



Seoul
Software
ACademy

웹 개발자 부트캠프 과정

SeSAC x CODINGOn

With. 팀 뽀빠

CSS





귀여운 — 형용사



미니언이 — 명사



춤춘다 — 동사



CSS

CSS



Cascading Style Sheet

웹 페이지를 디자인하기 위해 사용하는 언어

CSS 사용 이유?



Apple에 CSS가 없다면?

우리의 친구! 개발자 도구

- 브라우저에서 개발 편의를 위해 제공하는 도구
- 지금 보고 있는 웹페이지의 HTML, CSS, JavaScript 등을 볼 수 있습니다
- 해당 코드를 수정하여 자기 마음대로 변경이 가능!
- HTML 구조 및 CSS 값 등을 확인 가능하므로 자주 사용 예정
- 특정 요소 코드 보기 / 크기 줄이기 / 모바일 버전 확인 가능

CSS를 지워보자?!



CSS 참조 방식

- 인라인 방식
- 내장 방식
- 링크 방식

인라인 방식은 불편해요!!

```
<h1 style="color: ■ blue; background-color: ■ pink">Hello CSS!</h1>
```

- 각 요소의 style 속성에 전부 스타일을 직접 적어줘야 함
- 아래에서 같은 스타일을 가진 태그를 사용하려고 해도 코드를 복붙 필요
- 즉, 재사용이 전혀 불가능!

내장 방식

```
<style>
  h1 {
    color: red;
    background-color: skyblue;
  }
</style>
```

- HTML의 <head> 태그 내부에 사용할 Style을 <style> 태그로 미리 선언하여 사용!
- 태그에 같은 스타일 적용이 가능!!
- 그런데 다른 HTML 파일에는 어떻게 쓰죠?
- 혹시나 스타일이 바뀌면 모두 바뀌줘야 하나요?

링크 방식

- 모든 Style을 하나의 CSS 파일에 넣고, 필요한 HTML 파일에서 해당 파일을 링크해서 사용하는 방식
- 재사용? OK
- 스타일 변경 발생? OK

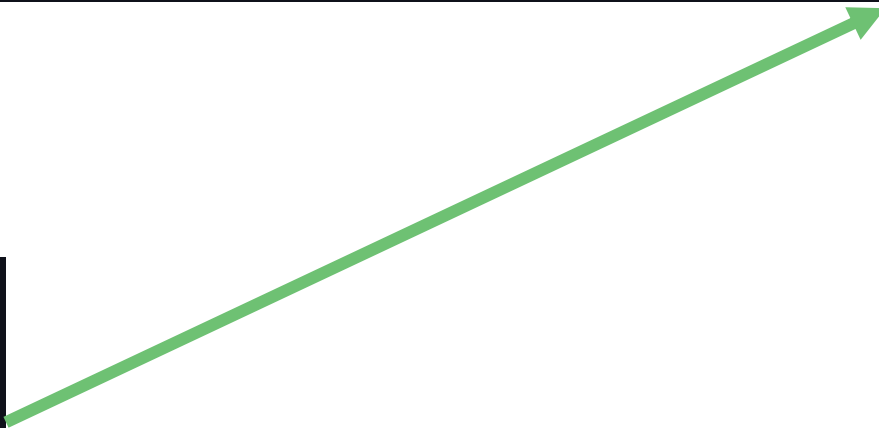
링크 방식

index.html

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

index.css

```
h1 {
  color: red;
  background-color: skyblue;
}
```



CSS 참조 방식



- 그런데 3가지 방식이 겹치면 어떻게 되죠?
- 기본 룰은 가장 늦게 읽히는 것이 우선으로 적용 됩니다!
- 인라인 방식은 내장, 링크 방식에 무조건 우선!
- 내장, 링크 방식은 늦게 쓰여진 것이 우선!

Quiz. 내장 방식을 이용할 때
〈style〉 태그는 어느 곳에 사용될까요?

