

# 10. CSS 고급 선택자

## 10-1 연결 선택자

## 10-2 속성 선택자

## 10-3 가상 클래스와 가상 요소



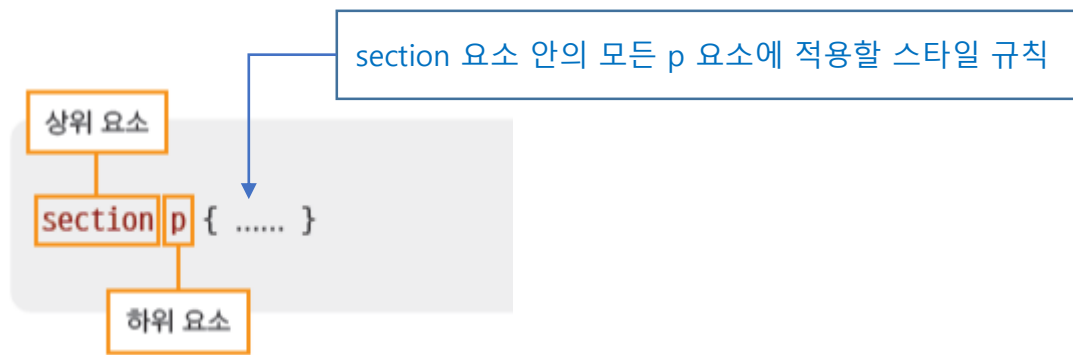
# 연결 선택자

- 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



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



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>
```

.....

```
<h1>예약 방법 & 사용 요금</h1>
```

```
<p>아직 온라인 예약 신청이.....</p>
```

section 요소의 자식인  
p 요소에만 스타일 적용

```
<div>
```

```
<p>가족실(2~4인): 60,000원/일</p>
```

```
<p>도미토리(4인 공용): 25,000원/일</p>
```

```
</div>
```

```
</section>
```

(... 생략 ...)

### 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(0xx-xxxx-xxxx)로 문의하시기 바랍니다.

가족실(2~4인) : 60,000원/일

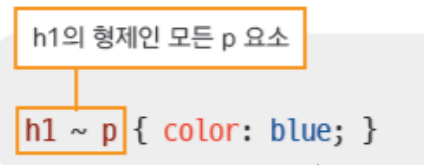
도미토리(4인 공용) : 25,000원/일

# 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



Do it! 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>

h1 요소의 형제인 p 요소에 모두 적용

(... 생략 ...)

### 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

실습>> child\_example.html → css/child.css 만들어 사용

1. 문서전체 : 배경색은 #eee로 설정
2. Section : 가로길이 600px,  
margin 위,아래 20px, 좌, 우 auto
3. p: 가로길이 500px, 패딩 : 모두 10px, 배경색 : #fff  
테두리 : 1px, 실선, #ccc, 줄높이 : 2


### 예약 방법 & 사용 요금

아직 온라인 예약 신청이 준비되어 있지 않습니다.  
전화(0xx-xxxx-xxxx)로 문의 바랍니다.


가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일


## 실습>> child\_example.html

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


- section 요소의 모든 p 요소에 적용

