

11. HTML5와 멀티미디어

11-1 웹과 멀티미디어

11-2 오디오 & 비디오 재생하기



웹과 멀티미디어

플러그인 프로그램

- HTML4까지는 웹 브라우저에서 멀티미디어를 직접 재생할 수 없기 때문에 플러그인 프로그램 연결해서 사용
- HTML5 웹 표준 이후 웹 브라우저에서 직접 멀티미디어 재생
- 일부 사이트에서는 아직도 플래시 플레이어 사용 중

<object>, <embed> - 외부 파일 삽입

기본형 `<object data="경로" type="유형" name="이름" width="너비" height="높이"]>`
`</object>`

<object> 태그를 지원하지 않는 브라우저에서는 <embed> 태그 사용

기본형 `<embed src="경로" type="유형" width="너비" height="높이">`

웹과 멀티미디어

웹에서 지원하는 비디오/오디오 파일

웹에서 지원되는 비디오/오디오 파일에는 여러 종류가 있습니다. 뒤이어 배울 코덱으로 비디오/오디오 파일을 제작할 수 있으며 파일 확장자도 바꿀 수 있습니다.

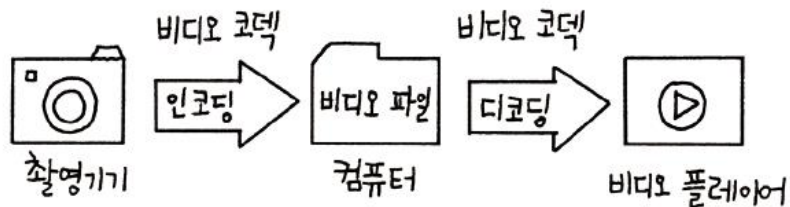
종류	확장자	설명
비디오	mp4	고화질 영상을 지원해 많은 사이트에서 사용되고 있습니다. 라이선스가 있으므로 웹 용을 제외하면 유료로 사용해야 합니다.
	webm	화질이 우수하고 무료로 제공되어 최근 많이 사용되고 있습니다.
	ogv	화질은 다른 비디오 유형보다 떨어지지만 무료라는 장점 때문에 webm 형식이 등장하기 전에 많이 사용되었습니다.
오디오	mp3	대부분의 음원에서 사용되지만 라이선스 때문에 유료로 사용해야 합니다.
	ogg	무료이므로 게임 등에서 많이 사용합니다.

최신 모던 브라우저를 사용한다면 mp4와 mp3를 사용하면 간단하지만, 사용자들의 브라우저는 다양하기 때문에 여러 종류의 파일 형식을 지정해야 한다.

웹과 멀티미디어

HTML5와 비디오 코덱

- 인코딩(encoding) : 원본 비디오를 컴퓨터에서 사용할 수 있는 비디오 파일로 변환
- 디코딩(decoding) : 컴퓨터 비디오 파일에 있는 비디오 정보를 가져와 플레이어에 보여주는 과정
- 비디오 코덱 : 인코딩과 디코딩 수행. HTML5에서는 브라우저에서 직접 재생할 수 있는 비디오 코덱만 허용.



① H.264/AVC

- 고화질의 영상. mp4 파일, mov 파일 등에서 사용
- 유료 코덱이지만 온라인에서 사용할 경우 무료로 사용 가능.
- 대부분 모던 브라우저에서 지원

② 오그 테오라(Ogg Theora)

- 공개 코덱. ogv 파일 형식에서 사용
- 파이어폭스와 오페라, 크롬에서 지원.

③ v8, v9

- 오픈 소스로 공개한 코덱. webm 파일에서 사용
- 화질이 좋고 무료로 제공
- 파이어폭스와 오페라, 크롬 등에서 지원.

