;  
}
```



```
<div class="fruits">  
  <span>딸기</span>  
  <span>수박</span>  
  <div>오렌지</div>  
  <p>망고</p>  
  <h3>사과</h3>  
</div>
```


ABC:first-child

가상 클래스 선택자 (Pseudo-Classes)

FIRST CHILD

선택자 ABC가 형제 요소 중 첫째라면 선택.

?  `.fruits div:first-child {
 color:  red;
}`

```
<div class="fruits">  
  <span>딸기</span>  
  <span>수박</span>  
  <div>오렌지</div>  
  <p>망고</p>  
  <h3>사과</h3>  
</div>
```

ABC:last-child

가상 클래스 선택자 (Pseudo-Classes)

LAST CHILD

선택자 ABC가 형제 요소 중 막내라면 선택.

```
<div class="fruits">
  <span>딸기</span>
  <span>수박</span>
  <div>오렌지</div>
  <p>망고</p>
  <h3>사과</h3>
</div>
```

선택

```
.fruits h3:last-child {
  color: ■ red;
}
```

ABC:nth-child(n)

가상 클래스 선택자 (Pseudo-Classes)

NTH CHILD

선택자 ABC가 형제 요소 중 (n)째라면 선택.

```
.fruits *:nth-child(2) {  
  color: ■ red;  
}
```

선택

```
<div class="fruits">  
  <span>딸기</span>  
  <span>수박</span>  
  <div>오렌지</div>  
  <p>망고</p>  
  <h3>사과</h3>  
</div>
```


ABC:nth-child(n)

가상 클래스 선택자 (Pseudo-Classes)

NTH CHILD

선택자 ABC가 형제 요소 중 (n)째라면 선택.

```
<div class="fruits">
  <span>딸기</span>
  <span>수박</span>
  <div>오렌지</div>
  <p>망고</p>
  <h3>사과</h3>
</div>
```

선택

```
.fruits *:nth-child(2n) {
  color: ■ red;
}
```

n은 0부터 시작!
(Zero-Based Numbering)

ABC:nth-child(n)

가상 클래스 선택자 (Pseudo-Classes)

NTH CHILD

선택자 ABC가 형제 요소 중 (n)째라면 선택.

```
<div class="fruits">
  <span>딸기</span>
  <span>수박</span>
  <div>오렌지</div>
  <p>망고</p>
  <h3>사과</h3>
</div>
```

선택

```
.fruits *:nth-child(2n+1) {
  color: ■ red;
}
```

n은 0부터 시작!
(Zero-Based Numbering)

ABC:nth-child(n)

가상 클래스 선택자 (Pseudo-Classes)

NTH CHILD

선택자 ABC가 형제 요소 중 (n)째라면 선택.

```
<div class="fruits">  
  <span>딸기</span>  
  <span>수박</span>  
  <div>오렌지</div>  
  <p>망고</p>  
  <h3>사과</h3>  
</div>
```

선택

```
.fruits *:nth-child(n+2) {  
  color: ■ red;  
}
```

n은 0부터 시작!
(Zero-Based Numbering)

ABC:not(XYZ)

부정 선택자 (Negation)

NOT

선택자 XYZ가 아닌 ABC 요소 선택.

선택

```
.fruits *:not(span) {  
  color: ■ red;  
}
```

```
<div class="fruits">  
  <span>딸기</span>  
  <span>수박</span>  
  <div>오렌지</div>  
  <p>망고</p>  
  <h3>사과</h3>  
</div>
```

ABC::before

인라인(글자) 요소

화면에 출력!

앞! Content!

```
<div class="box">
```

Content!

```
</div>
```

가상 요소 선택자 (Pseudo-Elements)

BEFORE

선택자 ABC 요소의 내부 앞에 내용(Content)을 삽입.

```
.box::before {  
  content: "앞!";  
}
```

::before

ABC::after

인라인(글자) 요소

가상 요소 선택자 (Pseudo-Elements)

AFTER

선택자 ABC 요소의 내부 뒤에 내용(Content)을 삽입.

화면에 출력!

Content! 뒤!

```
<div class="box">
```

Content!

```
</div>
```

::after

```
.box::after {  
  content: "뒤!";  
}
```

[ABC]

속성 선택자 (Attribute)

ATTR

속성 ABC를 포함한 요소 선택

```
[disabled] {  
  color: ■ red;  
}
```

선택

```
<input type="text" value="HEROPY">  
<input type="password" value="1234">  
<input type="text" value="ABCD" disabled>
```


[ABC]

속성 선택자 (Attribute)

ATTR

속성 ABC를 포함한 요소 선택

선택

```
[type] {  
  color: ■ red;  
}
```

```
<input type="text" value="HEROPY">  
<input type="password" value="1234">  
<input type="text" value="ABCD" disabled>
```


[ABC="XYZ"]

속성 선택자 (Attribute)

ATTR=VALUE

속성 ABC를 포함하고 값이 XYZ인 요소 선택.