```
section > p {  
  color:  blue;  
}
```

- section 요소의 자식 p 요소에 적용

```
h1 + p {  
  color:  blue;  
}
```

- h1의 첫번째 형제 요소에 적용

```
h1 ~ p {  
  color:  red;  
}
```

- h1의 모든 형제 요소에 적용

# 태그 안의 속성값 선택자

## [속성] 선택자

지정한 속성을 가진 요소를 찾아 스타일 적용

ex) a[href] -> a태그의 href속성 찾을

## [속성 = 값] 선택자

주어진 속성과 속성 값이 일치하는 요소를 찾아 스타일 적용

ex) a[target = \_blank] -> a태그의 target속성이 \_blank인 속성 찾을

## [속성 ~ = 값] 선택자

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

ex) [class ~ = "button"] -> class태그중 button이 포함된 요소 찾을

## [속성 |= 값] 선택자

특정 값이 포함된 속성을 가진 요소를 찾아 스타일 적용

하이픈으로 연결해 한 단어 값을 이루는 요소에도 적용

ex) a[title |= "us"] -> us 또는 us- 로 시작하는 요소 찾을

## [속성 ^ = 값] 선택자

정확하지 않더라도 특정 값으로 시작하는 속성을 가진 요소를 찾아 스타일 적용

ex) a[ title ^ = "eng" ] -> eng로 시작하면 찾을

## [속성 \$ = 값] 선택자

특정 값으로 끝나는 속성을 가진 요소를 찾아 스타일 적용

ex) a[ title \$ = "hwp" ] -> hwp로 끝나는 요소 찾을

## [속성 \* = 값] 선택자

값의 일부가 일치하는 속성을 가진 요소를 찾아 스타일 적용

ex) a[ title \* = "w3" ] -> w3가 포함된 요소 찾을



## 실습>> attr-1.html → css/attr-1.css 만들어 사용

1. ul : 순서없는 목록 스타일 없애기
2. li : 가로 120px, 디스플레이 inline block, margin 10px
3. li 의 모든 a 태그 : 안쪽여백 위,아래 5px, 좌우 20px / 글자크기 14px / 밑줄 없애기 / 글자굵게

- 메인 메뉴 :
- 메뉴 1
- 메뉴 2
- 메뉴 3
- 메뉴 4

실습1 >> a 태그의 모든 href 속성

```
a[href] {  
  background: yellow;  
  border: 1px solid #ccc;  
  font-weight: normal;  
}
```

실습2 >> a 태그의 target이 \_blank인 속성에 적용

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

실습3 >> class가 flat 와 a태그중 class가 button을 포함하고 있는 속성에 적용

```
.flat {  
  background: #eee;  
  border: 1px solid #222;  
}  
a[class ~="button"] {  
  box-shadow: rgba(0,0,0,0.5) 4px 4px; /* 그림자 지정 */  
  border-radius: 5px; /* 테두리를 둥글게 */  
  border: 1px solid #222;  
}
```

## 실습4 >> attr-4.html

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

## 실습5 >> attr-5.html

```
a[title ^= "eng"] { /* 속성값이 "eng"로 시작하는 요소를 찾는 선택자 */
  background: url(images/us.png) no-repeat left center;
  padding: 5px 25px;
}
a[title $= "se"] { /* 속성값이 "se"로 끝나는 요소를 찾는 선택자 */
  background: url(images/jp.png) no-repeat left center;
  padding: 5px 25px;
}
a[title *= "hin"] { /* 속성값이 "hin"을 포함하는 요소를 찾는 선택자 */
  background: url(images/ch.png) no-repeat left center;
  padding: 5px 25px;
}
```

# register1.html

- register1.css를 만들어 이해하기

# 가상 클래스와 가상 요소

## 사용자 동작에 반응하는 가상 클래스 선택자

**:link** - 방문하지 않은 링크에 스타일 적용

**:visited** - 방문한 링크에 스타일 적용

**:active** - 웹 요소를 활성화(클릭)했을 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

실습 >> navi.html

```
.navi a:link, .navi a:visited{
  display:block;
  font-size:14px;
  color:■#000;
  padding: 10px;
  text-decoration: none; /* 밑줄 없앴 */
  text-align: center;
}

.navi a:hover, .navi a:focus {
  background-color:■#222; /* 배경 색 */
  color:■#fff; /* 글자 색 */
}

.navi a:active {
  background-color:■#f00; /* 배경 색 */
}
```

# 가상 클래스와 가상 요소

## 요소 상태에 따른 가상 클래스 선택자

- 다른 사이트의 페이지로 이동할때 링크(link) 이용
  - 같은 문서안 에서 다른 위치로 이동할 때는 앵커(anchor)
- :target** – 앵커로 연결된 부분에 스타일 적용

실습 >> ps-1.html

