

K-Digital Training KDT 풀스택 웹 개발자 양성 부트캠프 3기

# CSS

WITH 팀 리처드



# CSS





귀여운 — 형용사



미니언이 — 명사



춤춘다 — 동사



# CSS

CSS



## Cascading Style Sheet

웹 페이지를 디자인하기 위해 사용하는 언어

# CSS 사용 이유?



# CSS 사용 이유?

[뉴스스탠드 바로가기](#) [주제별캐스트 바로가기](#) [타임스퀘어 바로가기](#) [쇼핑캐스트 바로가기](#) [로그인 바로가기](#)

## 네이버

[네이버를 시작페이지로](#) [주니어네이버](#) [해피빈](#)

검색

통합검색

검색어를 입력해 주세요.

검색

[한글 입력기](#)

[자동완성 레이어](#)

- [메일](#)
- [카페](#)
- [블로그](#)
- [지식iN](#)
- [쇼핑](#)
- [쇼핑LIVE](#)
- [Pay](#)
- [TV](#)
- [사전](#)
- [뉴스](#)
- [증권](#)
- [부동산](#)
- [지도](#)
- [VIBE](#)
- [책](#)
- [웹툰](#)

[더보기](#)

[서비스 전체보기](#) [메뉴설정](#)

ㄱ-ㅁ

- [구독](#)
- [날씨](#)
- [네이버 MY CAR](#)
- [네이버 게임](#)
- [네이버 예약](#)
- [네이버 프리미엄콘텐츠](#)
- [네이버플러스 멤버십](#)
- [뉴스](#)
- [라이프](#)

# CSS 참조 방식

- 외부 스타일 시트 ( External Style Sheet )
- 내부 스타일 시트 ( Internal Style Sheet )
- HTML 태그 내의 스타일 지정 ( Inline Styles )

# CSS 참조 방식

## • 외부 스타일 시트 ( External Style Sheet )

1. .css 확장자를 가진 스타일 시트 파일을 생성
2. HTML 문서에 연결해 사용



# CSS 참조 방식

## • 내부 스타일 시트 ( Internal Style Sheet )

1. HTML 문서 내에서 <head> 태그 내부에 <style> 태그를  
이용해 CSS 정의하는 방식

# CSS 참조 방식

## • HTML 태그 내의 스타일 지정 ( Inline Styles )

1. HTML 태그 안에 **style 속성**을 이용해 정의하는 방법

# CSS 참조 방식 우선순위



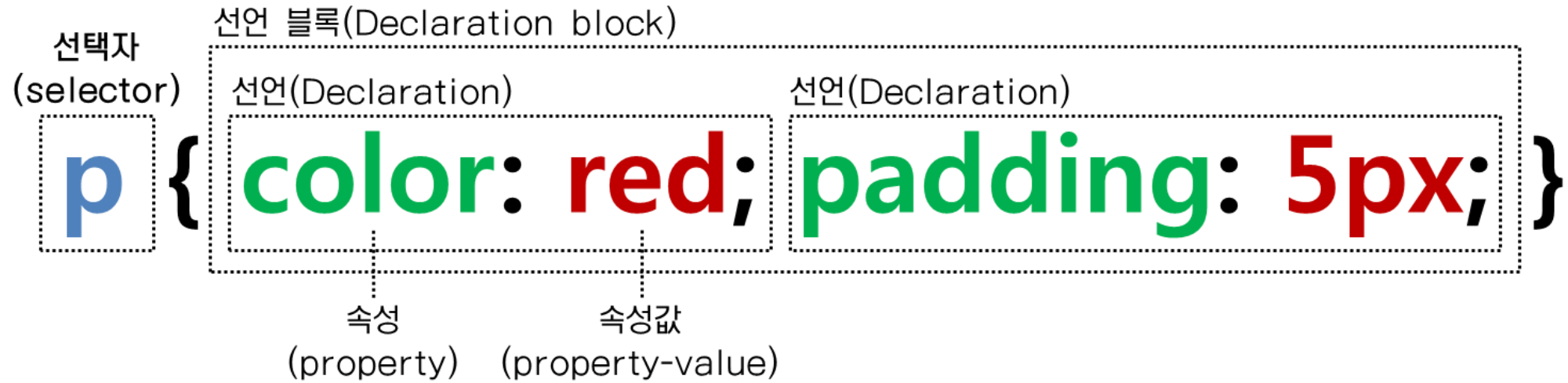
**Quiz.** 내장 방식을 이용할 때  
〈style〉 태그는 어느 곳에 사용될까요?

