

13

CSS 고급 선택자 (p398)

❖ 교재 예제 참고

- 연결 선택자
- 속성 선택자
- 가상클래스와 가상 요소
- CSS 함수

❖ 고급 선택자 : 코드의 위치나 속성값에 따라 특정요소 선택

하위 선택자

자식 선택자

인접 형제

형제 선택자

- descendent selector : 모든 하위 요소에 스타일 적용하기
적용 대상을 한정 시키려고 할 때 사용(손자 요소 까지)

```
상위요소 하위요소 {
    속성:속성값;
}
```

```
section p{
    color:red;
}
```

❖ 하위 선택자

```
<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(000-0000-0000)로 문의하시기 바랍니다.

가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일

❖ 고급 선택자 : 속성이 어디에 적용되는 지 표시

하위 선택자

자식 선택자

인접 형제

형제 선택자

- child selector : 자식 요소에만 스타일 적용하기

```
부모 요소 > 자식 요소{  
    속성:속성값;  
}
```

```
section > p{  
    color : red;  
    margin-left:20px;  
}
```

❖ 자식 선택자

```
<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(000-0000-0000)로 문의하시기 바랍니다.

가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일

❖ 고급 선택자 : 속성이 어디에 적용되는 지 표시

하위 선택자

자식 선택자

인접 형제

형제 선택자

- adjacent selector : 첫번째 동생 요소에 스타일 적용하기

```
요소1+요소2 {
    속성:속성값;
}
```

```
H1+p{
    font-size : 12px;
    color : red;
}
```

같은 부모를 가진 형제 요소 중
첫번째 동생 요소에 스타일 적용
하기(h1태그 다음에 오는 p 태그
에 적용)

❖ 인접 형제 선택자

```
<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(000-0000-0000)로 문의하시기 바랍니다.

가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일

❖ 고급 선택자 : 속성이 어디에 적용되는 지 표시

하위 선택자

자식 선택자

인접 형제

형제 선택자

- sibling selector : 모든 형제 요소에 스타일 적용하기

```
요소1~요소2 {
    속성:속성값;
}
```

```
h1~p{
    font-size : 12px;
    color : red;
}
```

~(틸드)

h1 다음에 오는 모든 p
요소에 스타일 적용

❖ 형제 선택자

```

<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(000-0000-0000)로 문의하시기 바랍니다.

가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일

❖ 속성 선택자 : 태그 안에 속성 선택자

[속성]

지정한 '속성'일 때 스타일 적용

`a[target] {background-color : skyblue;}`

[속성=값]

속성 값이 일치할 때

`a[target="_blank"] {background-color : skyblue;}`

[속성~=값]

여러 값 중 특정 값(단어)이 포함되어 있을 때

`[class ~="button"] {border-radius:10px;}`

[속성|=값]

특정 값이 포함되어 있을 때 (- 포함)

`a[href="en"] {font-style : italic;}`

[속성^=값]

캐럿(^) 지정한 값으로 시작할 때

`a[href^="http//"] {font-style : italic;}`

[속성\$=값]

지정한 값으로 끝날 때

`a[href$=".zip"] {font-style : italic;}`

[속성*=값]

해당 값이 속성값의 일부로 포함되어 있을 때

`p[class *="accent"] {font-style:italic;}`

❖ 속성 선택자

[속성] : 지정한 속성을 가진 요소를 찾아 스타일 적용

(... 생략 ...)

```
a[h ] { /* href 속성이 있는 a 요소를 찾는 선택자 */
  background: yellow; /* 배경색을 노란색으로 */
  border: 1px solid #ccc; /* 테두리 지정 */
  font-weight: normal; /* 글꼴 굵기 지정 */
}
```

</style>

.....