- 대부분의 브라우저에 H.264 코덱을 지원하므로 mp4 파일을 기본적으로 사용
- 무료이면서 최신 코덱인 v9 코덱을 이용한 webm 파일도 함께 사용

웹과 멀티미디어



HTML5와 오디오 코덱

① MPEG-1 AUDIO Layer3 (MP3 코덱)

- 가장 많이 사용하는 오디오 코덱
- mp3 파일에서 사용
- 특허권이 등록되어 있어 유료

② 오그 보비스(Ogg Vorbis)

- 공개 코덱. ogg 파일 형식에서 사용
 - 재생 플레이어가 적고 인코딩 시간이 더 걸린다는 단점.
- but, 무료라서 PC 게임 등에 많이 사용됨

오디오 & 비디오 재생하기

<audio> 태그

- 배경 음악이나 효과음 등 오디오 재생
- 대부분 브라우저에서 mp3 지원하므로 mp3 파일만 사용

기본형 <audio src="오디오 파일 경로" [속성] [속성="속성 값"]></audio>

속성	설명
autoplay	오디오를 자동 재생합니다.
controls	웹 화면에 컨트롤 막대를 표시합니다. 컨트롤 막대에는 재생/멈춤, 진행 바, 볼륨 등이 표시됩니다.
loop	오디오를 반복 재생합니다.
muted	오디오를 재생해 진행하지만 소리는 끕니다.
preload	재생 버튼을 눌러 재생하기 전에 오디오 파일을 다운로드해 준비해 둡니다.

<video> 태그

웹 문서에 비디오 파일 삽입

기본형 <video src="비디오 파일 경로" [속성] [속성="속성 값"]></video>

❖ 2018년 1월부터 크롬 브라우저에서 오디오 자동 재생 차단

오디오 & 비디오 재생하기

<source> 태그

사용자들의 브라우저 환경을 고려해서 최신 브라우저와 이전 브라우저에서 모두 재생할 수 있도록 여러 코덱의 파일을 함께 지정

```
<source src="video.ogv" type="video/ogg; codecs='theora,vorbis'" >
```

속성	설명
src	미디어 파일의 경로를 지정하는 필수 속성으로 파일 경로를 지정할 때는 경로에 공백이 있으면 안 됩니다.
type	웹 브라우저가 해당 미디어 파일을 재생할 수 있는지 여부를 확인할 수 있도록 미디어 파일의 유형을 알려 줍니다.
codecs	비디오 코덱을 지정합니다.

```
<video controls>
  <source src="media/Painting.mp4" type="video/mp4">
  <source src="media/Painting.webm" type="video/webm">
</video>
```

HTML5를 지원하지 않는 브라우저에서는?

- HTML5 지원 브라우저가 필요하다는 대체 텍스트 표시

```
<video controls>
  <source src="media/Painting.mp4" type="video/mp4">
  <source src="media/Painting.webm" type="video/webm">
  이 영상을 보기 위해서는 HTML5를 지원하는 브라우저가 필요합니다.
</video>
```

- 플래시 무비로 변환한 후 <embed> 태그나 <object> 태그로 삽입

```
<video controls>
  <source src="media/Painting.mp4" type="video/mp4">
  <source src="media/Painting.webm" type="video/webm">
  <object data="media/Painting.swf" type="application/x-shockwave-flash"> </object>
</video>
```

