

# 13. CSS 고급 선택자



## 13-1 연결 선택자

## 13-2 속성 선택자

## 13-3 가상 클래스와 가상 요소

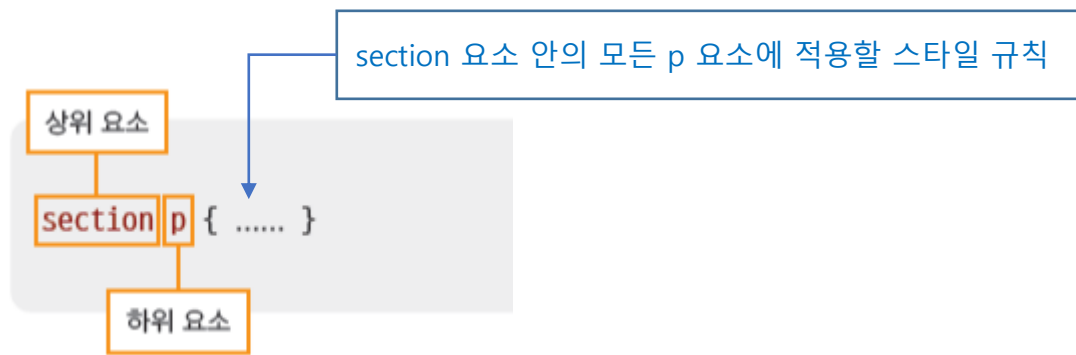
# 연결 선택자 – 하위 선택자

- 연결 선택자 : 선택자와 선택자를 연결해 적용 대상을 제한하는 선택자.
- **컴비네이션 선택자**(combination selector) 또는 '**조합 선택자**'라고도 함

## 하위 선택자(descendant selector)

- 부모 요소에 포함된 모든 하위 요소에 스타일이 적용된다
- 자식 요소뿐만 아니라 손자 요소, 손자의 손자 요소 등 모든 하위 요소까지 적용
- 하위 선택자를 정의할 때는 상위 요소와 하위 요소를 나란히 쓴다.

기본형   상위요소   하위요소



```
<style>
  section p { color: blue; }
</style>

.....

<section>
  <h1>예약 방법 & 사용 요금</h1>
  <p>아직 온라인 예약 신청이.....</p>
  <div>
    <p>가족실(2~4인) : 60,000원/일</p>
    <p>도미토리(4인 공용) : 25,000원/일</p>
  </div>
</section>
```

### 예약 방법 & 사용 요금

아직 온라인 예약 신청이 준비되지 않았습니다.  
전화(0xx-xxxx-xxxx)로 문의하시기 바랍니다.

가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일

# 연결 선택자 – 자식 선택자 (child selector)

- 자식 요소에 스타일을 적용하는 선택자
- 두 요소 사이에 '>(부등호)'를 표시해 부모 요소와 자식 요소를 구분

기본형 부모요소 > 자식요소



section 요소 안에 포함된 p 요소 중  
자식 p 요소에만 적용할 스타일 규칙

```
<style>
  section > p { color: blue; }
</style>

.....

<section>
  <h1>예약 방법 & 사용 요금</h1>
  <p>아직 온라인 예약 신청이.....</p>
  <div>
    <p>가족실(2~4인): 60,000원/일</p>
    <p>도미토리(4인 공용): 25,000원/일</p>
  </div>
</section>
```

## 예약 방법 & 사용 요금

아직 온라인 예약 신청이 준비되지 않았습니다.  
전화(0xx-xxxx-xxxx)로 문의하시기 바랍니다.

가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일

# 연결 선택자 – 인접 형제 선택자(adjacent selector)

- 같은 부모를 가진 형제 요소 중 첫 번째 동생 요소에만 스타일 적용
- 요소1과 요소2 사이에 '+' 기호 사용
- 요소1과 요소2는 같은 레벨이면서 요소1 이후 첫 번째 요소2에 적용

기본형 요소1 + 요소2

h1의 형제인 p 요소 중 첫 번째 p

h1 + p { color: blue; }

h1 요소 다음에 오는 p 요소들 중  
첫 번째 p 요소에만 파란색 글씨 적용

```
<style>
  h1 + p {
    background-color: #222; /* 배경색은 검은색으로 */
    color: #fff; /* 글자색은 흰색으로 */
  }
</style>
```

.....

