

10. CSS 고급 선택자

10-1 연결 선택자

10-2 속성 선택자

10-3 가상 클래스와 가상 요소



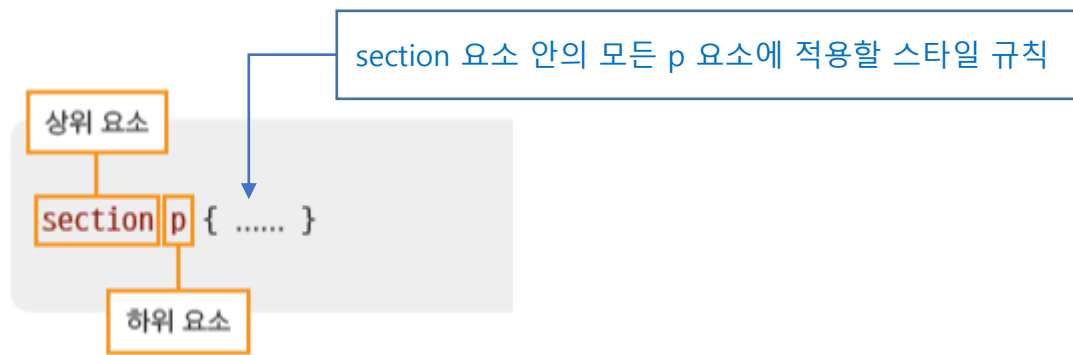
연결 선택자

- 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



Do it! 하위 선택자를 사용하여 글자색 적용하기

예제 파일 10\descendant.html

(... 생략 ...)

```
section p { color: blue; } /* section 요소의 모든 하위 p 요소를 파란색 글자로 지정 */
</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>
```

(... 생략 ...)

section의 모든 하위(자식, 손자) p 요소에 적용

예약 방법 & 사용 요금

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

가족실(2~4인) : 60,000원/일

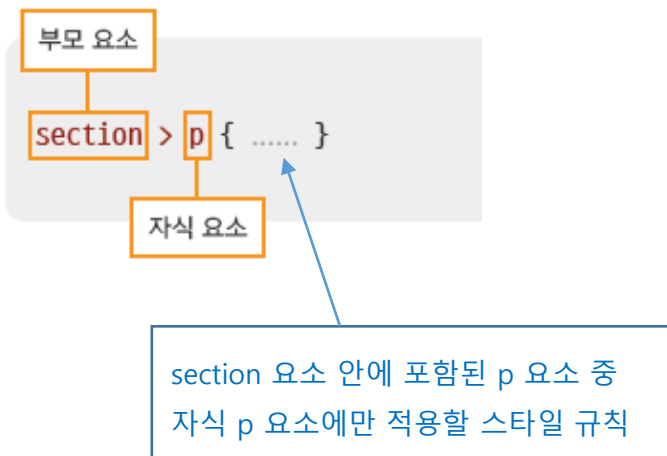
도미토리(4인 공용) : 25,000원/일

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



Do it! 자식 선택자를 사용하여 글자색 적용하기

예제 파일 10\child.html

```
(... 생략 ...)  
section > p { color: blue; } /* section 요소의 자식인 p 요소를 파란색 글자로 지정 */  
</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>  
(... 생략 ...)
```

section 요소의 자식인
p 요소에만 스타일 적용

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

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 첫번째 요소2에 적용

기본형 요소1 + 요소2

h1의 형제인 p 요소 중 첫 번째 p

`h1 + p { color: blue; }`

h1 요소 다음에 오는 p 요소들 중
첫번째 p 요소에만 파란색 글씨 적용



Do it! 인접 형제 요소에 스타일 적용하기

예제 파일 10\adj.html

(... 생략 ...)

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

</style>

.....

<h1>예약 방법 & 이용 요금</h1>

<p>아직 온라인 예약.....</p>

h1 요소의 형제 요소 중에서 첫 번째 형제인 p 요소에 적용

<p>가족실(2~4인) : 60,000원/일</p>

<p>도미토리(4인 공용) : 25,000원/일</p>

(... 생략 ...)

