

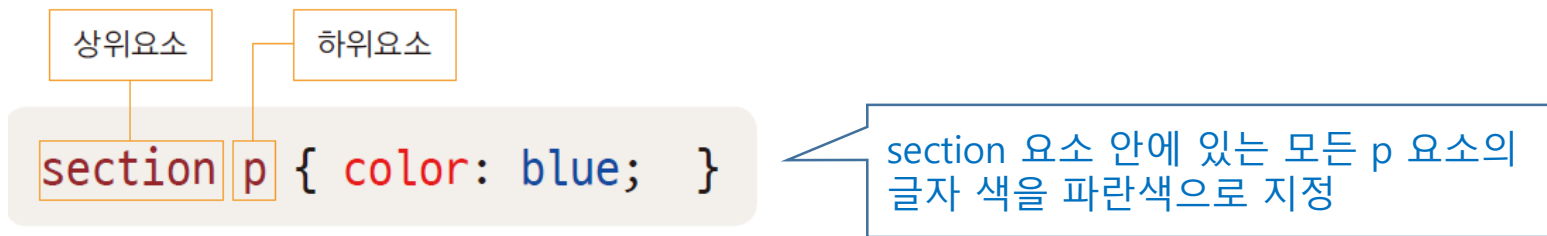
[illegible]

- 연결 선택자 : 선택자와 선택자를 연결해 적용 대상을 한정하는 선택자.
- 콤비네이션 선택자(combination selector) 또는 '조합 선택자'라고도 함

하위 선택자

기본형 상위요소 하위요소

- 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.