<h1>예약 방법 & 이용 요금</h1>

<p>아직 온라인 예약.....</p>

<p>가족실(2~4인) : 60,000원/일</p>

<p>도미토리(4인 공용) : 25,000원/일</p>

## 예약 방법 & 이용 요금

아직 온라인 예약 신청이 준비되지 않았습니다.

전화(000-xxxx-xxxx)로 문의하시기 바랍니다.

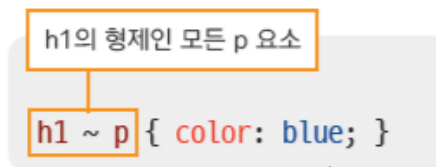
가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일

# 연결 선택자 - 형제 선택자(sibling selector)

- 형제 요소들에 스타일 적용
- 인접 형제 선택자와 다른 점은 모든 형제 요소에 다 적용된다는 것
- 요소1과 요소2 사이에 기호 '~' 사용

기본형 요소1 ~ 요소2



h1 요소 다음에 오는  
모든 형제 p 요소에 밑줄 적용

```
<style>
  h1 ~ p {
    background-color: #222; /* 배경색은 검은색으로 */
    color: #fff; /* 글자색은 흰색으로 */
  }
</style>
```

```
.....
<h1>예약 방법 & 이용 요금</h1>
<p>아직 온라인 예약.....</p>
<p>가족실(2~4인) : 60,000원/일</p>
<p>도미토리(4인 공용) : 25,000원/일</p>
```

## 예약 방법 & 이용 요금

아직 온라인 예약 신청이 준비되지 않았습니다.  
전화(0000-0000-0000)로 문의하시기 바랍니다.


가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일

# 속성 선택자

## [속성] 선택자

지정한 속성을 가진 요소를 찾아 스타일 적용

 **Do it!** 요소 중에서 링크가 있는 요소만 스타일 적용하기 예제 파일 10\attr-1.html

```
(... 생략 ...)  
a[ href ] { /* href 속성이 있는 a 요소를 찾는 선택자 */  
  background: yellow; /* 배경색을 노란색으로 */  
  border: 1px solid #ccc; /* 테두리 지정 */  
  font-weight: normal; /* 글꼴 굵기 지정 */  
}  
</style>  
.....  
<ul>  
  <li><a>메인 메뉴 : </a></li>  
  <li><a href="#">메뉴 1</a></li>  
  <li><a href="#">메뉴 2</a></li>  
  <li><a href="#">메뉴 3</a></li>  
  <li><a href="#">메뉴 4</a></li>  
</ul>
```

a 요소 중에서 href 속성값이 있는 요소만 적용

메인 메뉴 :

메뉴 1


메뉴 2

메뉴 3

메뉴 4


## [속성 = 값] 선택자

주어진 속성과 속성 값이 일치하는 요소를 찾아 스타일 적용

 **Do it!** 새 탭으로 열리는 링크에만 아이콘 추가하기 예제 파일 10\attr-2.html

```
(... 생략 ...)  
a[ target="_blank" ] { /* target 속성값이 _blank인 a 요소를 선택하는 선택자 */  
  padding-right: 30px; /* 오른쪽에 패딩 지정 */  
  background: url(images/newwindow.png) no-repeat center right; /* 배경 이미지 지정 */  
}  
</style>  
.....  
<ul>  
  <li><a href="https://html.spec.whatwg.org" target="_blank">HTML</a></li>  
  <li><a href="https://www.w3.org/TR/selectors">CSS Selector Level 3</a></li>  
  <li><a href="https://www.w3.org/TR/css3-mediaqueries">미디어쿼리</a></li>  
</ul>  
(... 생략 ...)
```