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

h1의 형제인 모든 p 요소

`h1 ~ p { color: blue; }`

h1 요소 다음에 오는
모든 형제 p 요소에 밀줄 적용



모든 형제 요소의 스타일 지정하기

예제 파일 10\sibling.html

```
(... 생략 ...)  
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>  
(... 생략 ...)
```

h1 요소의 형제인 p 요소에 모두 적용

예약 방법 & 이용 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

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>  
    <li><a href="#" class="flat button">메뉴 4</a></li>  
</ul>
```

속성값이 button인 요소에 적용

속성값으로 button을 포함하는 요소에 적용

(... 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>  
    <li><a href="#" title="ja">일본어</a></li>  
    <li><a href="#" title="chn">중국어</a></li>  
</ul>
```

속성값이 "us-"로 시작하는 한 단어이므로 ❶ 스타일이 적용됨

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

속성값이 "chn"과 일치하므로 ❸ 스타일이 적용됨

(... 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 ...)

```
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 */
}

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 */
}

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 ...)

```
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 */
}

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 

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 클래스와 가상 요소

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



(... 생략 ...)

```
.navi ul li {
  float: left;
  width: 150px;
  padding: 10px;
}

.navi a:link, .navi a:visited { /* 방문한 링크와 방문하지 않은 링크 설정 */
  display: block;
  font-size: 14px;
  color: #000;
  padding: 10px;
  text-decoration: none; /* 밑줄을 없앴 */
  text-align: center;
}

.navi a:hover, .navi a:focus { /* 마우스 포인터를 올렸을 때와 초점을 맞췄을 때 설정 */
  background-color: #222; /* 배경색을 짙은 회색으로 변경 */
  color: #fff;
}

.navi a:active { /* 마우스 포인터로 클릭했을 때 지정 */
  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



Do it! 앵커 대상에 스타일 적용하기

예제 파일 10\ps-1.html

(... 생략 ...)

```
#intro:target {  
  background-color: #0069e0; /* 배경색을 파란색으로 지정 */  
  color: #fff; /* 글자색을 흰색으로 지정 */  
}
```

</style>

.....

```
<nav class="navi">  
  <ul>  
    <li><a href="#intro">이용 안내</a></li>  
    <li><a href="#room">객실 소개</a></li>  
    <li><a href="#reservation">예약 방법 및 요금</a></li>  
    <li><a href="ps-2.html">예약하기</a></li>  
  </ul>  
</nav>
```

```
<div id="intro" class="contents">  
  <h2>이용 안내</h2>
```

.....

```
</div>  
<div id="room" class="contents">  
  <h2>객실 소개</h2>
```

.....

```
</div>  
<div id="reservation" class="contents">  
  <h2>예약 방법 및 요금</h2>
```

.....

```
</div>
```

(... 생략 ...)

이제! : 응용

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)



Do it! 선택된 라디오 버튼의 스타일 적용하기

예제 파일 10\ps-2.html

(... 생략 ...)

```
#signup input:checked + label { /* input 요소에 checked 속성이 추가되었을 때 label  
요소의 스타일 */
```

```
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>
```

(... 생략 ...)

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

:not – 특정 요소를 제외하고 스타일 적용

type="text" 필드와 type="tel" 필드에 스타일 적용하기

개인 정보

이름

연락처

객실 형태

☐ 기본형(최대 2인)

☐ 가족형(최대 8인)

제출

일반적인 방법



Do it! 일반적으로 텍스트 필드 선택하기

예제 파일 10\ps-2.html

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

:not 선택자 사용



Do it! not 선택자로 라디오 필드를 제외한 텍스트 필드 선택하기

예제 파일 10\ps-2.html