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

# CSS 작성법



**선택자 { 프로퍼티: 값; }**

**= 선택자의 프로퍼티를 값으로 변경하겠음!**

# CSS 많이 쓰이는 속성

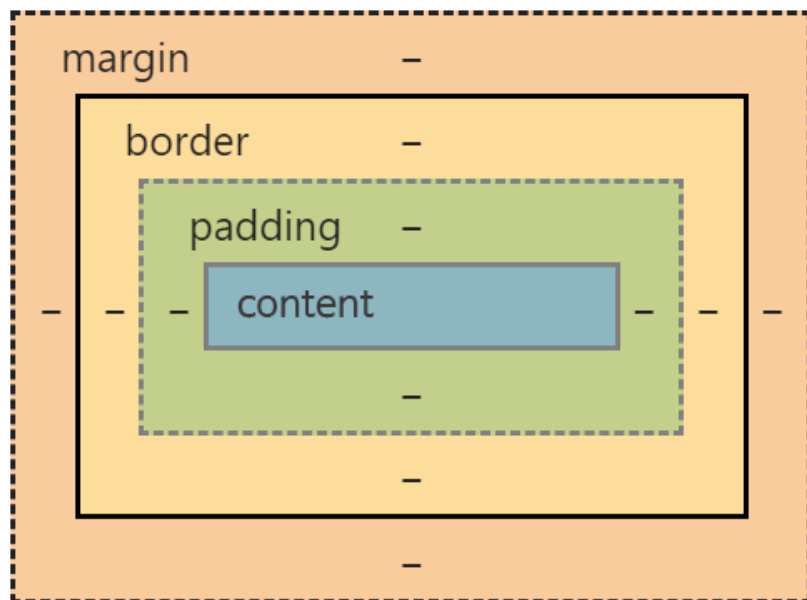
# CSS 많이 쓰이는 속성

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



# CSS 박스 모델

- **box-sizing** 속성



## 1. **content-box**

: 콘텐츠 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 많이 쓰이는 속성

- margin, padding

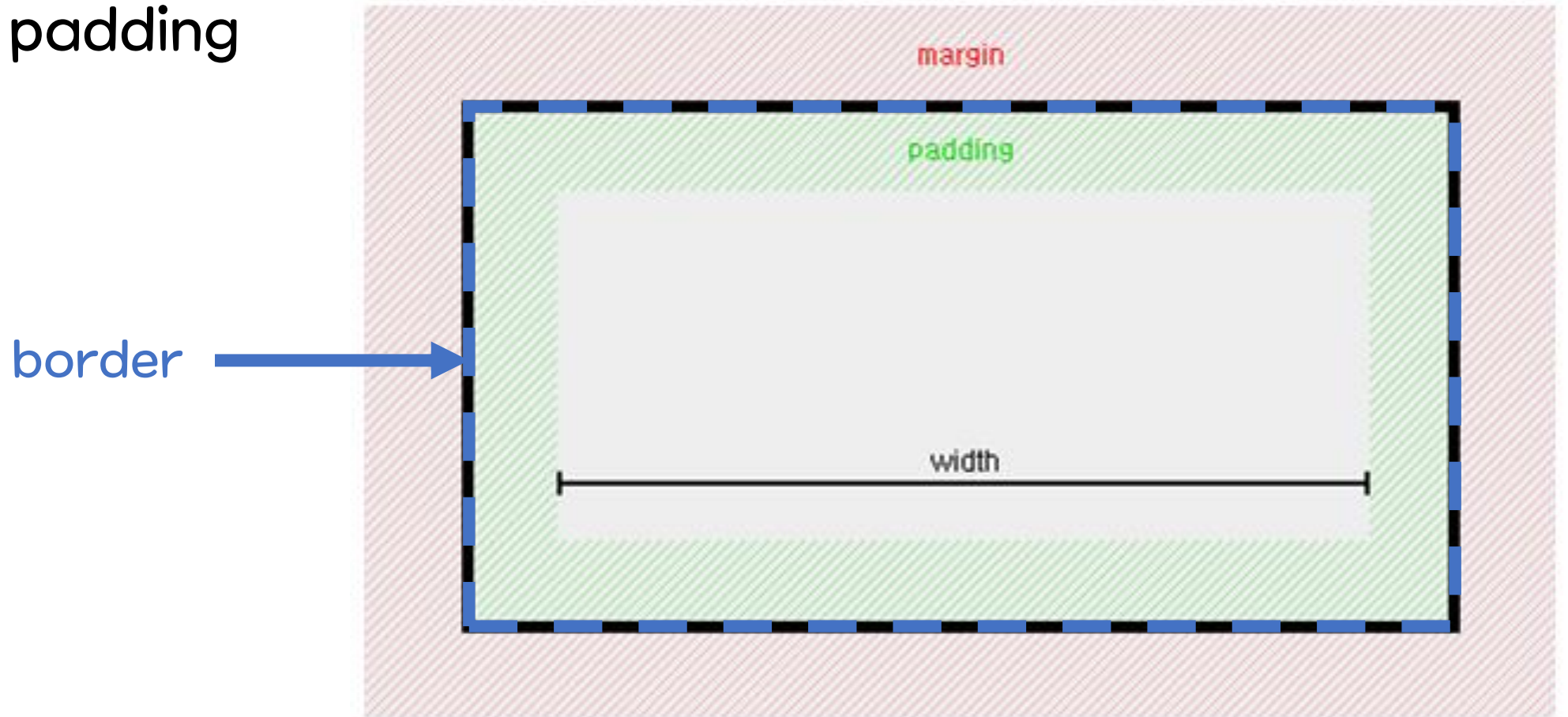


사진 출처: <https://ofcourse.kr/css-course/margin-padding-%EC%86%8D%EC%84%B1>