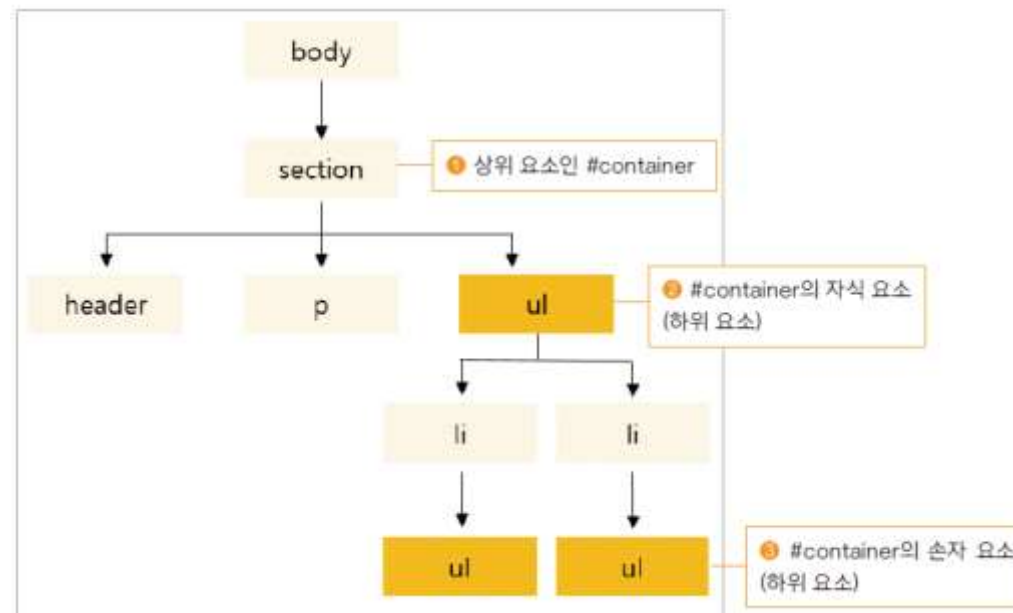


하위 선택자

```

<style>
  #container ul {
    border: 1px dotted blue;
  }
</style>
<section id="container">
  <header><h1>예약 방법 및 요금</h1></header>
  <p> 요안도라에 예약하려면?
  <ul>
    <li> 예약 방법
      <ul>
        <li>직접 통화</li>
        <li>문자 남기기</li>
      </ul>
    </li>
    <li>요금
      <ul>
        <li>1인 : 40,000원</li>
        <li>2인 : 60,000원</li>
        <li>3인 : 80,000원</li>
        <li>4인 : 100,000원</li>
      </ul>
    </li>
  </ul>
</section>

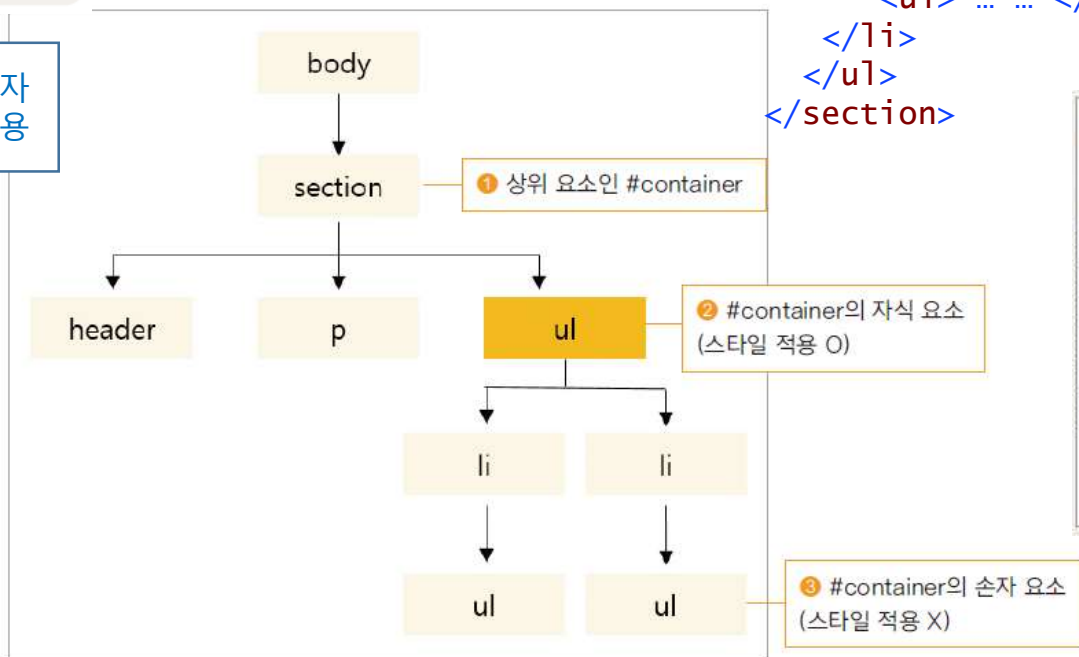
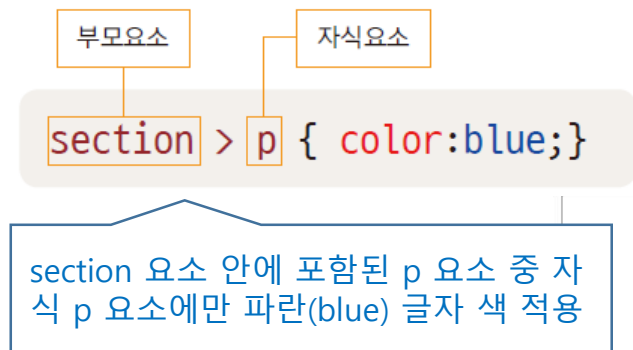
```



자식 선택자

기본형 부모요소 > 자식요소

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



```

<style>
  #container > ul {
    border: 1px dotted blue;
  }
</style>
<section id="container">
  <header><h1>예약 방법 및 요금</h1></header>
  <p> 요안도라에 예약하려면?
  <ul>
    <li> 예약 방법
      <ul> ... .. </ul>
    </li>
    <li>요금
      <ul> ... .. </ul>
    </li>
  </ul>
</section>
  
```

예약 방법 및 요금

요안도라에 예약하려면?