오디오 & 비디오 재생하기



<audio>, <video> 태그의 속성

- **width, height** : 비디오 크기 조절
- **controls** : 미디어 파일에 컨트롤 막대 표시. 표시되는 컨트롤 막대는 웹 브라우저마다 다르며, 사용할 수 있는 명령도 조금씩 다르다.
`<video src="media/Painting.mp4" controls></video>`
- **preload** : 재생하기 전에 비디오 파일을 모두 다운로드할 것인지 또는 일부 정보만 다운로드할 것인지 여부를 지정한다.
 - none – 재생 버튼을 눌러야 다운로드하기 시작 `<video src="media/Painting.mp4" controls preload="none"></video>`
 - metadata – 미디어 파일 전체를 다운로드하지 않고 메타 정보만 다운로드 `<video src="media/Painting.mp4" controls preload="metadata"></video>`
 - auto – 웹 문서를 로드할 때 미디어 파일도 모두 다운로드. 기본값 `<video src="media/Painting.mp4" controls preload="auto"></video>`
- **muted** : 비디오를 재생할 때 소리는 끄고 화면만 재생. 비디오를 문서 배경으로 사용하거나 소리가 필요하지 않을 때 사용
- **autoplay** : 미디어 파일을 다운로드 하자마자 자동으로 재생. 크롬 브라우저와 모바일 기기에서는 자동 재생되지 않는다.
- **loop** : 미디어 파일 반복 재생. 속성 값 없이 loop 라고 하면 된다.
- **poster** : 포스터 이미지(비디오를 재생할 수 없을 경우 비디오 화면 자리에 대신 표시하는 이미지) 지정
`<video src="media/Painting.mp4" controls poster="fireworks.jpg"></video>`