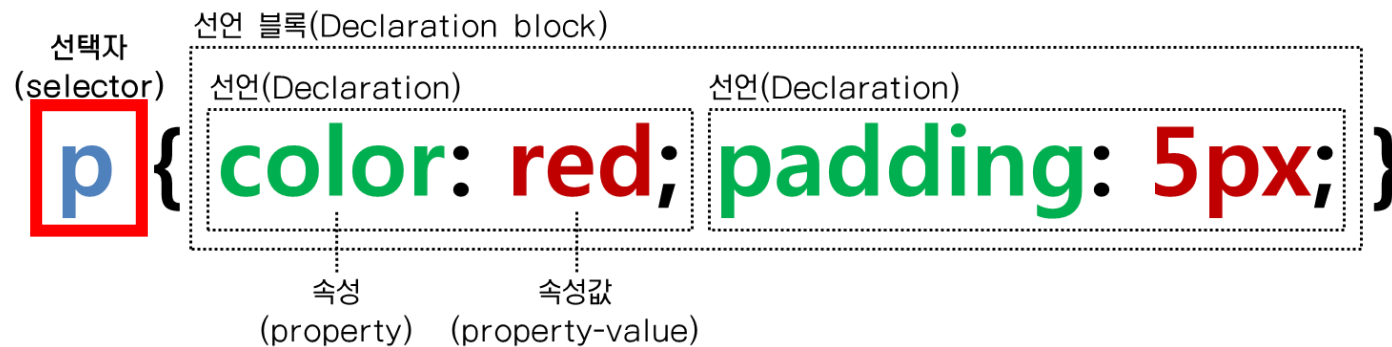
1. 〈html〉
2. 〈head〉
3. 〈body〉
4. 기타

Quiz. 내장 방식을 이용할 때
〈style〉 태그는 어느 곳에 사용될까요?

1. 〈html〉
2. 〈head〉
3. 〈body〉
4. 기타

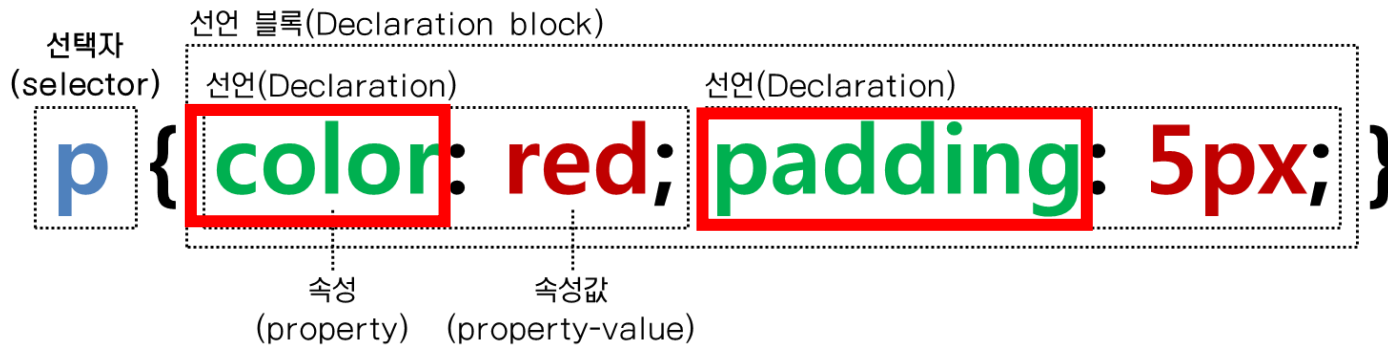
Selector, 선택자

CSS, 선택자(Selector)



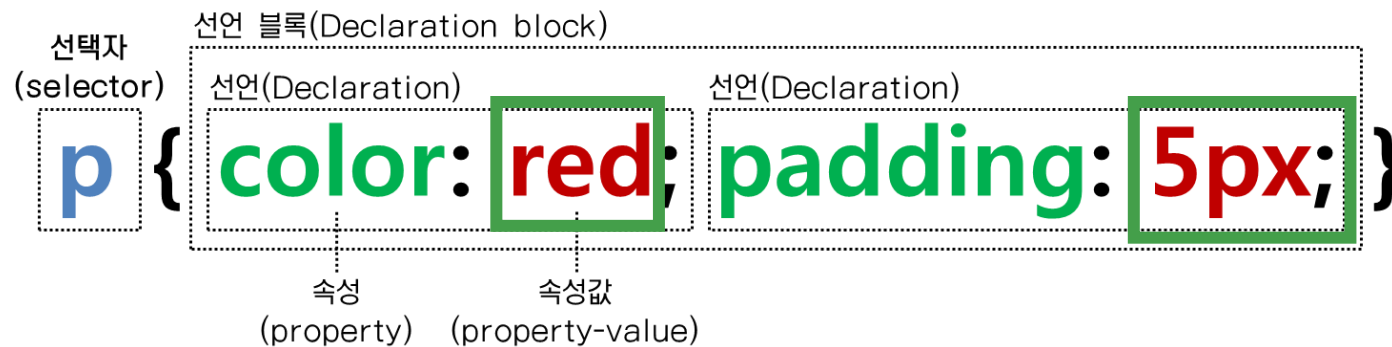
- CSS는 HTML 요소의 style을 정의하여야 하므로 스타일을 적용하고자 하는 HTML 요소를 선택할 수 있어야 한다
- 선택자는 스타일을 적용하고자 하는 HTML 요소를 선택하기 위해 CSS에서 제공하는 수단

CSS, 속성(Property)



- 선택자로 HTML 요소를 선택하고 { } 내에 속성 값을 지정하여 다양한 style을 정의
- 여러 개의 프로퍼티를 연속해서 지정할 수 있으며 세미콜론(;)으로 구분
- 속성은 표준 스펙으로 이미 지정되어 있는 것을 사용, 사용자가 임의로 정의할 수 없다