a 요소 중에서 href 속성값이 \_blank인 요소에만 적용

HTML 


CSS Selector Level 3

미디어쿼리

# 속성 선택자

## [속성 ~= 값] 선택자

여러 속성 값 중에 해당 값이 포함되어 있는 요소를 찾아 스타일 적용

 **Do it!** 특정 속성값이 포함된 요소에 스타일 적용하기 예제 파일 10\attr-3.html

(... 생략 ...)

```
[class ~= "button"] {  
    box-shadow: rgba(0, 0, 0, 0.4) 4px 4px; /* 그림자 지정 */  
    border-radius: 5px; /* 모서리 둥글게 지정 */  
}
```

class 속성에 button 값이 있는 요소를 찾는 선택자

```
</style>  
  
.....  
<ul>  
    <li><a href="#" class="flat">메뉴 1</a></li>  
    <li><a href="#" class="flat">메뉴 2</a></li>  
    <li><a href="#" class="button">메뉴 3</a></li> 속성값이 button인 요소에 적용  
    <li><a href="#" class="flat button">메뉴 4</a></li> 속성값으로 button을 포함하는 요소에 적용  
</ul>
```

(... 생략 ...)



외국어 서비스 :  영어  일본어  중국어

## [속성 |= 값] 선택자

- 특정 값이 포함된 속성을 가진 요소를 찾아 스타일 적용
- 하이픈으로 연결해 한 단어 값을 이루는 요소에도 적용

 **Do it!** title 속성값에 따라 아이콘 다르게 표시하기 예제 파일 10\attr-4.html

(... 생략 ...)

```
1 a[title |= us] { /* 속성값이 "us"이거나 "us-"로 시작하는 a 요소를 찾는 선택자 */  
    background: url(images/us.png) no-repeat left center; /* 아이콘 적용 */  
}  
  
2 a[title |= jap] { /* 속성값이 "jap"이거나 "jap-"로 시작하는 a 요소를 찾는 선택자 */  
    background: url(images/jp.png) no-repeat left center; /* 아이콘 적용 */  
}  
  
3 a[title |= chn] { /* 속성값이 "chn"이거나 "chn-"으로 시작하는 a 요소를 찾는 선택자 */  
    background: url(images/ch.png) no-repeat left center; /* 아이콘 적용 */  
}
```

```
</style>  
  
.....  
<ul>  
    <li>외국어 서비스 : </li>  
    <li><a href="#" title="us-english">영어</a></li> 속성값이 "us-"로 시작하는 한 단어이므로 1 스타일이 적용됨  
    <li><a href="#" title="ja">일본어</a></li> 속성값이 "jap"도, "jap-"도 아니므로 2 스타일이 적용되지 않음  
    <li><a href="#" title="chn">중국어</a></li> 속성값이 "chn"과 일치하므로 3 스타일이 적용됨  
</ul>
```

(... 생략 ...)

# 속성 선택자

## [속성 ^= 값] 선택자

특정 값으로 시작하는 속성을 가진 요소를 찾아 스타일 적용

 **Do it!** 속성값의 시작 부분이 일치하는 요소에 스타일 적용하기

예제 파일 10\attr-5.html



(... 생략 ...)

```
1 a[title ^= "eng"] { /* 속성값이 "eng"로 시작하는 a 요소를 찾는 선택자 */
  background: url(images/us.png) no-repeat left center; /* 아이콘 적용 */
  padding: 5px 25px; /* 패딩 지정 */
}
2 a[title ^= "jap"] { /* 속성값이 "jap"로 시작하는 요소를 찾는 선택자 */
  background: url(images/jp.png) no-repeat left center; /* 아이콘 적용 */
  padding: 5px 25px; /* 패딩 지정 */
}
3 a[title ^= "chin"] { /* 속성값이 "chin"으로 시작하는 요소를 찾는 선택자 */
  background: url(images/ch.png) no-repeat left center; /* 아이콘 적용 */
  padding: 5px 25px; /* 패딩 지정 */
}
</style>
.....
<ul>
  <li>외국어 서비스 : </li>
  <li><a href="#" title="english">영어</a></li>
  <li><a href="#" title="japanese">일본어</a></li>
  <li><a href="#" title="chinese">중국어</a></li>
</ul>
(... 생략 ...)
```