# CSS Selector

# CSS 선택자 – 단일 선택자

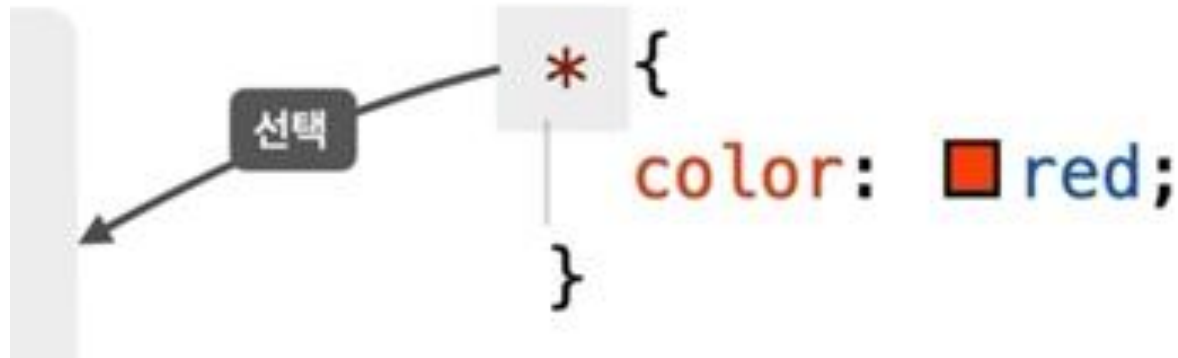
- 전체 선택자
- 태그 선택자
- Class 선택자
- ID 선택자

\*

기본

전체 선택자 (Universal Selector)

모든 요소를 선택.



# CSS 선택자 – 단일 선택자

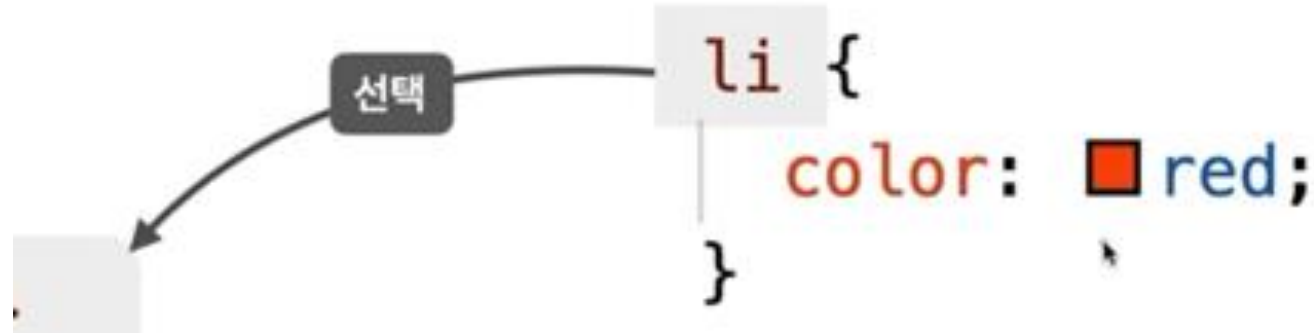
- 전체 선택자
- 태그 선택자
- Class 선택자
- ID 선택자

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>
```

선택

```
.orange {
  color: ■ red;
}
```

여러 요소에 같은 class값을 지정하여  
동일한 스타일을 적용

# 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>
```