CSS, 값(Value)



- 값은 해당 속성에 사용할 수 있는 값을 키워드나 크기 단위 또는 색상 단위 등의 특정 단위로 지정

CSS 선택자 – 단일 선택자

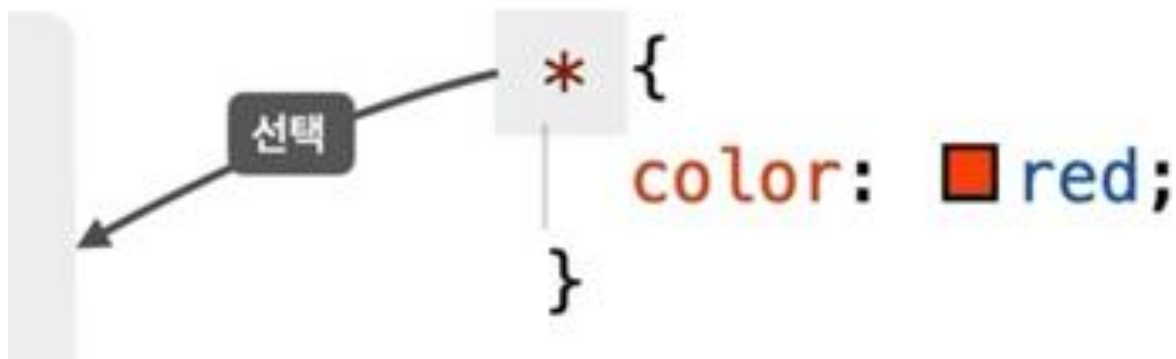
- 전체 선택자
- 태그 선택자
- Class 선택자
- ID 선택자

*

기본

전체 선택자 (Universal Selector)

모든 요소를 선택.



CSS 선택자 – 단일 선택자

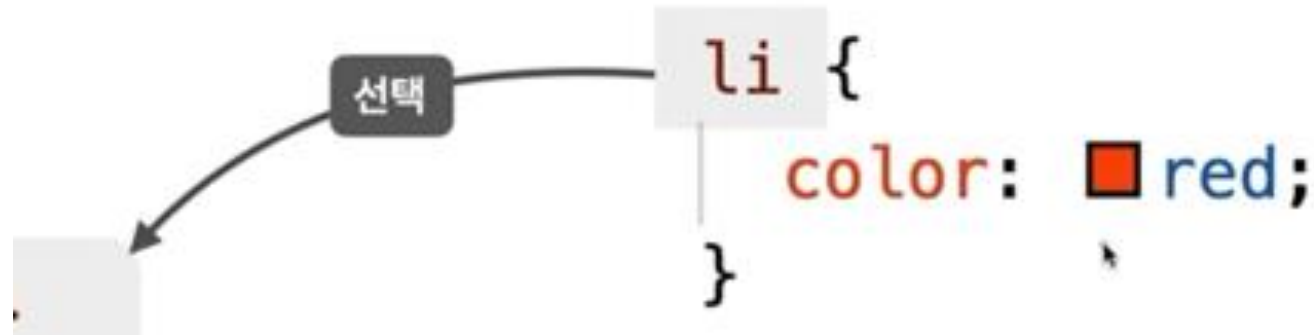
- 전체 선택자
- 태그 선택자
- Class 선택자
- ID 선택자

ABC

기본

태그 선택자 (Type Selector)

태그 이름이 ABC인 요소 선택.