```
42  .contents {
43      margin:30px auto;
44      width:400px;
45      padding:20px;
46      border:1px solid #222;
47      border-radius:5px;
48  }
49  #intro:target {
50      background-color:#0069e0; /* 배경색 - 파란색 */
51      color:#fff; /* 글자색 - 흰색 */
52  }
53  #room:target {
54      background-color:#ff9844; /* 배경색 - 주황색 */
55  }
56  #reservation:target{
57      background-color:#fff8dc; /* 배경색 - 베이지색 */
58  }
```

# 가상 클래스와 가상 요소

## 요소 상태에 따른 가상 클래스 선택자

클래스 이름앞에 콜론1개(:) 붙인다

: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



## 실습>> register1.html, register1.css 추가하기

```
40  #signup button {
41      border: 1px solid ☐ #222;
42      border-radius: 20px;
43      display: block;
44      font-size: 16px;
45      letter-spacing: 1px;
46      margin: auto;
47      padding: 7px 25px;
48  }
49
50  #signup label {
51      float: left;
52      font-size: 13px;
53      width: 110px;
54  }
55
```

## 실습>> register1.html, register1.css 추가하기

```
#signup input:not([type=radio]) { /* 가상 선택자를 사용해 간단히 줄일 수 있음 */  
  border: 1px solid  #ccc; /* 텍스트 필드 테두리 지정 */  
  border-radius: 3px; /* 텍스트 필드 모서리를 둥글게 지정 */  
  font-size: 13px; /* 텍스트 필드 안의 글자 크기 지정 */  
  padding: 5px; /* 텍스트 필드와 내용 사이의 패딩 지정 */  
  width: 200px; /* 텍스트 필드 너비 지정 */  
}  
  
#signup input[required] { /* 필수 입력 필드의 스타일 */  
  border: 1px  red solid;  
}  
  
#signup input[readonly] { /* 읽기 전용 필드의 스타일 */  
  border: none;  
}
```

# 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.

# 가상 클래스와 가상 요소



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>

contents의 모든 자식 요소 중에서 3번째이므로 ① 스타일이 지정됨

contents의 p 자식 요소 중에서 3번째이므로 ② 스타일이 지정됨

## 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.

# 실습 >> ps-3.html 이용하기

```
.contents :nth-child(3) {  
  background-color: ■ #ffff00;  
}  
  
.contents p:nth-of-type(3) {  
  background-color: ■ #00b900;  
}
```

## 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.

## 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.

```
.contents :nth-child(odd) {  
  background-color: ■ #ffff00;  
}  
  
.contents :nth-child(even) {  
  background-color: ■ #00b900;  
}
```

```
.contents :nth-child(2n) {  
  background-color: ■ #ffff00;  
}  
  
.contents :nth-child(2n+1) {  
  background-color: ■ #00b900;  
}
```

# 가상 클래스와 가상 요소

## 가상 요소

- 화면 꾸미기용 요소를 웹 문서에 포함시키지 않기 위해 가상 요소 사용
- 가상요소이름 앞에 콜론2개(::<) 붙임
- ::first-line : 특정 요소의 **첫번째 줄**에 스타일 적용
- ::first-letter : 특정 요소의 **첫번째 글자**에 스타일 적용
- ::before : 특정 요소의 **앞**에 지정한 콘텐츠 추가
- ::after : 특정 요소의 **뒤**에 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!!**

## 실습 >> register1.html, register1.css 추가하기

```
#signup fieldset:last-of-type { /* 마지막 fieldset의 스타일 */  
  border:none; /* 테두리 없앰 */  
  margin-bottom:0; /* 아래쪽 마진 없앰 */  
}  
  
#signup button:hover { /* #singup 하위에 있는 버튼에 마우스 올렸을 때 스타일 */  
  background-color:□#222; /* 배경색 - 검정 계열색 */  
  color:■#fff; /* 글자색 - 흰색 */  
}
```

로그인 정보

아이디

비밀번호

비밀번호 확인

회원 등급 ☐ 준회원

개인 정보

이름

메일 주소

연락처



로그인 정보

아이디

비밀번호

비밀번호 확인

회원 등급 ☐ 준회원

개인 정보

이름

메일 주소

연락처