[실습] 웹 문서에 멀티미디어 파일 넣기



12. 다재다능한 CSS3 선택자

12-1 연결 선택자

12-2 속성 선택자

12-3 가상 클래스와 가상 요소



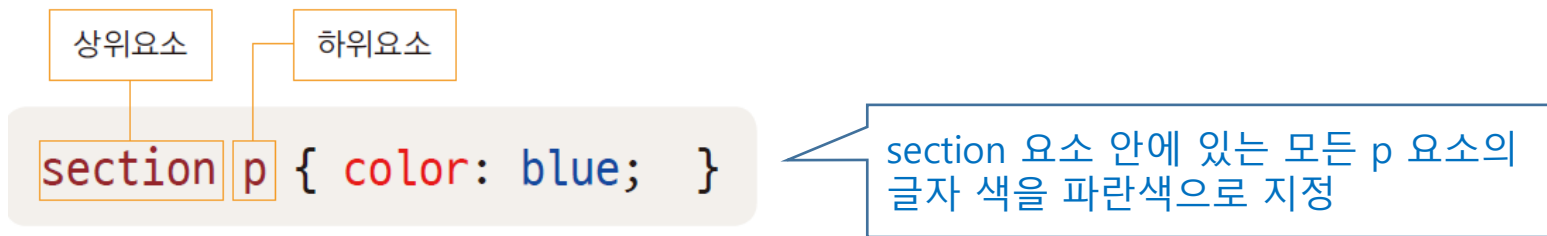
연결 선택자



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

하위 선택자

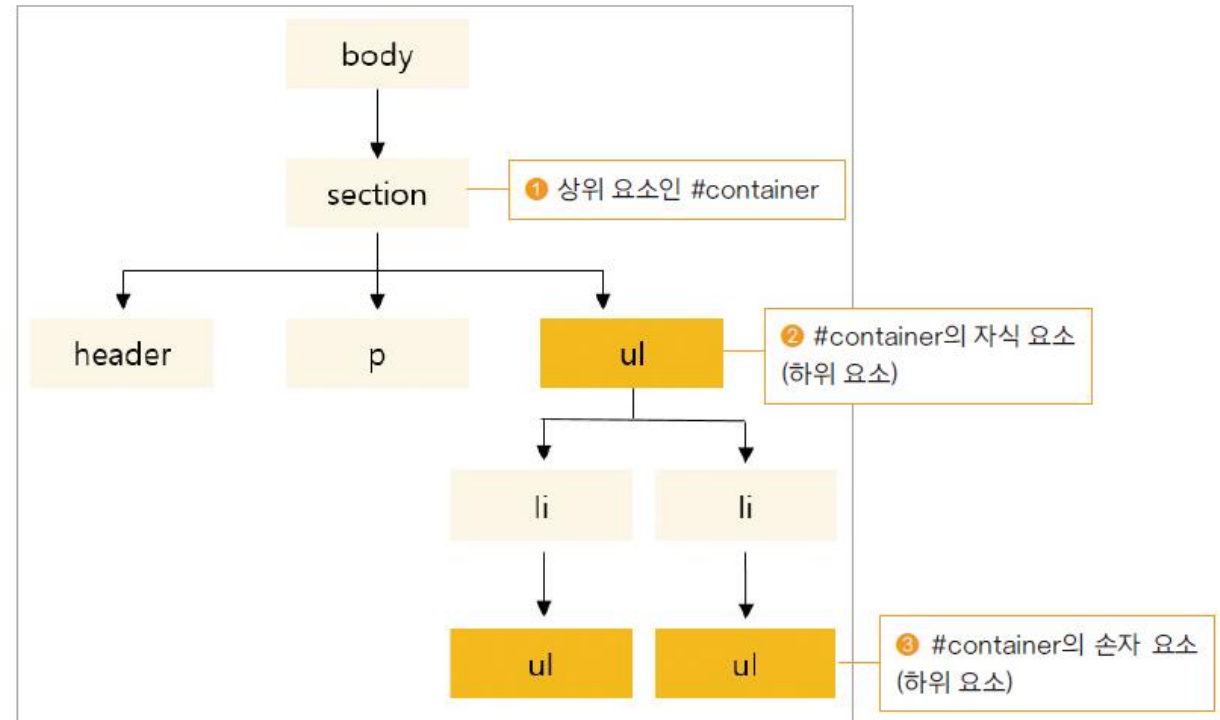
- 부모 요소에 포함된 모든 하위 요소에 스타일이 적용된다
- 자식 요소뿐만 아니라 손자 요소, 손자의 손자 요소 등 모든 하위 요소까지 적용
- 하위 선택자를 정의할 때는 상위 요소와 하위 요소를 나란히 쓴다.