CSS 선택자 – 단일 선택자

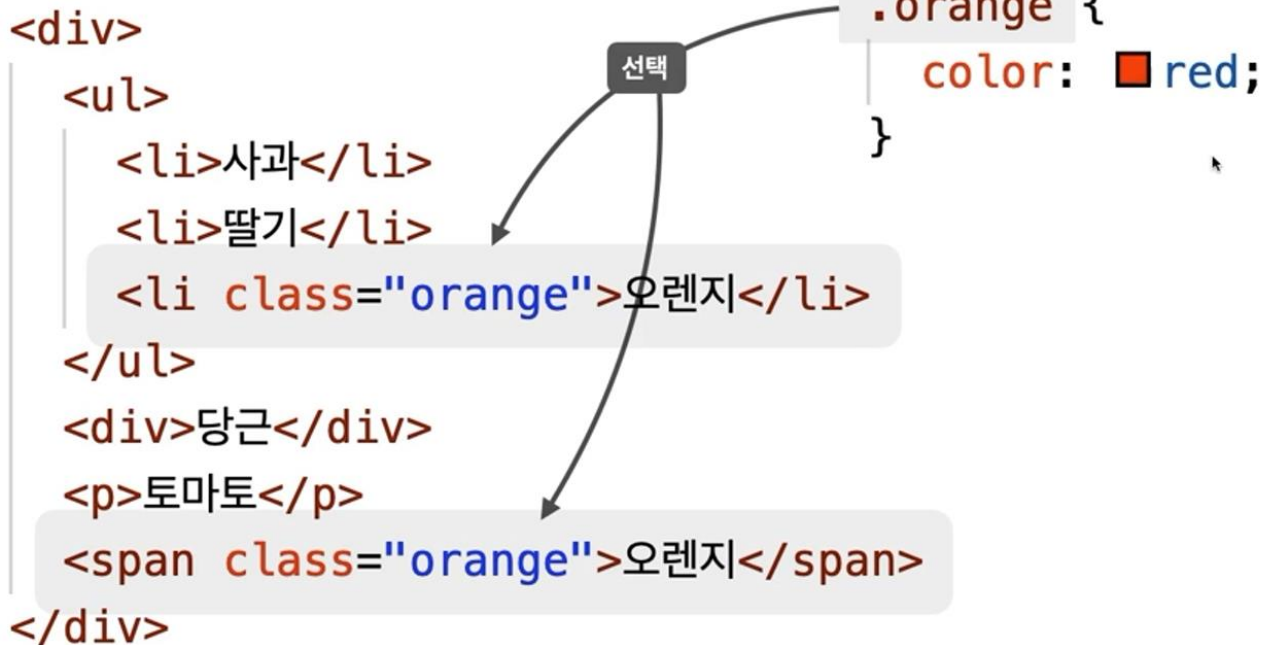
- 전체 선택자
- 태그 선택자
- Class 선택자
- ID 선택자

.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>
```

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

CSS 선택자 – 단일 선택자

- 전체 선택자
- 태그 선택자
- Class 선택자
- ID 선택자

#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>

```

#orange {
color: ■ red;
}

선택

CSS 선택자 – 복합 선택자

- 특수한 요소를 호출하고 싶을 때, 기본 선택자만으로는 선택이 불가능한 경우
에 사용

CSS 선택자 – 복합 선택자

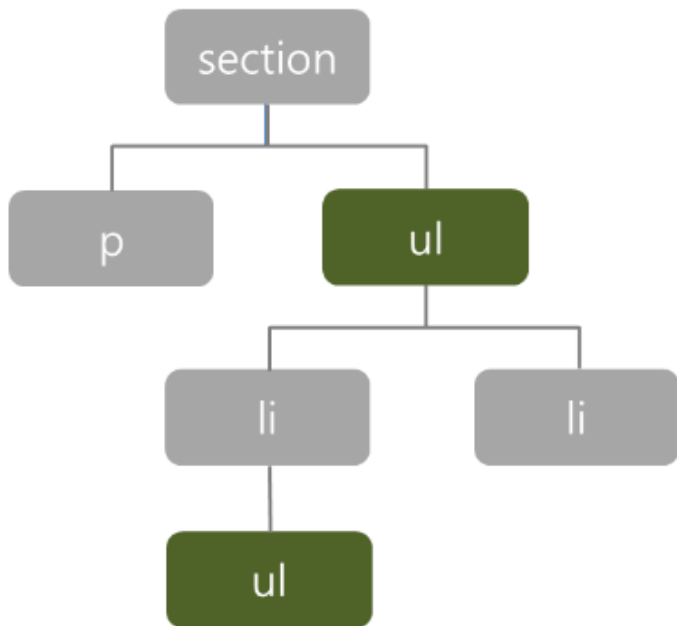
• 하위 선택자

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>
  
```

```

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

선택

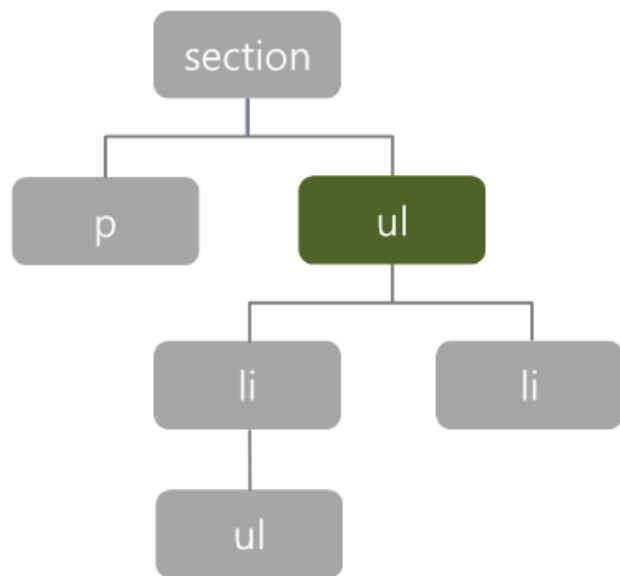
CSS 선택자 – 복합 선택자

• 자식 선택자 (>)