<a>메인 메뉴 :

메뉴 1

메뉴 2

메뉴 3

메뉴 4

a 요소 중에서 href 속성값이 있는 요소만 적용

메인 메뉴 :

메뉴 1

메뉴 2

메뉴 3

메뉴 4

❖ 속성 선택자

[속성 = 값] : 주어진 속성과 속성 값이 일치하는 요소를 찾아 스타일 적용

(... 생략 ...)

```
a[t      ="_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>


.....

HTML

CSS Selector Level 3

미디어쿼리

a 요소 중에서 href 속성값이 _blank인 요소에만 적용

HTML 

CSS Selector Level 3

미디어쿼리

❖ 속성 선택자

[속성~=값] : 여러 속성 값 중에 해당 값이 포함되어 있는 요소를 찾아 스타일 적용

(... 생략 ...)

class 속성에 button 값이 있는 요소를 찾는 선택자

```
[class ~= "b"]{
```

```
  box-shadow: rgba(0, 0, 0, 0.4) 4px 4px; /* 그림자 지정 */
```

```
  border-radius: 5px; /* 모서리 둥글게 지정 */
```

```
}
```

```
</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>
```

```
</ul>
```

(... 생략 ...)

속성값으로 button을 포함하는 요소에 적용

“button” : 버튼



❖ 속성 선택자

[속성 |= 값] : 특정 값이 포함된 속성을 가진 요소를 찾아 스타일 적용(하이픈)

외국어 서비스 :  영어  일본어  중국어

(... 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>

.....

외국어 서비스 :

영어

일본어

중국어

(... 생략 ...)

속성값이 "us-"로 시작하는 한 단어이므로 ① 스타일이 적용됨

속성값이 "jap"도, "jap-"도 아니므로
② 스타일이 적용되지 않음

속성값이 "chn"과 일치하므로 ③ 스타일이 적용됨

❖ 속성 선택자

[속성^=값] : 특정 값으로 시작하는 속성을 가진 요소를 찾아 스타일 적용

외국어 서비스 :  영어  일본어  중국어

(... 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>

.....

외국어 서비스 :

영어

일본어

중국어

(... 생략 ...)

속성값이 "eng"로 시작하므로 ① 스타일 적용

속성값이 "jap"로 시작하므로 ② 스타일 적용

속성값이 "chin"으로 시작하므로 ③ 스타일 적용

아이콘 적용

❖ 속성 선택자

[속성 \$= 값] : 특정 값으로 끝나는 속성을 가진 요소를 찾아 스타일 적용

(... 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>

속성값이 "hwp"로 끝나므로 ❶ 스타일이 적용됨

hwp 파일


엑셀 파일

속성값이 "xls"로 끝나므로
❷ 스타일이 적용됨

(... 생략 ...)

jeon:kyun

회사 소개 파일 내려받기

- hwp 파일 
- 엑셀 파일 

❖ 속성 선택자

[속성 *= 값] : 값의 일부가 일치하는 속성을 가진 요소를 찾아 스타일 적용

```
(... 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

❖ 가상 클래스(pseudo class)

- 웹 문서에 실제로 존재하지 않지만 필요에 의해 임의로 **가상의 선택자**를 지정하여 사용하는 것.
- 웹 요소를 클릭하거나 마우스 커서를 올려 놓는 등 **특정 동작에 반응**하는 스타일 작업 시 사용

링크 관련

링크와 관련된 여러 가지 **상황 정의**를 위한 선택자

위치 기준

문서 안의 요소가 **몇 번째**에 있느냐에 따라 스타일을 다르게 지정할 때 지정하는 선택자

가상 요소

실제로 존재하지 않지만 존재하는 것처럼 스타일 지정하는 가상 선택자(내용의 일부에 적용)

❖ 가상 클래스(pseudo class)

링크 관련

요소 상태

위치 기준

가상 요소

- ① `a:link {color : black; text-decoration : none;}` ← 방문 전 링크
- ② `a:visited {color: black; }` ← 방문 한 링크 스타일
- ③ `a:hover {color : red; text-decoration : underline;}`
← 마우스를 올려 놓을 때 스타일(rollover 효과)
- ④ `a:active {color: black;}` ← 마우스 누르고 있을 때 스타일
- ① ~ ④ 순서에 맞게 정의 해야 함
- ⑤ `a:focus {background-color : yellow;}` ← 초점이 맞춰졌을 때
(마우스 커서 위치, 키보드의 탭 키로 초점 이동 시 적용)

❖ 가상 클래스(pseudo class)

링크 관련

요소 상태

위치 기준

가상 요소

- :target ← 앵커의 목적지가 되는 부분의 스타일 지정
- :enabled, :disabled
텍스트 입력 영역에서 사용자가 입력할 수 있도록[없도록] 상태 지정
- :checked
라디오 버튼이나 체크박스가 체크 되었을 때 적용할 스타일 지정(css3)
- :not ← '()'안에 있는 요소를 제외한' 모든 요소에 스타일 지정
:not(h1) {color:blue;}