선택

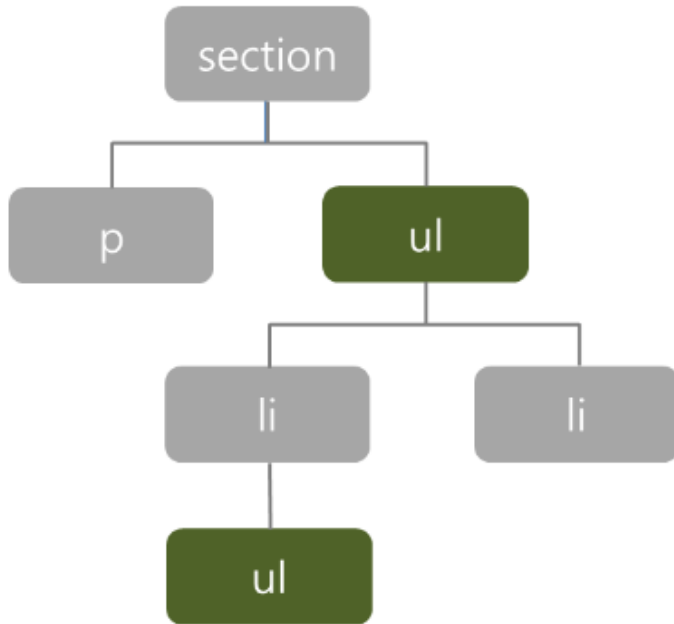
```
#orange {
  color: ■ red;
}
```

페이지당 하나만 갖는 단일 값



# 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>
  
```

선택

```

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



# CSS 선택자 – 복합 선택자

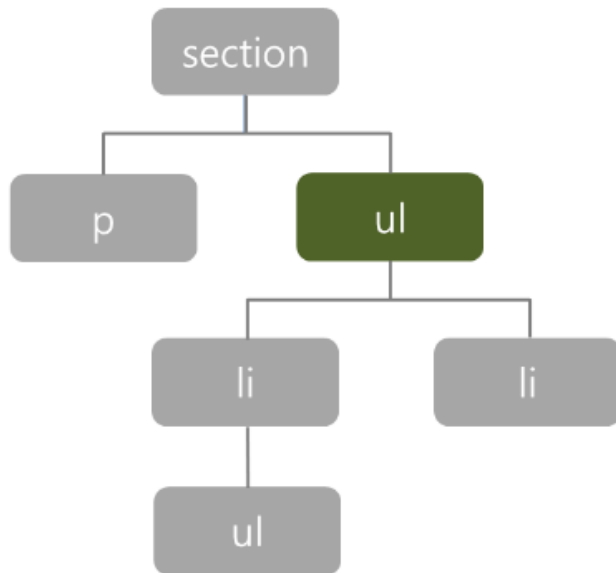
## • 자식 선택자

### ABC > XYZ

복합

자식 선택자 (Child Combinator)

선택자 ABC의 자식 요소 XYZ 선택.



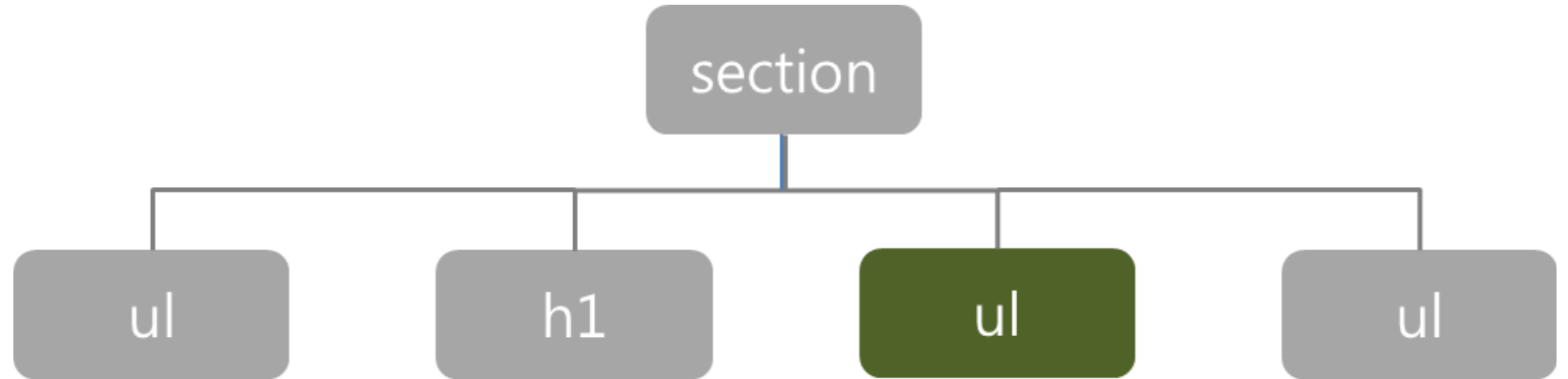
```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
```