속성값이 "eng"로 시작하므로 ❶ 스타일 적용


속성값이 "jap"로 시작하므로 ❷ 스타일 적용

속성값이 "chin"으로 시작하므로 ❸ 스타일 적용

외국어 서비스 :  영어  일본어  중국어

## [속성 \$= 값] 선택자

특정 값으로 끝나는 속성을 가진 요소를 찾아 스타일 적용

 **Do it!** 파일 확장자에 따라 아이콘 다르게 표시하기

예제 파일 10\attr-6.html

(... 생략 ...)

```
1 a[href $= hwp] { /* 링크 속성값이 hwp로 끝나는 a 요소를 찾는 선택자 */
  background: url(images/hwp_icon.gif) center right no-repeat;
  padding-right: 25px; /* 오른쪽 패딩 지정 */
}
2 a[href $= xls] { /* 링크 속성값이 xls로 끝나는 a 요소를 찾는 선택자 */
  background: url(images/excel_icon.gif) center right no-repeat;
  padding-right: 25px; /* 오른쪽 패딩 지정 */
}
</style>
.....
<h3>회사 소개 파일 내려받기</h3>
<ul>
  <li><a href="intro.hwp">hwp 파일</a></li>
  <li><a href="intro.xls">엑셀 파일</a></li>
</ul>
(... 생략 ...)
```

속성값이 "hwp"로 끝나므로 ❶ 스타일이 적용됨

속성값이 "xls"로 끝나므로 ❷ 스타일이 적용됨

회사 소개 파일 내려받기

- hwp 파일 
- 엑셀 파일 

jeju:김윤



# 속성 선택자

## [속성\*=값] 선택자

값의 일부가 일치하는 속성을 가진 요소를 찾아 스타일 적용



Do it! 속성값의 일부가 일치하는 요소에 스타일 적용하기

예제 파일 10\attr-7.html

(... 생략 ...)

```
a[href *= w3] { /* 속성값 중 일부에 w3가 있는 a 요소를 찾는 선택자 */
  background: blue;
  color: white;
}
</style>

.....
<p>(아래 링크 중에서 배경색이 파란색인 링크는 W3C 사이트로 연결됩니다.)</p>
<ul>
  <li><a href="https://html.spec.whatwg.org/">HTML 표준안 사이트</a></li>
  <li><a href="https://caniuse.com/">HTML 지원 여부 체크</a></li>
  <li><a href="https://www.w3.org/TR/css3-mediaqueries">미디어쿼리</a></li>
</ul>
```

(... 생략 ...)

href의 속성값 중에서 "w3"가 일치하므로 스타일을 적용

## HTML5 참고 사이트

(아래 링크 중에서 배경색이 파란색인 링크는 W3C 사이트로 연결됩니다.)

- HTML 표준안 사이트
- HTML 지원 여부 체크
- 미디어쿼리

## 속성 선택자 정리

종류	선택 요소	예시
[속성]	해당 속성이 있는 요소	[required]
[속성 = 값]	지정한 속성값이 있는 요소	[target = _blank]
[속성 ~= 값]	지정한 속성값이 포함된 요소(단어별)	[class ~= button]
[속성  = 값]	지정한 속성값이 포함된 요소(하이픈 포함, 단어별)	[title  = us]
[속성 ^= 값]	지정한 속성값으로 시작하는 요소	[title ^= eng]
[속성 \$= 값]	지정한 속성값으로 끝나는 요소	[href \$= xls]
[속성 *= 값]	지정한 속성값의 일부가 일치하는 요소	[href *= w3]