연결 선택자

하위 선택자

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

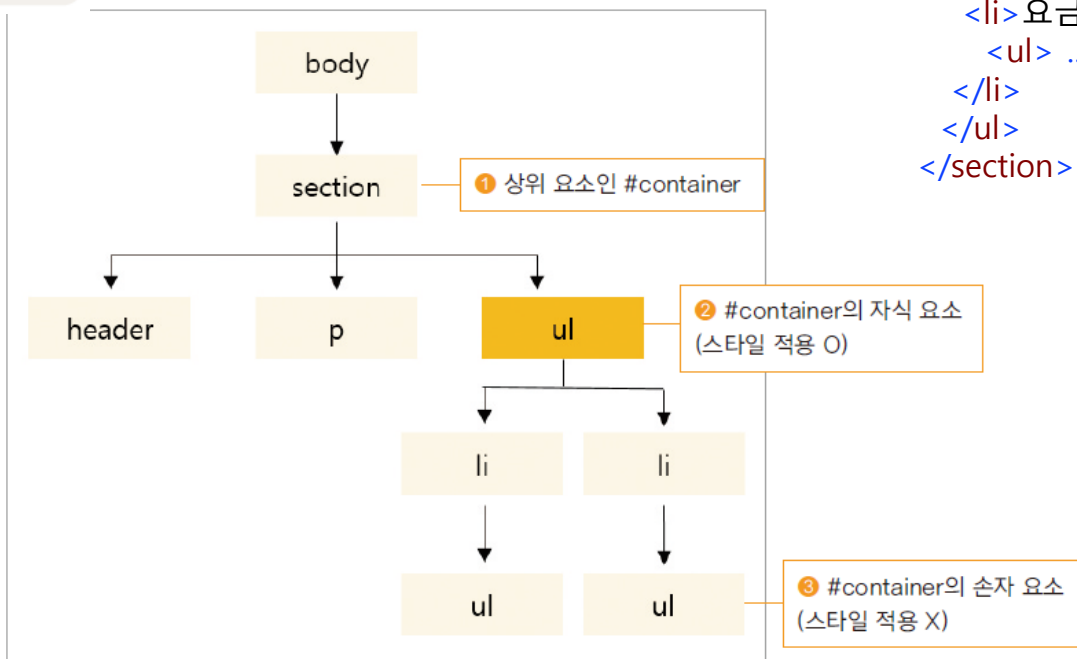


연결 선택자



자식 선택자

- 자식 요소에만 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>
```

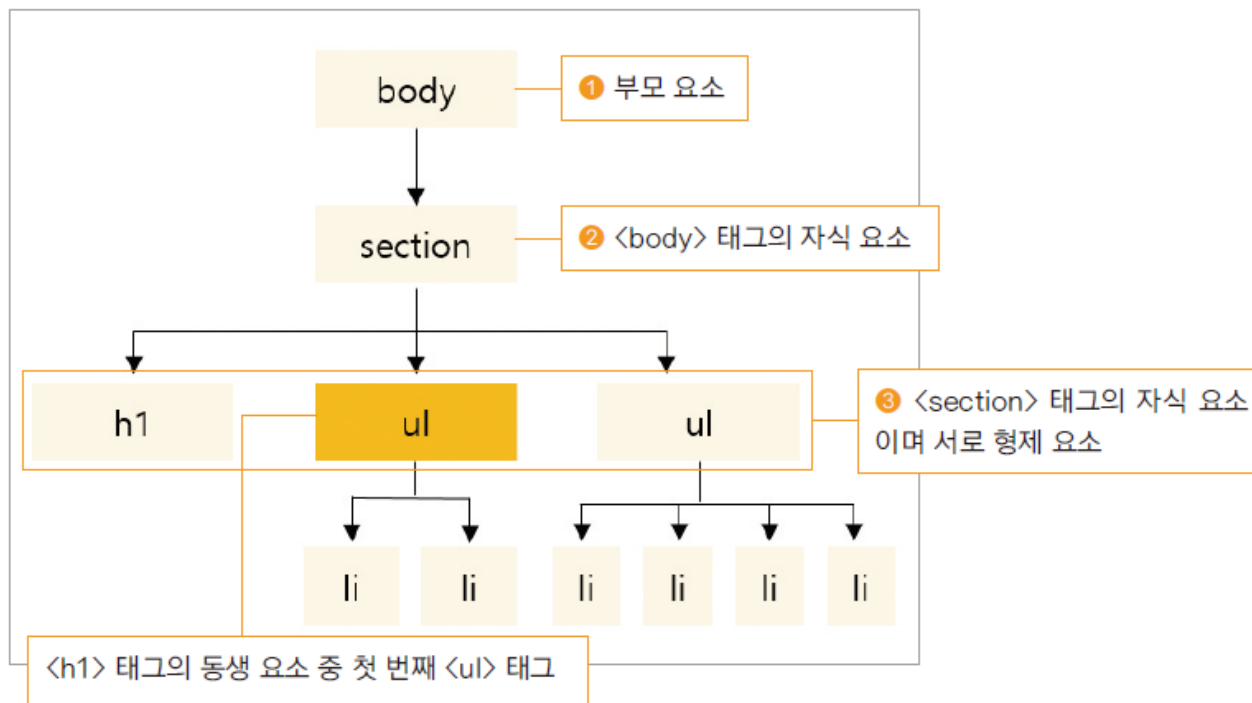
연결 선택자

인접 형제 선택자

- 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; }
```