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>
  
```

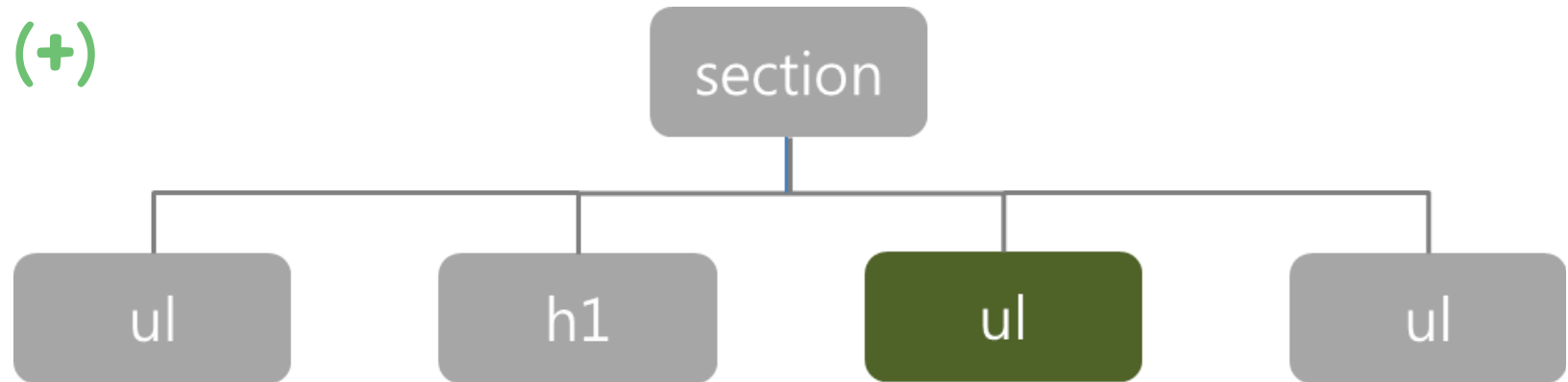
선택

```

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

CSS 선택자 – 복합 선택자

- 인접 형제 선택자 (+)

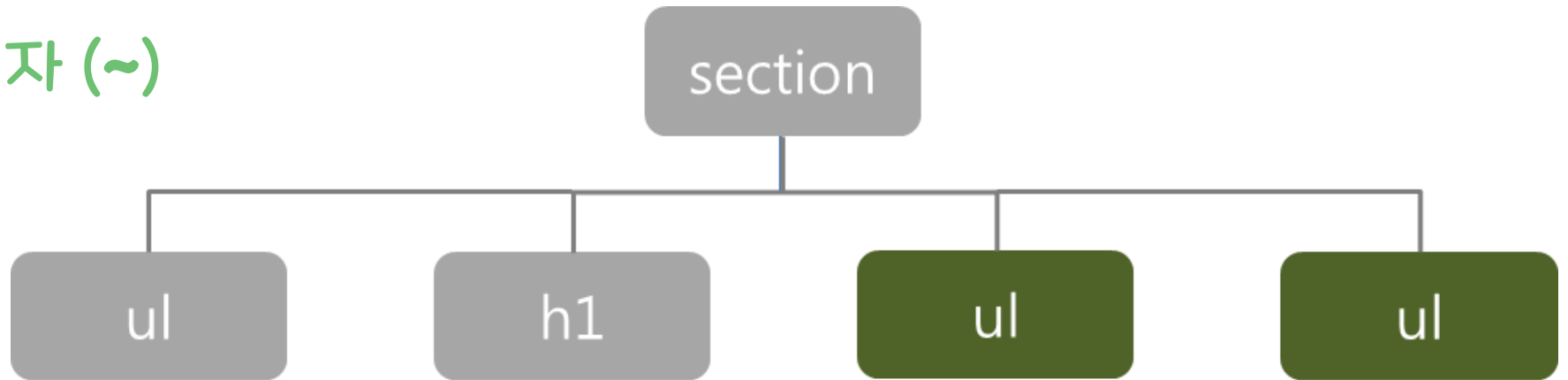


h1 다음 형제 중 ul 하나 선택

`h1 + ul { }`

CSS 선택자 – 복합 선택자

- 일반 형제 선택자 (~)