선택

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

# CSS 선택자 – 복합 선택자

## • 인접 형제 선택자

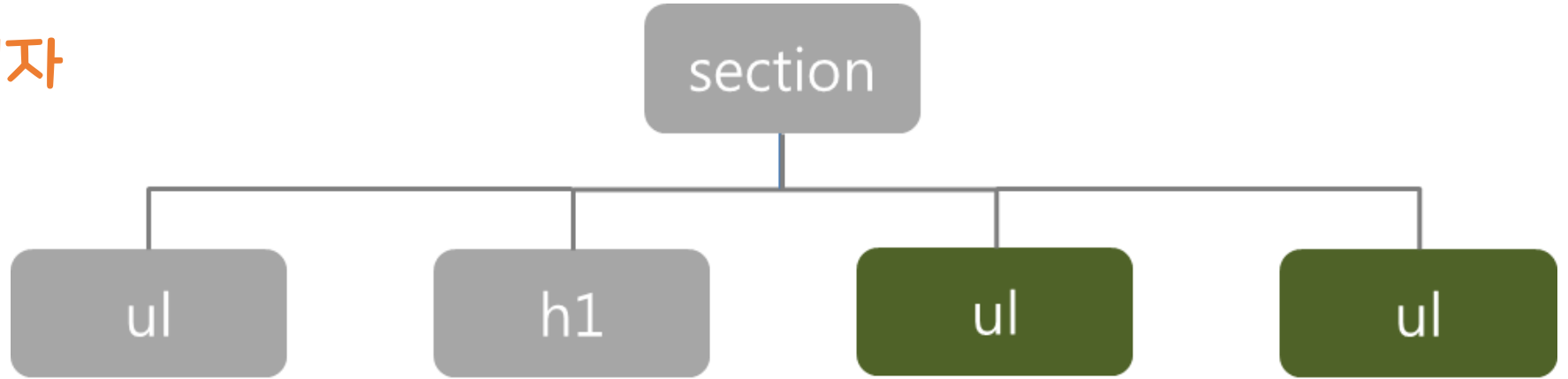


# h1의 형제 중 h1 바로 뒤에 있는 ul

`h1 + ul { }`

# CSS 선택자 – 복합 선택자

## • 일반 형제 선택자



# h1 다음 형제 중 h1 뒤에 위치한 ul 모두 선택

`h1 ~ ul { }`

# CSS 선택자 – 복합 선택자

- 가상 선택자      E:active { }

E:hover { }

E:focus { }

E:visited { }

# CSS 선택자 – 복합 선택자

## • 가상 선택자

E:first-child { }

## ABC:first-child

가상 클래스 선택자 (Pseudo-Classes)

FIRST CHILD

선택자 ABC가 형제 요소 중 첫째라면 선택.

```
<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 – 복합 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 – 복합 선택자

## • 가상 선택자

E:nth-child(n) { }

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(2) {
  color: ■ red;
}
```