- 예약 방법
 - 직접 통화
 - 문자 남기기
- 요금
 - 1인 : 40,000원
 - 2인 : 60,000원
 - 3인 : 80,000원
 - 4인 : 100,000원

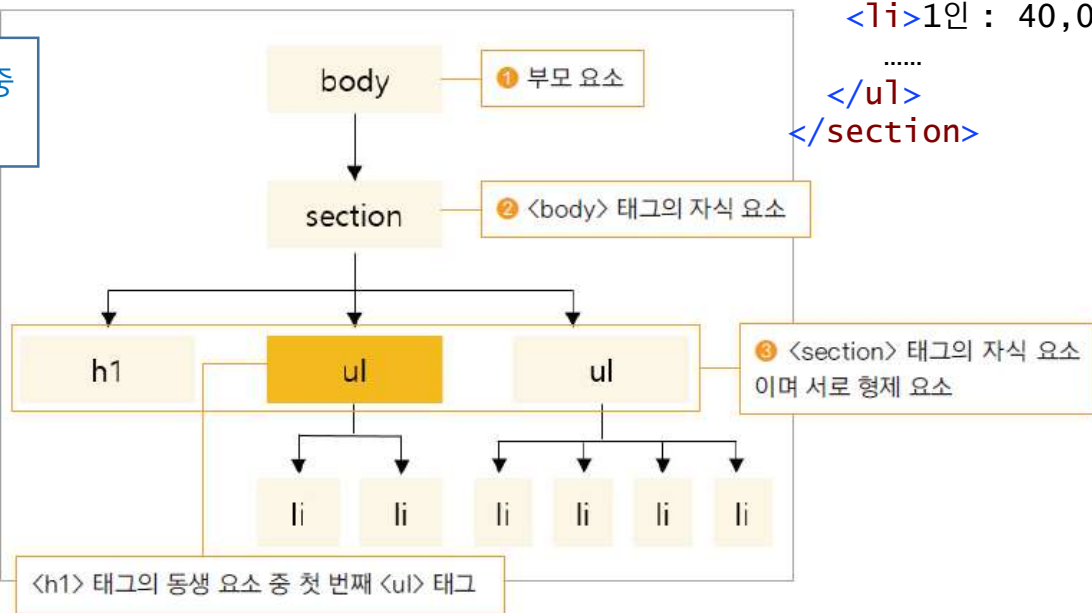
인접 형제 선택자

기본형 요소 1 + 요소 2

- 같은 부모를 가진 형제 요소 중 첫 번째 동생 요소에만 스타일 적용
- 요소 1과 요소 2는 같은 레벨이면서 요소 1 이후 맨 먼저 오는 요소 2에 스타일을 적용

```
h1 + p { text-decoration: underline; }
```

h1 요소 다음에 오는 p 요소들 중 첫번째 p 요소에만 밑줄적용



```

<style>
  h1 + ul {
    color: blue;
    font-weight: bold;
  }
</style>

<section id="container">
  <h1>예약 방법 및 요금</h1>
  <ul>
    <li>직접 통화</li>
    <li>문자 남기기</li>
  </ul>
  <ul>
    <li>1인 : 40,000원</li>
    .....
  </ul>
</section>