```
(... 생략 ...)  
#signup input:not([type=radio]) {  
  border: 1px solid #ccc;  
  border-radius: 3px;  
  padding: 5px;  
  width: 200px;  
}  
(... 생략 ...)
```

이제: 10\ps-2.html

가상 클래스와 가상 요소

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.

종류	설명
:only-child	부모 안에 자식 요소가 하나뿐일 때 자식 요소를 선택합니다.
A:only-type-of	부모 안에 A 요소가 하나뿐일 때 선택합니다.
:first-child	부모 안에 있는 모든 요소 중에서 첫 번째 자식 요소를 선택합니다
:last-child	부모 안에 있는 모든 요소 중에서 마지막 자식 요소를 선택합니다
A:first-of-type	부모 안에 있는 A 요소 중에서 첫 번째 요소를 선택합니다.
A:last-of-type	부모 안에 있는 A 요소 중에서 마지막 요소를 선택합니다.
:nth-child(n)	부모 안에 있는 모든 요소 중에서 n 번째 자식 요소를 선택합니다.
:nth-last-child(n)	부모 안에 있는 모든 요소 중에서 끝에서 n 번째 자식 요소를 선택합니다.
A:nth-of-type(n)	부모 안에 있는 A 요소 중에서 n 번째 요소를 선택합니다.
A:nth-last-of-type(n)	부모 안에 있는 A 요소 중에서 끝에서 n 번째 요소를 선택합니다.



Do it! 위치에 따라 자식 요소 선택하기

예제 파일 10\ps-3.html

(... 생략 ...)

```
1 .contents :nth-child(3) { /* .contents의 세 번째 자식 요소에 스타일 적용 */  
  background-color: #ffff00; /* 배경색을 노란색으로 */  
}  
2 .contents p:nth-of-type(3) { /* .contents의 p 요소 중에서 세 번째 자식 요소에 스타일 적용 */  
  background-color: #00b900; /* 배경색을 초록색으로 */  
}
```

</style>

.....

<div class="contents">

<h2>이용 안내</h2>

<p>Excepteur do </p>

<p>Qui magna culpa </p>

<h2>객실 소개</h2>

<p>Irure incididunt </p>

<h2>예약 방법 및 요금</h2>

<p>Fugiat aliquip </p>

</div>

이용 안내

Excepteur do est eiusmod nulla et veniam. Labore officia officia ex aliqua exercitation aliqua laborum Lorem deserunt ut ullamco labore anim. Officia eu duis aliquip incididunt. Do laborum et consequat aliqua sint consectetur.

Qui magna culpa minim reprehenderit magna in nisi ipsum. Ad cillum tempor minim fugiat est dolor. Cillum sit qui minim sint officia nostrud cillum cupidatat pariatur ipsum eiusmod velit labore. Sit in non fugiat minim sit.

객실 소개

Irure incididunt duis ut cupidatat laborum adipisicing veniam irure.

예약 방법 및 요금

Fugiat aliquip mollit proident velit magna esse ea officia eu. Esse do aliqua proident culpa eiusmod duis minim deserunt eu reprehenderit ut tempor.

가상 클래스와 가상 요소

가상 요소

- 화면 꾸미기용 요소를 웹 문서에 포함시키지 않기 위해 가상 요소 사용
- ::first-line : 특정 요소의 첫번째 줄에 스타일 적용
- ::first-letter : 특정 요소의 첫번째 글자에 스타일 적용
- ::before : 특정 요소의 앞에 지정한 콘텐츠 추가
- ::after : 특정 요소의 뒤에 지정한 콘텐츠 추가



Do it! 가상 요소를 사용해 스타일 적용하기

(... 생략 ...)

```
li.new::after {  
  content: "NEW!!";  
  font-size: x-small;  
  padding: 2px 4px;  
  margin: 0 10px;  
  border-radius: 2px;  
  background: #f00;  
  color: #fff;  
}
```

</style>

.....

```
<div class="container">  
  <h1>제품 목록</h1>  
  <ul>  
    <li class="new">제품 A</li>  
    <li>제품 B</li>  
    <li>제품 C</li>  
    <li class="new">제품 D</li>  
  </ul>  
</div>  
(... 생략 ...)
```

제품 목록

- 제품 A **NEW!**
- 제품 B
- 제품 C
- 제품 D **NEW!**