

# 9강. CSS의 다양한 선택자



# 목 차

1

연결 선택자

2

속성 선택자

3

가상 클래스 선택자



# 연결 선택자

## 연결 선택자와 속성 선택자를 사용하는 이유

사용 이유 : id, class 선택자가 늘어나면 이름을 기억하기 쉽지 않고 복잡해짐에 따라 연결 또는 속성 선택자를 사용하여 소스의 위치나 속성값에 따라 특정 요소를 쉽게 선택할 수 있다.

### ▶ 연결 선택자

- 선택자와 선택자를 연결해 적용 대상을 한정하는 선택자, 조합선택자라고도 함.

#### 1. 하위 선택자

- 부모 요소에 포함된 모든 하위 요소(자손 선택자)에 스타일이 적용된다.
- **기본형** : 상위요소 하위요소 (상위 요소와 하위 요소를 공백을 띄고 나란히 쓴다.)

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



# 연결 선택자

## 2. 자식 선택자

- 부모 요소에 포함된 *자식 요소*에 스타일이 적용된다.
- **기본형** : 상위요소 > 하위요소 (상위 요소와 하위 요소를 부등호를 사용하여 나란히 쓴다.)

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



# 연결 선택자

## 예약 방법 & 사용 요금

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

가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일

후손 선택자 사용

자식 선택자 사용

## 예약 방법 & 사용 요금

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

가족실(2~4인) : 60,000원/일

도미토리(4인 공용) : 25,000원/일



# 연결 선택자

child.html

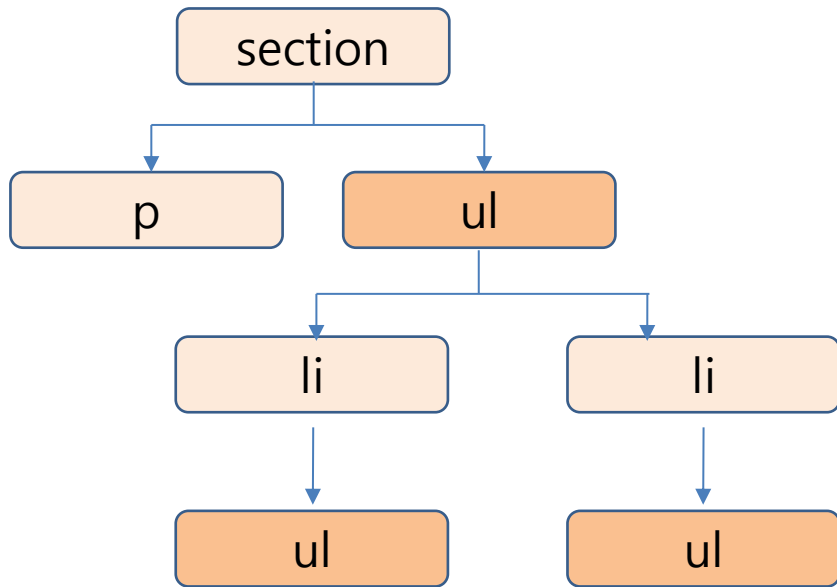
```
<section>
  <h1>예약 방법 & 사용 요금</h1>
  <p>아직 온라인 예약 신청이 준비되어 있지 않습니다. <br>
  | 전화(031-1234-0000)로 문의 바랍니다.</p>
  <div>
    <p>크리스탈 : 60,000원/일</p>
    <p>루비 : 75,000원/일</p>
  </div>
</section>
```

```
body {background-color: #eee;}
section {width:600px; margin:20px auto;}
p {
  width:500px;
  border:1px solid #ccc;
  padding:10px;
  background-color: #fff;
  line-height:2;
}
section > p { /* section 요소의 자식 p 요소에 적용 */
  color: blue; /* 글자색을 파란색으로 */
}
```

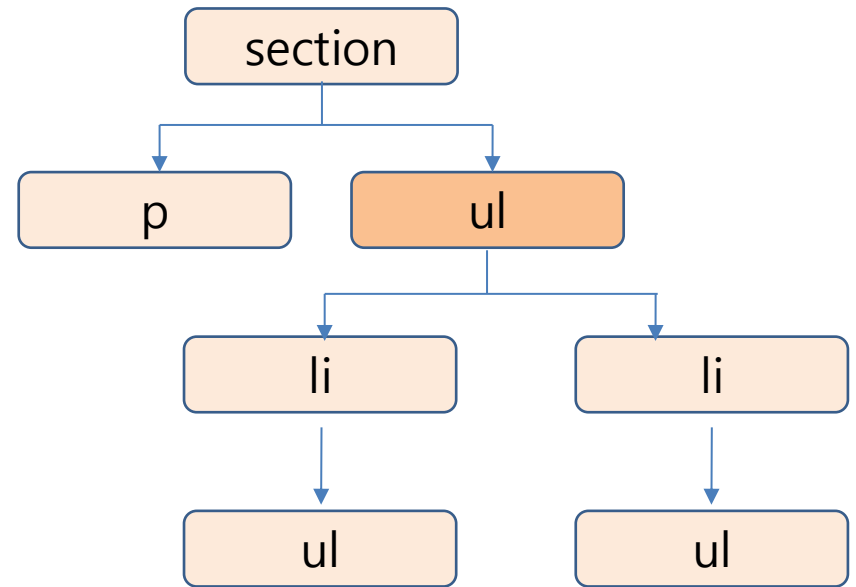


# 연결 선택자

연결 선택자 비교도.