❖ 가상 클래스(pseudo class)

링크 관련

요소 상태

위치 기준

가상 요소

- :only-child : only-of-type
 ← 자식요소가 하나일 때, 해당 요소가 하나일 때(css3)
- :first-child : last-child ← 첫 번째[마지막] 자식요소에 지정
- :first-of-type : last-of-type ← 형제관계에서 첫 번째[마지막] 요소에 적용
- :nth-child(n) : nth-last-child(n) ← 앞[뒤]에서 n 번 째 요소 지정
- :nth-of-type(n) : nth-last-of-type(n) ← 앞[뒤]에서 n 번 째 특정 태그 지정

❖ 가상 클래스(위치 기준)

```
<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(위치 기준)

:nth-child()

- 괄호 안에 위치값을 숫자로 지정
(예) :nth-child(5)
- 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) {
  background-color: #fc2;
}
</style>
```



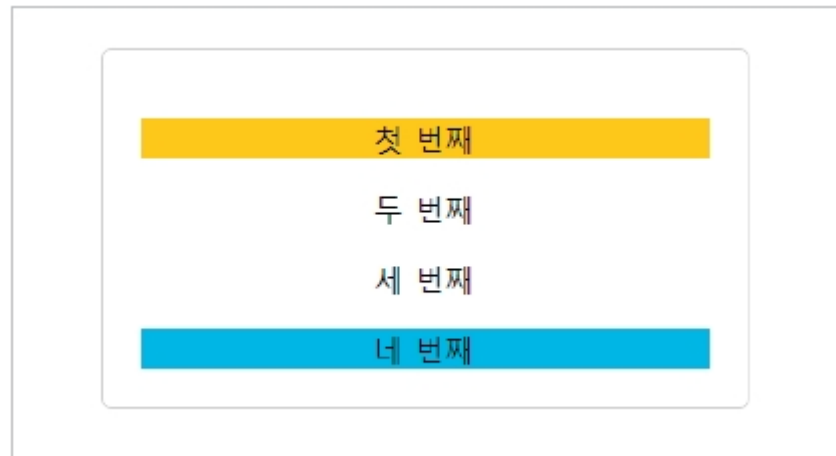
클래스	결과 화면
:nth-child(2n) 또는 :nth-child(even)	
:nth-child(2n+1) 또는 :nth-child(odd)	
:nth-child(3n)	

❖ 가상 클래스(위치 기준)

:nth-of-type()

- 같은 타입의 요소들 사이에서 순서를 지정하는 클래스

```
<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(논리적 조합)

: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(논리적 조합)

```
h1:hover, h2:hover, h3:hover {
    color : red;
}
```



```
:is(h1, h2, h3):hover {
    color : red;
}
```

A:has(B) : A 요소가 괄호 안의 B 요소를 가지고 있다면 선택

```
<style>
.....
nav a:hover {
    color: #ff7a18;
}
nav:has(a:hover) a:not(:hover) {
    opacity: 0.3;
}
</style>
```

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

❖ 가상 클래스(pseudo class)

링크 관련

요소 상태

위치 기준

가상 요소

- `::first-line`, `::first-letter` ← 요소의 첫 줄이나 첫 글자에 스타일 적용
- `::before`, `::after` ← 요소의 앞이나 뒤에 텍스트나 이미지를 추가
`p::before{content:"내용";}`
- `::selection` ← 마우스로 선택 시 커서가 드래그 되어 선택되는 스타일 지정
`::selection{color:#00ff00;}`

❖ 가상 클래스(가상 요소)

::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

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

❖ 가상 클래스(가상 요소)

::before, ::after 요소

- 웹 요소의 앞이나 뒤에 내용이나 스타일을 넣을 수 있다.
- 항상 content 속성을 사용해야 한다. 내용 없이 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
- CSS에서 변수를 사용하기 위해 사용

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 적은 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);
  /* 1rem < 2vw < 2rem */
}
</style>
```

CSS 함수

❖ CSS 함수 살펴보기

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>
```



❖ CSS 함수 살펴보기

- **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
 - * 세피아 톤 : 이미지에 색조를 추가해서 빈티지하거나 고풍스러운 느낌을 주는 효과
- **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.

❖ CSS 함수 살펴보기

<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>

❖ CSS 함수 살펴보기