```
[type="password"] {  
  color: ■ red;  
}
```

선택

```
<input type="text" value="HEROPY">  
<input type="password" value="1234">  
<input type="text" value="ABCD" disabled>
```

스타일 상속

```
.animal {  
  color: ■ red;  
}
```

선택

```
<div class="ecosystem">생태계  
  <div class="animal">동물  
    <div class="tiger">호랑이</div>  
    <div class="lion">사자</div>  
    <div class="elephant">코끼리</div>  
  </div>  
  <div class="plant">식물</div>  
</div>
```

화면에 출력!

생태계
동물
호랑이
사자
코끼리
식물

상속되는 CSS 속성들..

모두 글자/문자 관련 속성들!

(모든 글자/문자 속성은 아님 주의!)

font-style : 글자 기울기

font-weight : 글자 두께

font-size : 글자 크기

line-height : 줄 높이

font-family : 폰트(서체)

color : 글자 색상

text-align : 정렬

...

강제 상속

선택자 우선순위

**우선순위란, 같은 요소가 여러 선언의 대상이 된 경우,
어떤 선언의 CSS 속성을 우선 적용할지 결정하는 방법**

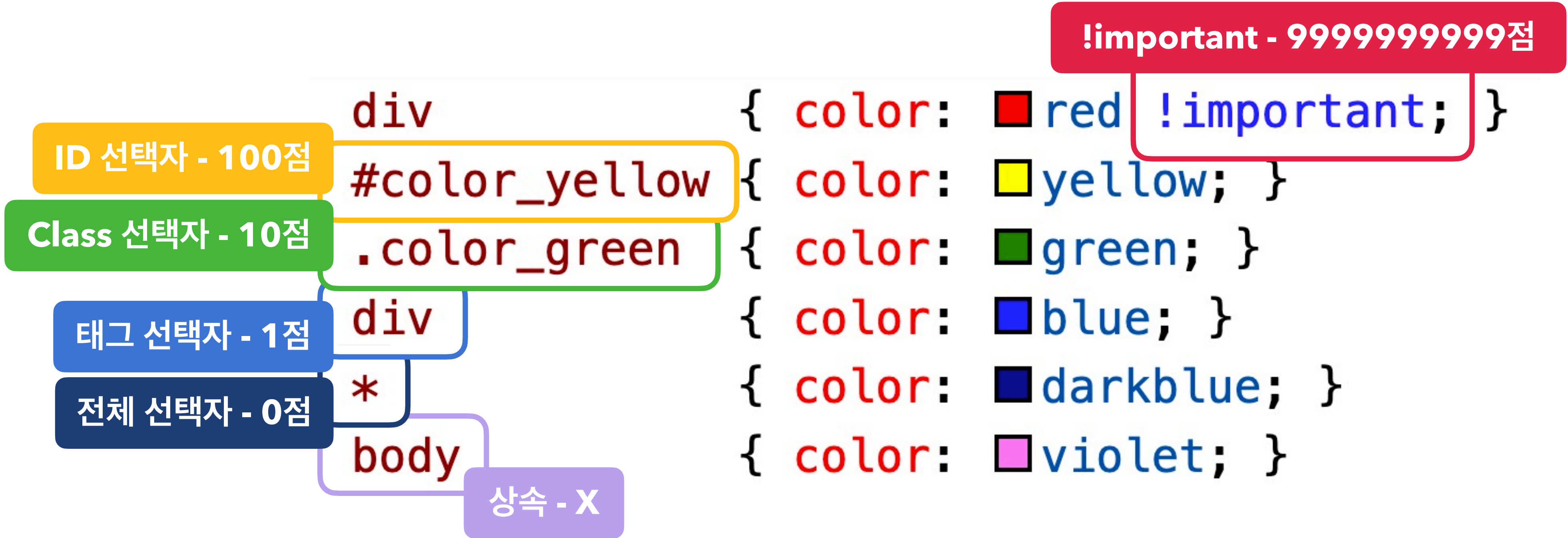
- 1, 점수가 높은 선언이 우선함!**
- 2, 점수가 같으면, 가장 마지막에 해석된 선언이 우선함!**

div	{ color: ■ red !important; }
#color_yellow	{ color: ■ yellow; }
.color_green	{ color: ■ green; }
div	{ color: ■ blue; }
*	{ color: ■ darkblue; }
body	{ color: ■ violet; }

선택

```
<div
  id="color_yellow"
  class="color_green"
  style="color: ■ orange;">
  Hello world!
</div>
```

과연 글자색은??



```
<div
  id="color_yellow"
  class="color_green"
  style="color: orange;">
  Hello world!
</div>
```

인라인 선언 - 1000점

21점 `.list li.item { color: red; }`

21점 `.list li:hover { color: red; }`

11점 `.box::before { content: "Good "; color: red; }`

101점 `#submit span { color: red; }`

22점 `header .menu li:nth-child(2) { color: red; }`

1점 `h1 { color: red; }`

10점 `:not(.box) { color: red; }`