h1 다음 형제 중 ul 모두 선택

`h1 ~ ul { }`

CSS 선택자 – 가상 클래스 선택자

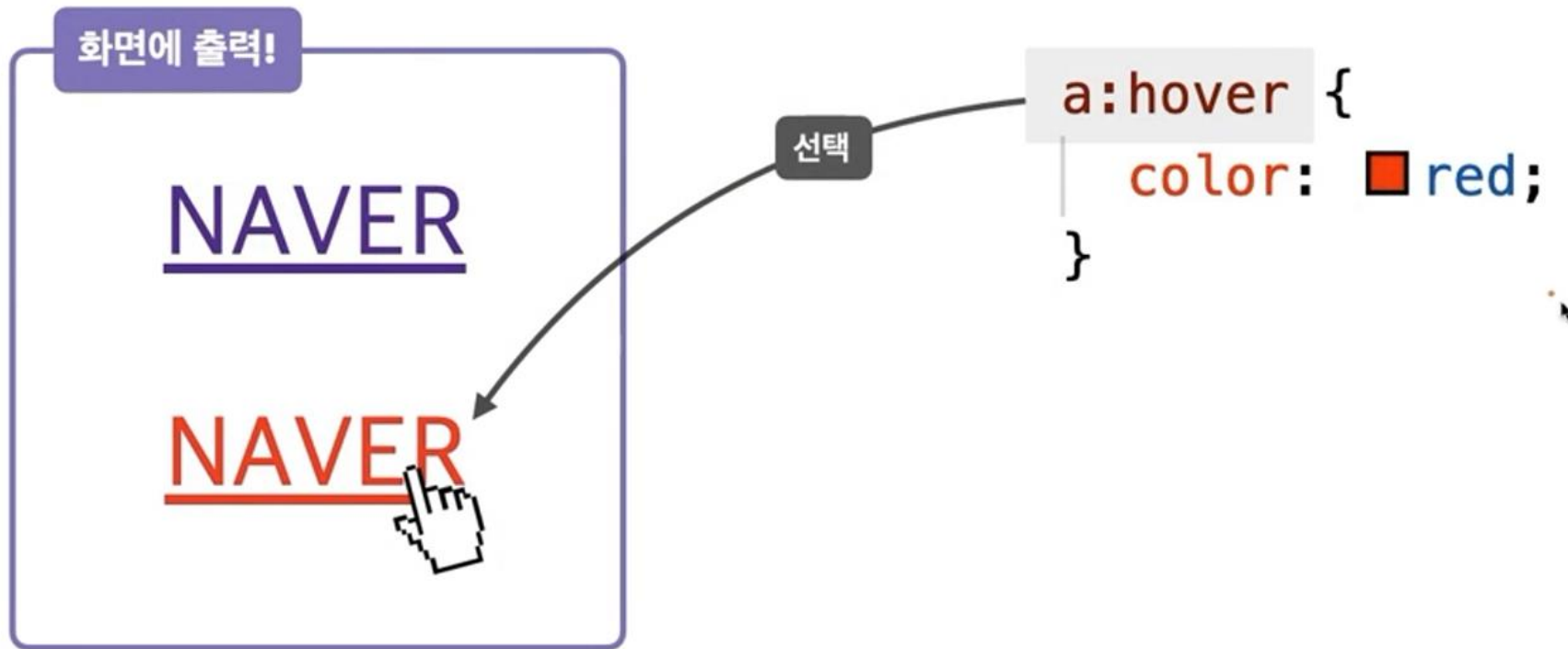
- 사용자의 행동에 따라 변화하는 가상 상황에 따라서 요소 선택 시
- 각 요소의 상황에 따라 사용자가 원하는 요소를 선택 할 때 사용
- 특정 요소를 부정 할 때 사용
- 종류
 - 가상 클래스 선택자
 - 사용자의 행동에 따라 변화 : `hover`, `active`, `focus`, `visited`
 - 요소의 상황 : `first-child`, `last-child`, `nth-child`, ...
 - 부정 선택 : `not`

ABC: hover

가상 클래스 선택자 (Pseudo-Classes)

HOVER

선택자 ABC 요소에 마우스 커서가 올라가 있는 동안 선택.



```
<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" />
```

CSS 선택자 – 가상클래스 선택자

- 가상 선택자

ABC:first-child

가상 클래스 선택자 (Pseudo-Classes)

FIRST CHILD

선택자 ABC가 형제 요소 중 첫째라면 선택.

E::first-child { }

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

선택

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

?

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

CSS 선택자 – 가상클래스 선택자

- 가상 선택자

ABC:last-child

가상 클래스 선택자 (Pseudo-Classes)

LAST CHILD

선택자 ABC가 형제 요소 중 막내라면 선택.

E::last-child { }

```
<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;
}
```

CSS 선택자 – 가상클래스 선택자

- 가상 선택자

ABC:nth-child(n)

가상 클래스 선택자 (Pseudo-Classes)

NTH CHILD

선택자 ABC가 형제 요소 중 (n)째라면 선택.

E::nth-child(n) { }

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

선택

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

CSS 선택자 – 가상클래스 선택자

• 가상 선택자

ABC:not(XYZ)

부정 선택자 (Negation)

NOT

선택자 XYZ가 아닌 ABC 요소 선택.

E::not(XYZ) { }

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

선택

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

CSS 선택자 – 가상 요소 선택자

- 선택 된 요소의 앞, 뒤에 별도의 Content 를 삽입하는 선택자
- 반드시 content 라는 속성을 사용
 - 빈 값(“”) 이라도 넣어 주어야 적용이 됨
- 종류
 - after : 요소의 뒤에 내용 삽입
 - before : 요소의 앞에 내용 삽입

ABC::before

인라인(글자) 요소

가상 요소 선택자 (Pseudo-Elements)

BEFORE

선택자 ABC 요소의 내부 앞에 내용(Content)을 삽입.