# CSS 선택자 – 복합 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>
```

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

선택



# 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

# CSS 복잡한 속성

# CSS 복잡한 속성 - position

- **static** : 정적 위치 지정 방식 (default)
- **relative** : 상대 위치 지정 방식
- **absolute** : 절대 위치 지정 방식
- **fixed** : 고정 위치 지정 방식

# CSS 복잡한 속성 - position

## position 속성

Aa 속성	설명	+
<u>position</u>	HTML 요소의 위치를 결정하는 방식을 설정함.	
<u>top</u>	위치가 설정된 조상 요소의 위로부터의 여백을 설정함.	
<u>right</u>	위치가 설정된 조상 요소의 오른쪽으로부터의 여백을 설정함.	
<u>bottom</u>	위치가 설정된 조상 요소의 아래로부터의 여백을 설정함.	
<u>left</u>	위치가 설정된 조상 요소의 왼쪽으로부터의 여백을 설정함.	
<u>z-index</u>	겹쳐지는 요소들이 쌓이는 스택(stack)의 순서를 설정함.	

# CSS 복잡한 속성 - transform

요소에 이동(translate), 회전(rotate), 확대축소(scale), 비틀기(skew) 효과를 부여하기 위한 함수를 제공

- **skew** : 기울기
- **scale** : 확대
- **rotate** : 회전
- **translate** : 이동

# CSS 복잡한 속성 - transform

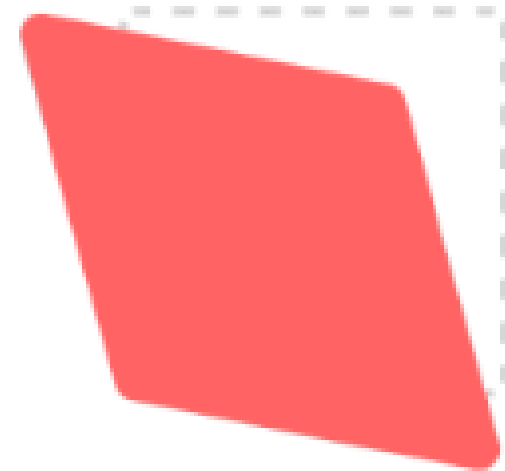
- skew : 기울기



`transform: skewX(10deg);`



`transform: skewY(10deg);`

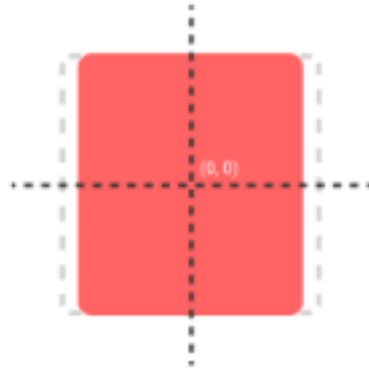


`transform: skew(10deg, 10deg);`