# 가상 클래스

## 사용자 동작에 반응하는 가상 클래스 선택자

**:link** - 방문하지 않은 링크에 스타일 적용

**:visited** - 방문한 링크에 스타일 적용

**:active** - 웹 요소를 활성화했을 때의 스타일 적용

**:hover** - 웹 요소에 마우스 커서를 올려놓을 때의 스타일 적용

**:focus** - 웹 요소에 초점이 맞추어졌을 때의 스타일 적용

순서 중요 :link → :visited → :hover → :active



```
<style>
.navi ul li {
  float: left;
  width: 150px;
  padding: 10px;
}
.navi a:link, .navi a:visited {
  display: block;
  font-size: 14px;
  color: #000;
  padding: 10px;
  text-decoration: none; /* 밑줄을 없앴 */
  text-align: center;
}
.navi a:hover, .navi a:focus {
  background-color: #222; /* 배경색을 짙은 회색으로 변경 */
  color: #fff;
}
.navi a:active {
  background-color: #f00; /* 배경색을 빨간색으로 변경 */
}
</style>

.....
<nav class="navi">
  <ul>
    <li><a href="#">이용 안내</a></li>
    <li><a href="#">객실 소개</a></li>
    <li><a href="#">예약 방법 및 요금</a></li>
    <li><a href="#">예약하기</a></li>
  </ul>
</nav>
```

# 가상 클래스와 가상 요소

## 요소 상태에 따른 가상 클래스 선택자

**:target** – 앵커로 연결된 부분에 스타일 적용

**:enabled**, **:disabled** – 요소의 사용 여부에 따라 스타일 적용

**:checked** – 라디오 버튼이나 체크 박스에 체크했을 때 스타일 적용

객실 형태

☐ 기본형(최대 2인)

☒ 가족형(최대 8인)

```
<style>
  #signup input:checked + label {
    color: red; /* 글자색을 빨간색으로 지정 */
    font-weight: bold; /* 글꼴을 굵게 지정 */
  }
</style>

.....

<ul>
  <li>
    <input type="radio" name="room" id="basic">
    <label for="basic">기본형(최대 2인)</label>
  </li>
  <li>
    <input type="radio" name="room" id="family">
    <label for="family">가족형(최대 8인)</label>
  </li>
</ul>
```

# 가상 클래스

## 문서 구조에 따른 가상 클래스 선택자

- 웹 문서의 구조를 기준으로 특정 위치에 있는 요소를 찾아 스타일 적용
- 위치가 계속 바뀐다면  $an+b$  처럼 수식을 사용할 수도 있음.  
이 때  $n$  값은 0부터.

선택자	설명
:first-child	여러 요소 중에서 첫 번째 요소를 선택합니다.
:last-child	여러 요소 중에서 마지막 요소를 선택합니다.
:only-child	여러 요소 중에서 자식 요소가 하나일 때 선택합니다.
:nth-child(숫자 또는 수식)	지정한 위치의 자식 요소를 선택합니다.
A:nth-of-type(숫자 또는 수식)	A 요소를 기준으로 지정한 위치의 자식 요소를 선택합니다.

# 가상 클래스

## :first-child, :last-child, :only-child

```
<div class="container">
  <p>첫 번째</p>
  <div>두 번째</div>
  <p>세 번째</p>
  <div>네 번째</div>
</div>
<div class="container">
  <div>혼자~</div>
</div>
```

첫 번째  
두 번째  
세 번째  
네 번째

혼자~

### 첫 번째 자식 요소 선택

```
<style>
.....
.container :first-child {
  background-color: #fc2;
}
</style>
```

첫 번째

두 번째

세 번째

네 번째

혼자~

### 마지막 자식와 유일 요소 선택

```
<style>
.....
.container :first-child {
  background-color: #fc2;
}
.container :last-child {
  background-color: #2cf;
}
.container div:only-child {
  background-color: #f0c8fc;
}
</style>
```

첫 번째

두 번째

세 번째

네 번째

혼자~

# 가상 클래스

## :nth-child()

- 괄호 안에 위치값을 숫자로 지정  
(예) :nth-child(5)
- odd, even을 사용할 수도 있음  
(예) :nth-child(even)
- 패턴이 있을 경우 처럼 식을 사용할 수 있음  
(예) :nth-child(2n), :nth-child(2n+1)
- 특정 위치부터 끝까지, 혹은 처음부터 특정 위치까지 선택할 수 있음  
(예) :nth-child(n+4) /\* 네번째부터 끝까지 선택 \*/  
:nth-child(-n+4) /\* 처음부터 네번째까지 선택 \*/

```
<style>
.....
.container :nth-child(3) { /*세 번째 자식 요소 선택 */
    background-color: #fc2;
}
</style>
```