화면에 출력!

앞! Content!

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

::before

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

Content!

```
</div>
```

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>
```

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

::after

CSS 선택자 – 가상 요소 선택자

- 실제로 의미 없는 HTML 태그를 만들지 않고 요소 삽입이 가능하여 자주 사용!
- 예를 들어 쇼핑몰 페이지에 메뉴에 Hot, 추천 등을 넣기 위해 별도의 태그를 삽입 하는 것이 아니라 가상 요소 선택자를 활용하여 처리하면 편리함!

CSS 우선순위

- 동일한 요소를 다른 선택자로 적용해 여러 스타일이 적용되는 난감한 상황이 발생하기도 함.
- CSS 파일이 여러 개라면 더욱 난감해짐

!important > 인라인 스타일 > 아이디 선택자
> 클래스/속성/가상 선택자 > 태그 선택자 > 전체 선택자

속성 선택자

- 지정한 **특정 속성**을 가지고 있는 태그를 선택하는 선택자
- 종류
 - 특정 속성만 지정
 - 속성과 속성의 값을 지정

[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="XYZ"]

속성 선택자 (Attribute)

ATTR=VALUE

속성 ABC를 포함하고 값이 XYZ인 요소 선택.

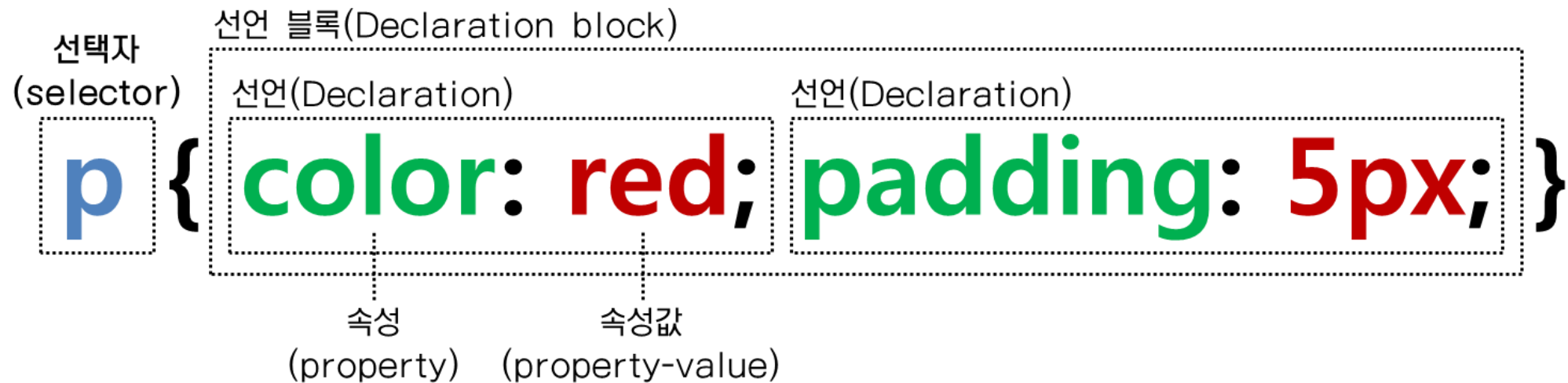
```
[type="password"] {  
  color: ■ red;  
}
```

선택

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

자주 쓰이는 CSS 속성

CSS 작성법



선택자 { 프로퍼티: 값; }

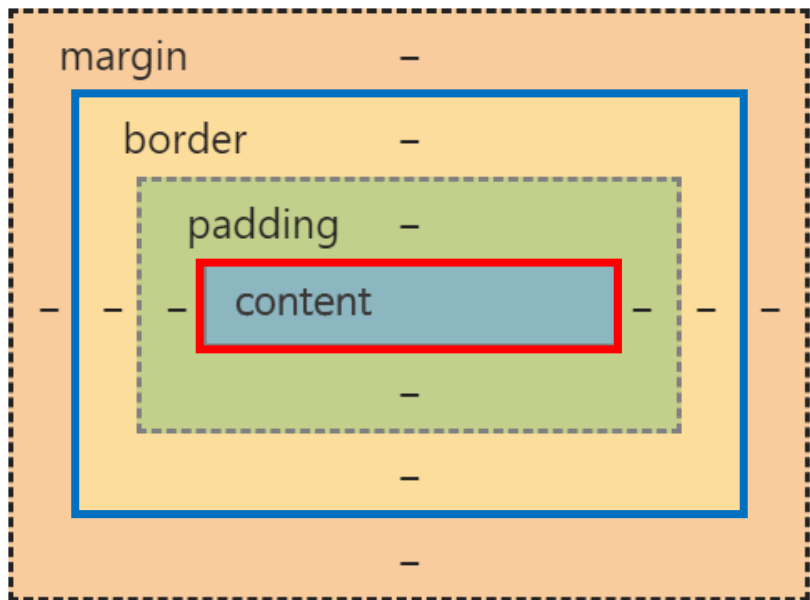
= 선택자의 프로퍼티를 값으로 변경하겠음!

자주 쓰이는 CSS 속성

- color, background-color
- font-size, font-weight, font-family, font-style
- width, height, max-width, min-width, max-height, min-height
- border
- margin, padding

CSS 박스 모델

- **box-sizing** 속성



1. **content-box**

: 콘텐츠를 영역을 기준으로 크기를 정함 (기본값)
(only **content**)

2. **border-box**

: 테두리를 기준으로 크기를 정함
(**content + padding + border**)

개발 초기 단계에서 두 속성 값 중 하나로 결정하는 것이 좋음!
그렇지 않으면 요소 크기 계산이 진짜 복잡해진다... 😞

CSS 여백

- margin, padding

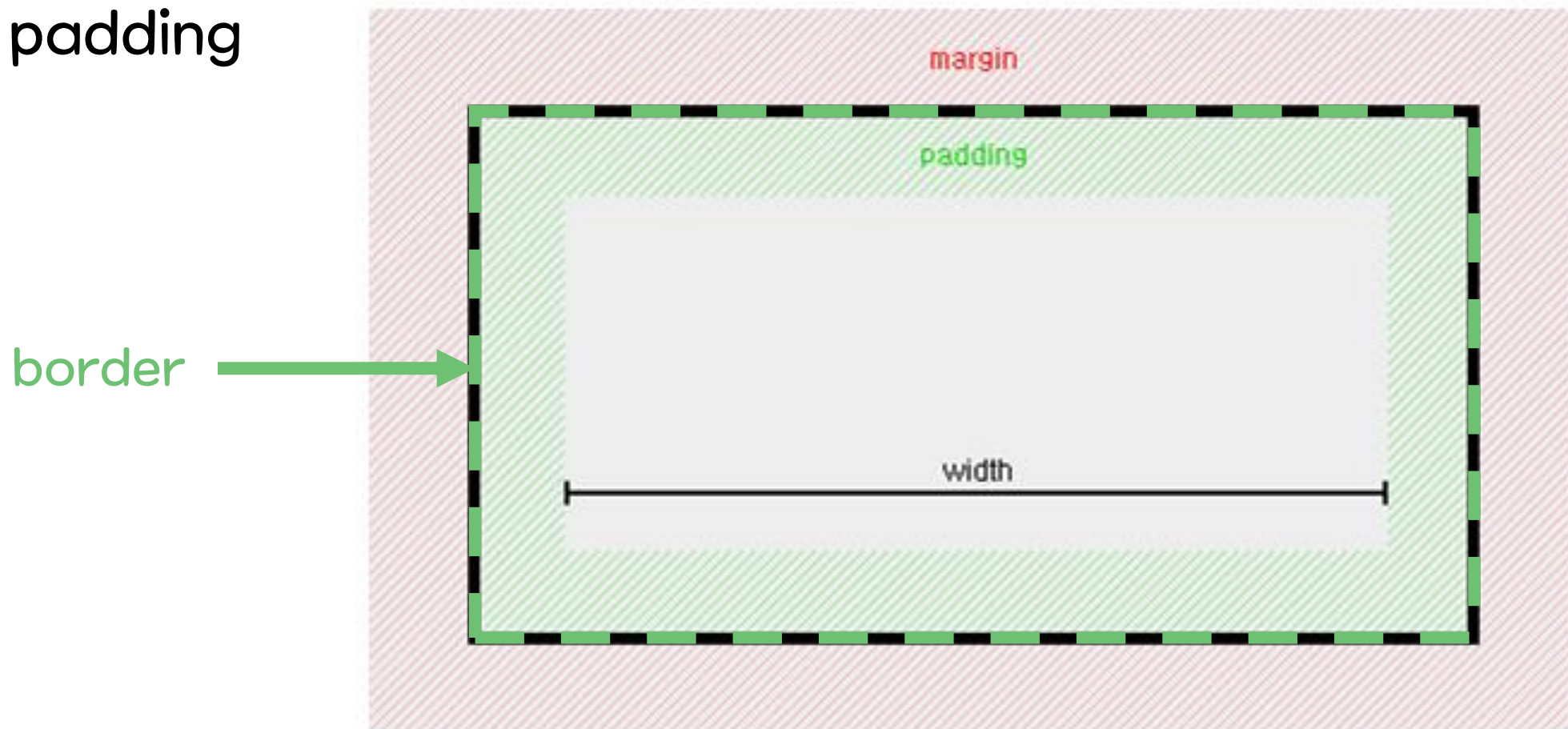


사진 출처: <https://ofcourse.kr/css-course/margin-padding-%EC%86%8D%EC%84%B1>