```

예약 방법 및 요금

- 직접 통화
- 문자 남기기
- 1인 : 40,000원
- 2인 : 60,000원
- 3인 : 80,000원
- 4인 : 100,000원

형제 선택자

기본형

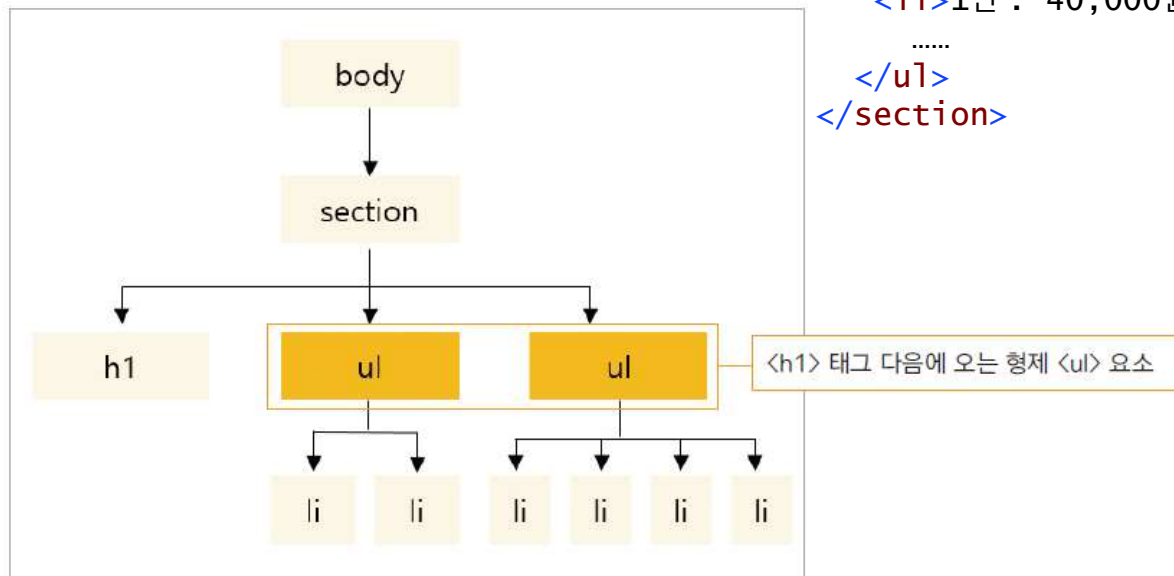
요소 1 ~ 요소 2

- 형제 요소들에 스타일 적용
- 인접 형제 선택자와 다른 점은 모든 형제 요소에 다 적용된다는 것

형제 요소

```
h1 ~ p { text-decoration: underline; }
```

h1 요소 다음에 오는 모든
형제 p 요소에 밑줄 적용



```

<style>
  h1 ~ ul {
    color: blue;
    font-weight: bold;
  }
</style>

<section id="container">
  <h1>예약 방법 및 요금</h1>
  <ul>
    <li>직접 통화</li>
    <li>문자 남기기</li>
  </ul>
  <ul>
    <li>1인 : 40,000원</li>
    .....
  </ul>
</section>

```

예약 방법 및 요금

- 직접 통화
- 문자 남기기
- 1인 : 40,000원
- 2인 : 60,000원
- 3인 : 80,000원
- 4인 : 100,000원

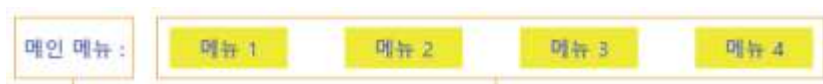
[속성] 선택자

지정한 속성을 가진 요소를 찾아 스타일 적용

```
<style>
  a[href] {
    background: yellow;
  }
</style>
```

<a> 태그 중 href라는 속성이 있는
요소를 찾아내 배경 색 지정

```
<body>
  <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>
</body>
```



[속성 = 값] 선택자

주어진 속성과 속성 값이 일치하는 요소를 찾아 스타일 적용

```
<style>
  a[target="_blank"] {
    padding-right: 30px;
    background: url(images/newwindow.png) no-repeat center
    right;
  }
</style>
```

```
<h1>HTML5 표준안 사이트 </h1>
<ul>
  <li><a href="http://www.w3c.org/TR/html"
  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">미디어
  쿼리</a></li>
</ul>
```

HTML5 표준안 사이트

HTML

CSS Selector Level 3

미디어쿼리



[속성~=값] 선택자

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

```
<style>
  [class ~= "button"] {
    border: 2px solid black;
    box-shadow: rgba(0,0,0,0.4) 5px 5px;
  }
</style>
```

class 속성 값에 button이 포함된 요
소를 찾아내 스타일 적용

```
<body>
  <ul>
    <li><a href="#">메뉴 1</a></li>
    <li><a href="#">메뉴 2</a></li>
    <li><a href="#" class="button">메뉴
3</a></li>
    <li><a href="#" class="flat button">메뉴
4</a></li>
  </ul>
</body>
```