클래스	결과 화면							
:nth-child(2n) 또는 :nth-child(even)	1	2	3	4	5	6	7	8
:nth-child(2n+1) 또는 :nth-child(odd)	1	2	3	4	5	6	7	8
:nth-child(3n)	1	2	3	4	5	6	7	8

# 가상 클래스

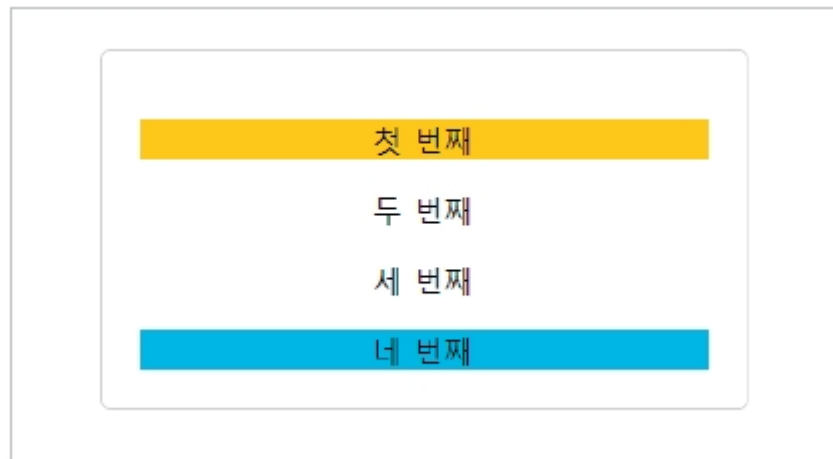
## :nth-of-type()

- 같은 타입의 요소들 사이에서 순서를 지정하는 클래스

(예) .container 하위 요소 중 p 태그 요소 중에서 첫번째를 선택하려면

.container p:nth-of-type(1)

```
<style>
.....
.container p:nth-of-type(1) { /* p 요소 중 첫 번째 요소 선택 */
  background-color: #fc2;
}
.container div:nth-of-type(2) { /* div 요소 중 두 번째 요소 선택 */
  background-color: #2cf;
}
</style>
```

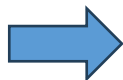


# 가상 클래스

## :not()

- 괄호 안에 있는 요소를 제외하고 선택
- CSS 선택자 부분을 간결하게 표현할 수 있다

```
<style>
.....
#signup input[type=text], input[type=tel] {
  border: 1px solid #ccc;
  border-radius: 3px;
  padding: 5px;
  width: 200px;
}
</style>
```



```
<style>
.....
#signup input:not[type=radio] {
  border: 1px solid #ccc;
  border-radius: 3px;
  padding: 5px;
  width: 200px;
}
</style>
```



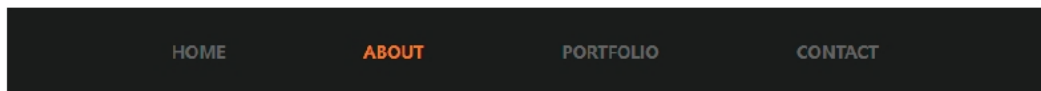
# 가상 클래스

## :has()

A:has(B)

→ A 요소가 괄호 안의 B 요소를 가지고 있다면 선택

(예) 메뉴 항목 위로 마우스 커서를 올렸을 때 해당 메뉴 외의 다른 메뉴 항목의 글자색이 바뀌고 희미해지게 바꾸려면



① nav 요소에 a:hover 스타일이 있다면

② :hover 선택자가 적용되지 않는 요소를 선택합니다.

2) :hover 선택자가 적용되지 않는 링크에 스타일 적용

nav:has(a:hover) a:not(:hover)

1) nav 요소에 a:hover 선택자가 있다면

```
<style>
```

```
.....
```

```
nav a:hover {  
  color: #ff7a18;  
}
```

```
nav:has(a:hover) a:not(:hover) {  
  opacity: 0.3;  
}
```

```
</style>
```