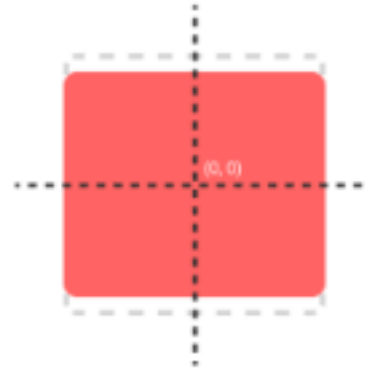
사진 출처: <https://dev.to/kunaal438/css-transform-complete-guide-on-css-transform-everything-you-need-for-good-developer-841>

# CSS 복잡한 속성 - transform

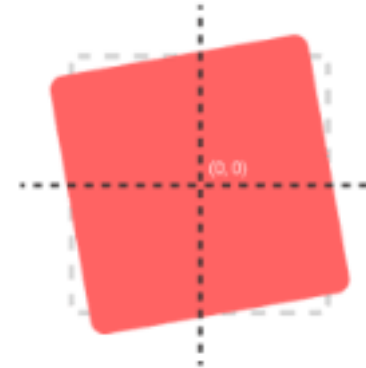
- rotate : 회전



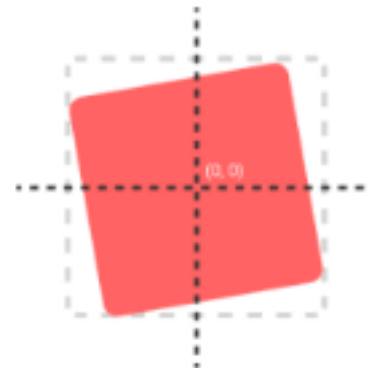
`transform: rotateX(10deg);`



`transform: rotateY(10deg);`



`transform: rotate(10deg);`

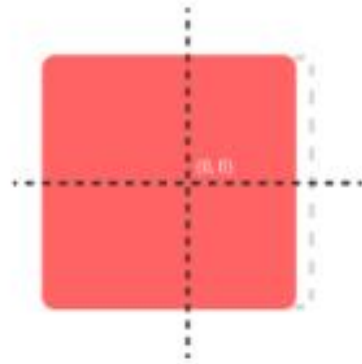


`transform: rotate3d(10deg, 10deg, 10deg);`

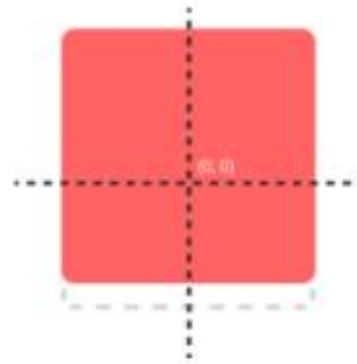
사진 출처: <https://dev.to/kunaal438/css-transform-complete-guide-on-css-transform-everything-you-need-for-good-developer-841>

# CSS 복잡한 속성 - transform

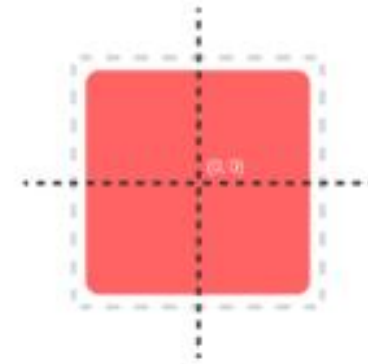
- translate : 이동



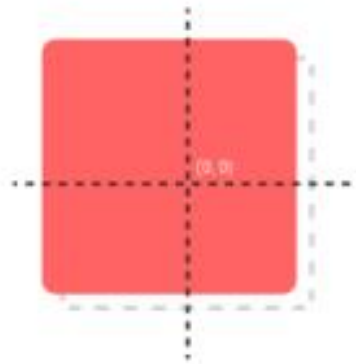
`transform: translateX(-20px);`



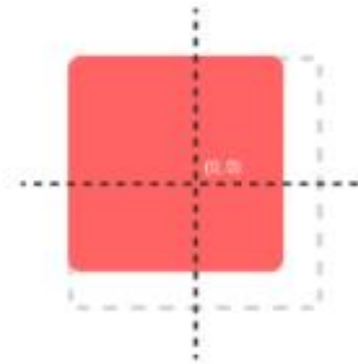
`transform: translateY(-20px);`



`transform: translateZ(-20px);`



`transform: translate(-20px, -20px);`

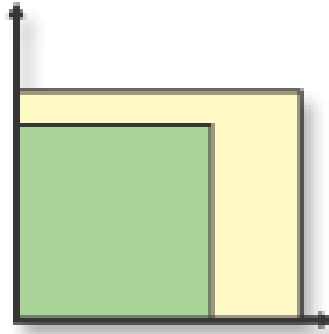


`transform: translate3d(-20px, -20px, -20px);`

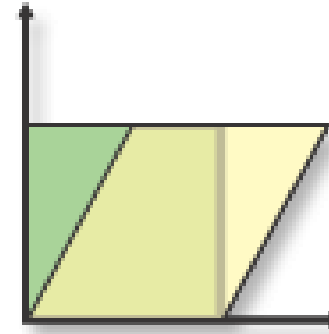