[속성|=값] 선택자

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

```
<style>
  a[title |= "us"] {
    background: url(images/us.png) no-repeat left center;
    padding: 5px 25px;
  }
  a[title |= "jap"] {
    background: url(images/jp.png) no-repeat left center;
    padding: 5px 25px;
  }
</style>
```

```
<ul>
  <li>외국어 서비스 : </li>
  <li><a href="#" title="us">영어</a></li>
  <li><a href="#" title="us-english">영어</a></li>
  <li><a href="#" title="japanese">일본어</a></li>
</ul>
```

외국어 서비스 :  영어  영어 일본어

[속성 ^= 값] 선택자

특정 값으로 시작하는 속성을 가진 요소를 찾아 스타일 적용

```
<style>
  a[title ^= "eng"] {
    background: url(images/us.png) no-repeat left center;
    padding: 5px 25px;
  }
  a[title ^= "jap"] {
    background: url(images/jp.png) no-repeat left center;
    padding: 5px 25px;
  }
  a[title ^= "chin"] {
    background: url(images/ch.png) no-repeat left center;
    padding: 5px 25px;
  }
</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>
```

외국어 서비스 :  영어  일본어  중국어



[속성 \$= 값] 선택자

특정 값으로 끝나는 속성을 가진 요소를 찾아 스타일 적용

```
<style>
  a[href $= "hwp"] {
    background: url(images/hwp_icon.gif) center right no-repeat;
    padding-right: 25px;
  }
  a[href $= "xls"] {
    background: url(images/excel_icon.gif) center right no-repeat;
    padding-right: 25px;
  }
</style>

<ul>
  <li><a href="intro.hwp">hwp 파일</a></li>
  <li><a href="intro.xls">엑셀 파일</a></li>
</ul>
```

회사 소개 파일 다운 받기

- hwp 파일 
- 엑셀 파일 

[속성*=값] 선택자

값의 일부가 일치하는 속성을 가진 요소를 찾아 스타일 적용

```
<style>
  a[href *= "w3"] {
    background: blue;
    color: white;
  }
</style>

<ul>
  <li><a href="http://www.w3c.org/TR/html">HTML 표준안 사이트
</a></li>
  <li><a href="http://www.webplatform.org">튜토리얼과 아티클
</a></li>
  <li><a href="http://caniuse.com">HTML 지원 여부 체크</a></li>
  <li><a href="http://www.w3c.org/TR/css3-mediaqueries">미디어 쿼리</a></li>
</ul>
```

기본형	스타일 적용 경우	예시 - 선택자	예시 - 적용되는 요소
[속성]	지정한 '속성' 일 때	[href]	메뉴 1
[속성 = 값]	'값'에 일치할 때	[target="_blank"]	HTML
[속성 ~= 값]	여러 값들 중 특정 '값'이 포함되어 있을 때(단어별)	[class~="button"]	메뉴 4
[속성 = 값]	'값'이 포함되어 있을 때 (하이픈 포함 단어별)	[class = "us"]	영어 영어
[속성 ^= 값]	'값'으로 시작할 때	[title^="eng"]	영어
[속성 \$= 값]	'값'으로 끝날 때	[href \$="xls"]	엑셀 파일
[속성 *= 값]	'값'이 속성 값의 일부 일 때	[href *= "w3"]	HTML 사이트