# 가상 요소

화면 꾸미기용 요소를 웹 문서에 포함시키지 않기 위해 가상 요소 사용

## ::first-line, ::first-letter 요소

- ::first-line : 특정 요소의 첫번째 줄에 스타일 적용
- ::first-letter : 특정 요소의 첫번째 글자에 스타일 적용

```
<style>
.....
p::first-letter {
  font-size: 2em;
  font-weight: bold;
}
</style>

<div class="container">
  <h1>CSS 가상 요소</h1>
  <p>가상 요소를 사용하면 ..... </p>
</div>
```

### CSS 가상 요소

**가**상 요소를 사용하면 CSS로 시각적 변화를 줄 수 있습니다. 이는 성능을 저하시키지 않으면서 웹 페이지를 꾸밀 수 있는 효율적인 방법입니다.

## 가상 요소

화면 꾸미기용 요소를 웹 문서에 포함시키지 않기 위해 가상 요소 사용

## ::before, ::after 요소

- 웹 요소의 앞이나 뒤에 내용이나 스타일을 넣을 수 있다.
- 항상 content 속성을 사용해야 한다. 내용없이 스타일만 지정할 경우에는 content: " "

```
<style>

.....

h1::before {
  content: "*";
  color: red;
  margin-right: 5px;
}

h1::after {
  content: " ";
  margin-left: 10px;
}

</style>

<div class="container">

  <h1>CSS 가상 요소</h1>

  <p>가상 요소를 사용하면 ..... </p>

</div>
```



# CSS 함수 살펴보기

## :root 가상 클래스

- 문서의 최상위 요소 선택. <html> 태그에 해당하는 부분
- CSS 변수를 사용할 때

## CSS 변수

- 변수 이름 앞에 하이픈(-- ) 2개 붙임
- 변수의 용도를 알 수 있도록 이름 붙이기

## var() 함수

- 사용자가 미리 만든 CSS 변수를 사용할 때
- 괄호 안에 변수 이름 넣기

```
:root {  
  --main-bg-color: #222;  
  --main-text-color: #fff;  
}  
body {  
  background-color: var(--main-bg-color);  
  color: var(--main-text-color);  
}
```

나중에 배경색과 글자색을 바꾸고 싶을 때 :root 클래스의 변수 값만 수정하고 코드의 다른 부분은 수정하지 않아도 된다.

# CSS 함수 살펴보기

## calc() 함수

- 괄호 안에는 크기 값이나 각도, 시간, 백분율, 숫자 등 다양한 단위 값을 사용한 계산식이 들어감

(예) nav 요소의 너비를 전체 너비에서 80px만큼 뺀 후 계산

```
nav {  
  width: calc(100% - 80px);  
}
```

- 더하거나 빼기 연산자를 사용할 때는 반드시 연산자 앞뒤에 공백

```
nav {  
  width: calc(100% - -80px);  
}
```

```
<style>  
  
.....  
.heading {  
  width: calc(100% - 80px);  
  margin: auto;  
  background-color: #222;  
  
.....  
}  
</style>
```

CSS 함수

CSS 함수

# CSS 함수 살펴보기

## min(), max() 함수

- min() 함수 : 괄호 안에 나열된 값 중 가장 작은 값
- max() 함수 : 괄호 안에 나열된 값 중 가장 큰 값

```
<style>
.....
.heading {
  width: min(100%, 700px);
  margin: auto;
.....
}

.heading h1 {
  font-size: max(3vw, 2em); /* 3vw, 2rem 중 큰 값 */
}
</style>
```

CSS 함수 - min()

## clamp() 함수

- 값의 범위 제한
- 최솟값과 최적값, 최댓값 3가지 매개변수 사용

```
<style>
.....
.heading h1 {
  font-size: clamp(1rem, 2vw, 2rem);
}
</style>
```



# filter 속성과 함수들

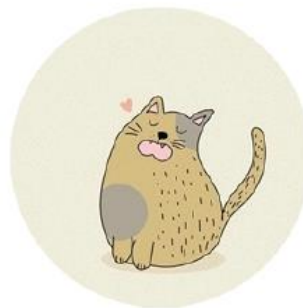
## filter 속성

- 포토샵 없이 코드만으로 여러 가지 시각 효과 추가
- filter: 함수

## blur() 함수

- 선택한 이미지에 블러(흐림) 효과 적용
- 픽셀 단위로 흐림 정도 지정

```
<style>
.....
  .blur { filter: blur(5px); }
</style>
.....
<div class="gallery">
  <div class="image-container">
    <div class="title">filter:blur(5px)</div>
    
  </div>
.....
</div>
```