사진 출처: <https://dev.to/kunaal438/css-transform-complete-guide-on-css-transform-everything-you-need-for-good-developer-841>



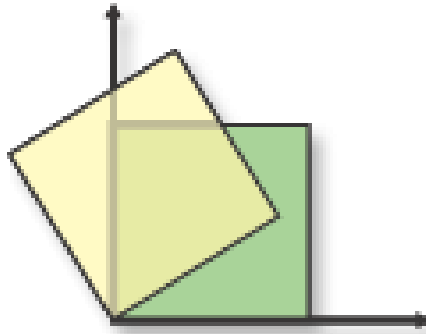
# CSS 복잡한 속성 - transform



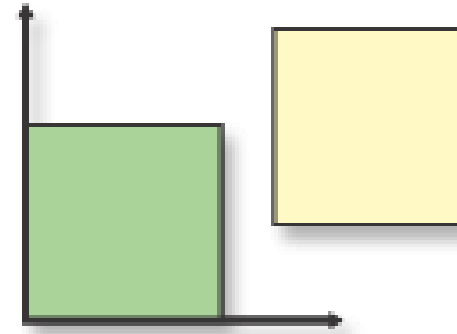
DIFFERENTIAL SCALING



SKEW



ROTATION



TRANSLATION

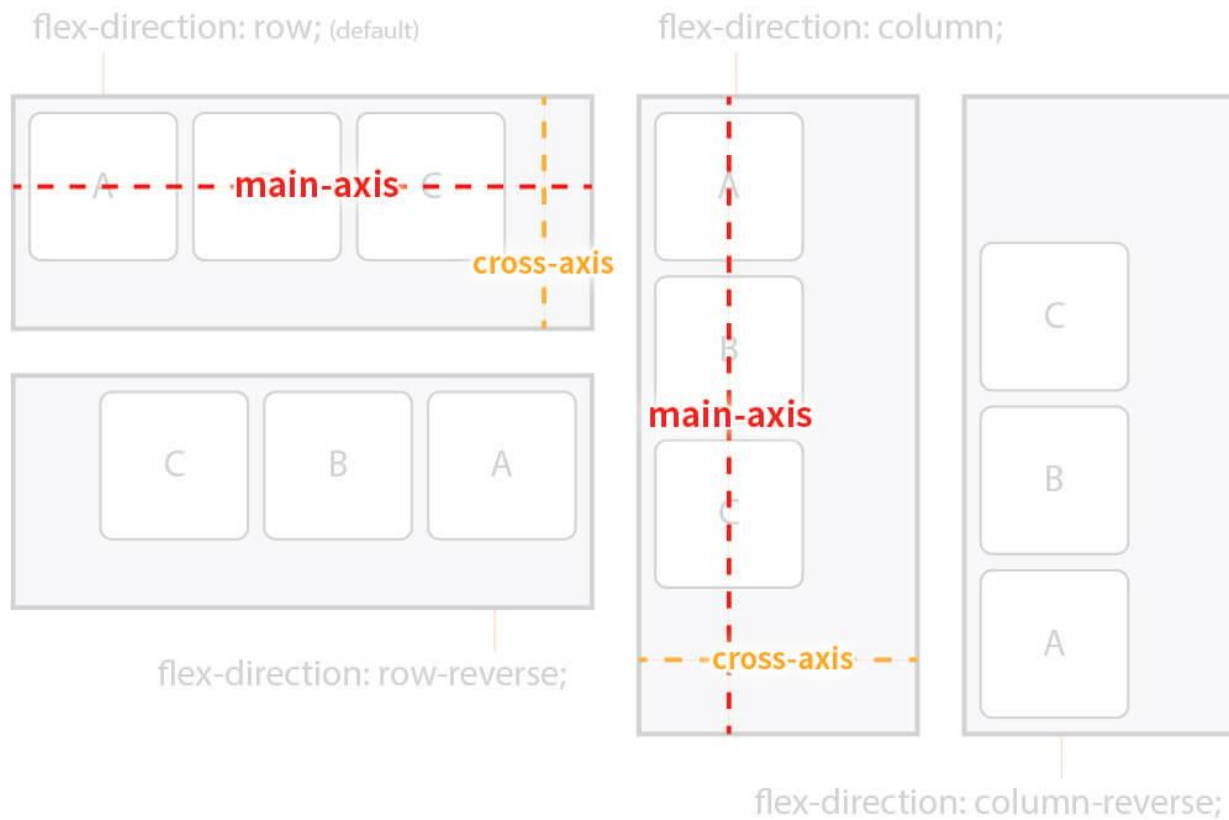
사진 출처: <https://desktop.arcgis.com/en/arcmap/10.6/tools/coverage-toolbox/how-transform-works.htm>

# CSS 복잡한 속성 - display

- inline
- block
- inline-block
- flex
- none

# CSS 복잡한 속성 - display

- flex 의 주축(main axis)과 교차축 (cross axis)



# CSS 복잡한 속성 - animation

애니메이션을 나타내는 css 스타일

+

애니메이션의 중간 상태를 나타내는 키 프레임

@keyframes 로 설정되는 값 ( to ~ from, 0% ~ 100% )

# CSS 복잡한 속성 - animation

- Animation-name
- Animation-duration
- Animation-delay
- Animation-iteration-count
- Animation-timing-function

# CSS 복잡한 속성 - animation

- Animation-timing-function

linear	처음 속도와 마지막 속도가 일정합니다.
ease	처음엔 천천히 시작하여 빨라지고 마지막에 다시 느려집니다.
ease-in	천천히 시작되어 정상 속도가 됩니다.
ease-out	마지막에 천천히 속도를 줄여 끝납니다.
ease-in-out	천천히 시작되어 정상 속도가 되었다가 마지막에 느려집니다.