가상 클래스와 가상 요소

사용자 동작에 반응하는 가상 클래스

표기	설명
: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



```

<style>
  .navi a:link, .navi a:visited {
    padding: 10px 5px 5px 35px;
    display: block;
    color: #fff;
    width: 150px;
    text-decoration: none;
  }
  .navi a:hover, .navi a:focus {
    text-shadow: 0px 2px 2px #000;
    color: #FC0;
  }
  .navi a:active {
    color: red;
  }
</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

UI 요소 상태에 따른 가상 클래스

표기	설명
:enabled, :disabled	요소를 사용할 수 있을 때와 없을 때의 스타일 지정
:checked	라디오 박스나 체크 박스에서 항목을 선택했을 때의 스타일 지정

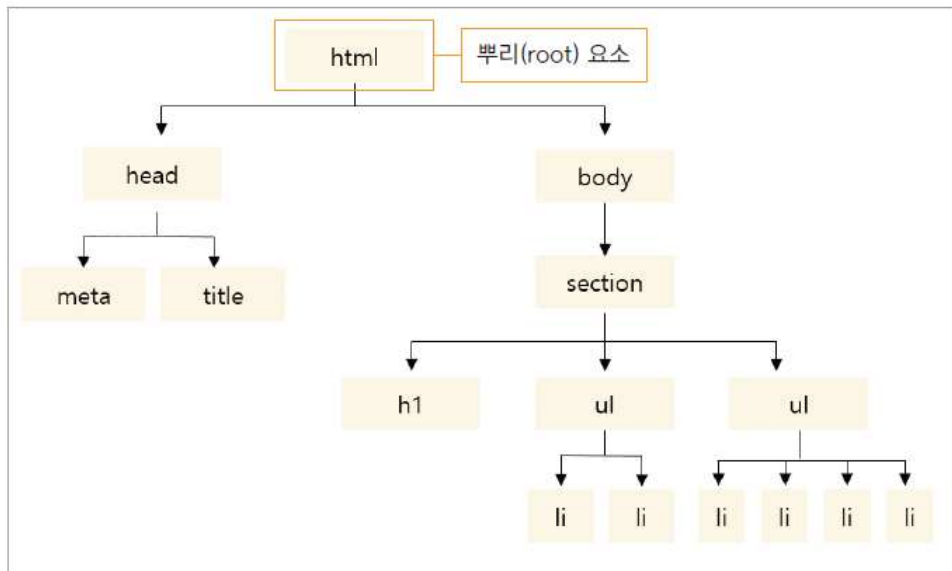
```
<style>
  input:disabled {
    background:#ddd;
    border:1px #ccc solid;
  }
  input:checked + span {
    color:blue;
  }
</style>
```

```
<p>이 달에 신청할 과목을 선택하세요</p>
<label>
  <input type="radio" name="subject"
  value="speaking"><span>회화</span>
</label>
<label>
  <input type="radio" name="subject"
  value="grammar"><span>문법</span>
</label>
<label>
  <input type="radio" name="subject"
  value="writing"><span>작문</span>
</label>
```

구조 가상 클래스

:root

- 문서 안의 루트(root) 요소에 스타일을 적용
- 웹 문서의 최상위 요소는 <html> 이므로 이 스타일을 이용하면 웹 문서 전체에 스타일 적용



가상 클래스와 가상 요소

구조 가상 클래스

:nth-child(n)와 :nth-last-child(n)

- :nth-child(n) : 앞에서부터 n번째 자식 요소에 스타일 적용
- :nth-last-child(n) : 뒤에서부터 n번째 자식 요소에 스타일 적용
- 위치를 나타낼 때 $an+b$ 처럼 수식을 사용할 수도 있음.
이 때 n 값은 0부터.

```
/* div 요소 안에서 세 번째 자식 요소인 p 요소에 스타일 적용 */
div p:nth-child(3)

/* div 요소 안에서 홀수 번째로 나타나는 자식 요소인 p 요소에 스타일 적용 */
div p:nth-child(odd), div p:nth-child(2n+1)

/* div 요소 안에서 짝수 번째로 나타나는 자식 요소인 p 요소에 스타일 적용 */
div p:nth-child(even), div p:nth-child(2n+0), div p:nth-child(2n)
```

```
<style>
  table tr:nth-child(2n+1) {
    background: lightgray;
    color: black;
  }
</style>
```

```
<table border="1">
  <tr> <td>블루베리</td></tr>
  <tr> <td>귀리</td></tr>
  <tr> <td>토마토</td></tr>
  .....
</table>>
```

건강에 좋은 건강 식품

블루베리
귀리
토마토
시금치
적포도주
견과류
브로콜리
연어
마늘
녹차

가상 클래스와 가상 요소

구조 가상 클래스

:nth-of-type(n), :nth-last-of-type(n)

- :nth-of-type(n) : 앞에서부터 n번째 요소에 스타일 적용
- :nth-last-of-type(n) : 뒤에서부터 n번째 요소에 스타일 적용

:first-child, :last-child

- :first-child: 첫번째 자식 요소스타일 적용
- :last-child: 마지막 자식 요소에 스타일 적용