# filter 속성과 함수들

- brightness() 함수 : 선택한 이미지를 좀더 밝게 표시. 백분율이나 숫자 사용
- contrast() 함수 : 대비 조절. 백분율 값 사용
- dropshadow() 함수 : 그림자 적용. box-shadow 속성과 같은 매개변수 사용
- grayscale() 함수 : 이미지를 회색조로 변경. 백분율 사용
- invert() 함수 : 이미지 색상 반전
- sepia() 함수 : 이미지를 세피아톤으로 변경
- opacity() 함수 : 이미지의 투명도 조절. 백분율 사용
- hue-rotate() 함수 : 이미지에 있는 색상의 hue 값 회전시킴. 각도 사용
- saturate() 함수 : 이미지의 채도 변경. 백분율 사용.

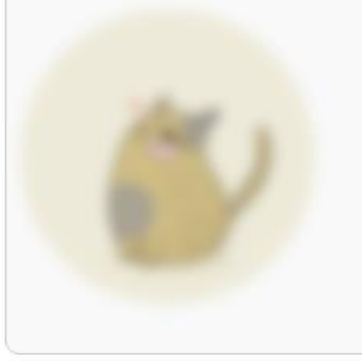
```
<style>
.....
.blur { filter: blur(5px); }          /* 흐리게 */
.brightness { filter: brightness(2); } /* 밝기 2배로 */
.contrast { filter: contrast(200%); } /* 대비 2배로 */
.drop-shadow { filter: drop-shadow(16px 16px 10px black); } /* 그림자 추가 */
.grayscale { filter: grayscale(100%); } /* 회색조 */
.invert { filter: invert(100%); }     /* 전체 색상 반전 */
.sepia { filter: sepia(100%) }        /* 세피아 톤으로 변경 */
.opacity { filter: opacity(50%); }    /* 반투명하게 */
.hue-rotate { filter: hue-rotate(90deg); } /* 모든 hue 값을 90도 이동 */
.saturate { filter: saturate(200%); } /* 채도 2배로 */
</style>
```



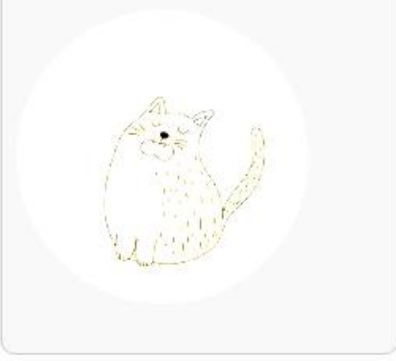
# filter 속성과 함수들



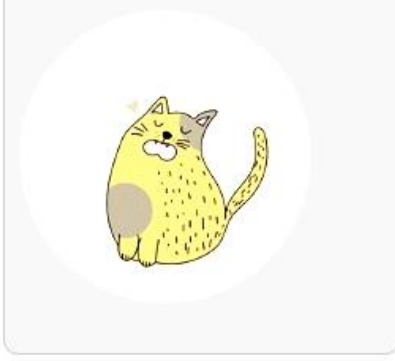
filter:blur(5px)



filter:brightness(2)



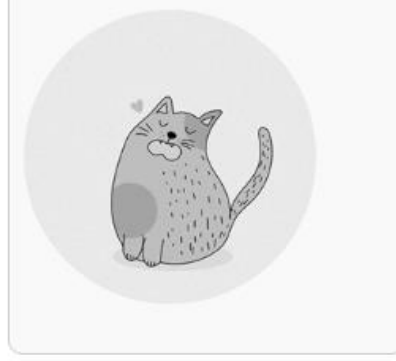
filter:contrast(200%)



filter:drop-shadow(16px 16px 10px black)



filter:grayscale(100%)



filter:invert(100%)



filter:sepia(100%)



filter:opacity(50%)



filter:hue-rotate(90deg)



filter:saturate(200%)