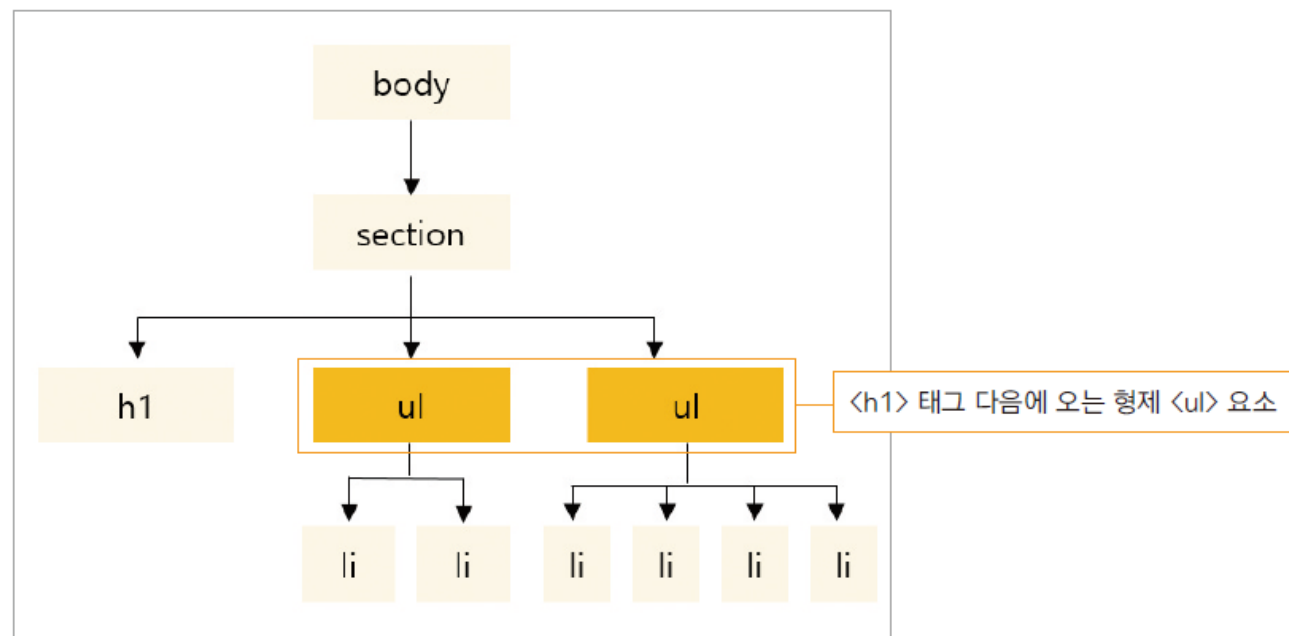
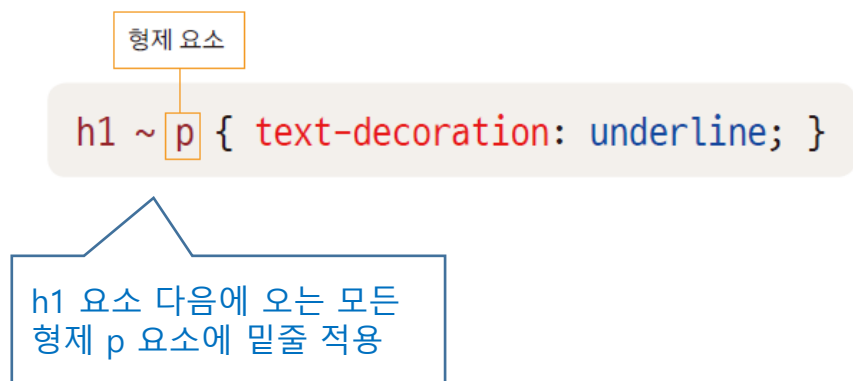
h1 요소 다음에 오는
p 요소들 중 첫번째
p 요소에만 밑줄적용



연결 선택자

형제 선택자

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



속성 선택자



[속성] 선택자

지정한 속성을 가진 요소를 찾아 스타일 적용

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

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

가상 클래스와 가상 요소



UI 요소 상태에 따른 가상 클래스

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

사용자 정보

이름

신청 과목

이 달에 신청할 과목을 선택하세요

☐ 회화 ☒ 문법 ☐ 작문

```
<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"> 회화`

`</label>`

`<label>`

`<input type="radio" name="subject" value="grammar"> 문법`

`</label>`

`<label>`

`<input type="radio" name="subject" value="writing"> 작문`

`</label>`

가상 클래스와 가상 요소



그 외 가상 클래스

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.