```
<style>
  ul.navi li:first-child {
    border-top-left-radius: 1em;
    border-bottom-left-radius: 1em;
  }
  ul.navi li:last-child {
    border-top-right-radius: 1em;
    border-bottom-right-radius: 1em;
  }
</style>
```

```
<ul class="navi">
  <li class="home"><a
href="#">Home</a></li>
  <li id="html"><a
href="#">HTML5</a></li>
  <li id="css"><a
href="#">CSS3</a></li>
  <li id="jquery"><a
```

Home

HTML5

CSS3

jQuery

가상 클래스와 가상 요소

구조 가상 클래스

:first-of-type(n), :last-of-type(n)

- :first-of-type : 형제 요소들 중 첫번째 요소에 스타일 적용
- :last-of-type : 형제 요소들 중 마지막 요소에 스타일 적용

```
p:first-of-type { color: blue; }  
p:last-of-type { color: red; }
```

:only-child, :only-of-type

- :only-child : 부모 요소 안에 자식 요소가 유일하게 하나일 때 스타일 적용
- :only-of-type : 자신이 유일한 요소일 때 스타일 적용

```
p:only-child { color: green; }
```

자식 요소가 오직 p 요소 뿐일 때 p의 글자색을 green으로
(다른 자식 요소가 있으면 안됨)

```
p:only-of-type { font-weight: bold; }
```

p 요소가 오직 하나 뿐일 때 (다른 자식 요소 있어도 됨)

가상 클래스와 가상 요소

그 외 가상 클래스

:target

앵커로 연결된 부분(목적지)에 스타일 지정

```
#intro:target { background-color: yellow; }
```

#intro 앵커가 연결하는 부분의 배경 색을 노랑으로

:not

괄호 안에 있는 요소를 제외한 부분에 스타일 지정

```
p:not(#ex) { color: blue; }
```

#ex가 아닌 모든 p 요소에서 글자색을 파랑으로

:only-child, :only-of-type

- :only-child : 부모 요소 안에 자식 요소가 유일하게 하나일 때 스타일 적용
- :only-of-type : 자신이 유일한 요소일 때 스타일 적용

```
p:only-child { color: green; }
```

자식 요소가 오직 p 요소 뿐일 때 p의 글자색을 green으로
(다른 자식 요소가 있으면 안됨)

```
p:only-of-type { font-weight: bold; }
```

p 요소가 오직 하나 뿐일 때 (다른 자식 요소 있어도 됨)

가상 클래스와 가상 요소

표기	설명
:target	앵커 목적지에 스타일 적용
:not	특정 요소가 아닐 때 스타일 적용하기
:nth-last-child(n)	부모 요소의 뒤로부터 n번째의 자식 요소에 스타일을 적용한다.
:nth-of-type(n)	같은 유형의 요소 중에서 n 번째 나타나는 요소에 스타일을 적용한다.
:nth-last-of-type(n)	같은 유형의 요소 중에서 끝에서부터 세어 n번째 나타나는 요소에 스타일을 적용한다.
:first-child	첫번째 자식 요소에 스타일을 적용한다.
:last-child	마지막 자식 요소에 스타일을 적용한다.
:first-of-type	형제 중에서 첫번째 요소에 스타일을 적용한다.
:last-of-type	형제 중에서 마지막 요소에 스타일을 적용한다.
:only-child	해당 요소가 유일한 자식 요소일 때 스타일을 적용한다.
:only-of-type	해당 요소가 하나 뿐인 요소에 스타일을 적용한다.

가상 클래스와 가상 요소

가상 요소

- `::first-line` : 특정 요소의 첫번째 줄에 스타일 적용
- `::first-letter` : 특정 요소의 첫번째 글자에 스타일 적용
- `::before` : 특정 요소의 앞에 지정한 내용을 추가
- `::after` : 특정 요소의 뒤에 지정한 내용을 추가