하위 선택자(후손)



자식 선택자

# 연결 선택자

descendant.html

```
<section id="container">
  <header>
    <h1>예약 방법 및 요금</h1>
  </header>
  <p>행복팬션에 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

descendant.html

## 예약 방법 및 요금

행복 펜션에 예약하려면?

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

```
#container ul{  
    border: 1px dotted blue;  
    /* 하위 요소 전체에 스타일 적용 */  
}
```

child.html

## 예약 방법 및 요금

행복펜션에 예약하려면?

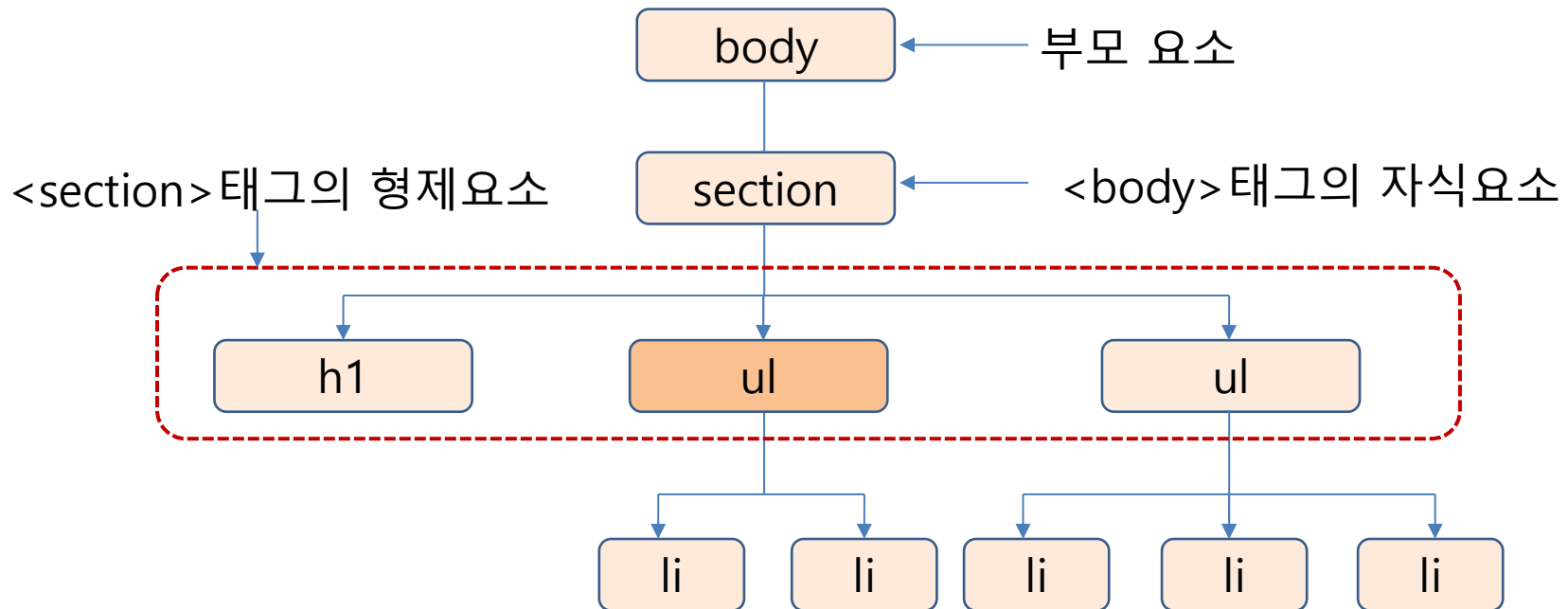
- 예약 방법
  - 직접 통화
  - 문자 남기기
- 요금
  - 1인 : 40,000원
  - 2인 : 60,000원
  - 3인 : 80,000원
  - 4인 : 100,000원

```
#container > ul{  
    border: 1px dotted blue;  
    /* 자식요소에만 style 적용 */  
}
```



# 연결 선택자

## 연결 선택자 계층도



# 연결 선택자

naver2.html

## 예약 방법 및 요금

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

```
<body>
  <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>
      <li>2인 : 60,000원</li>
      <li>3인 : 80,000원</li>
      <li>4인 : 100,000원</li>
    </ul>
  </section>
</body>
```



# 속성 선택자

## [속성] 선택자

- 주어진 속성과 속성 값이 일치하는 요소를 찾아 스타일 적용
- **a[target]**

### 주요 포털 사이트

구글(google) 

네이버(naver)

다음(daum)

# 속성 선택자

```
<h2>주요 포털 사이트</h2>
```

```
<ul>
```

```
  <li>
```

```
    <a href="http://www.google.com" target="_blank">구글(google)</a>
```

```
  </li>
```

```
  <li>
```

```
    <a href="http://www.naver.com">네이버(naver)</a>
```

```
  </li>
```

```
  <li>
```

```
    <a href="http://www.daum.net">다음(daum)</a>
```

```
  </li>
```

```
</ul>
```

attr\_selector.html

```
ul li{
```

```
  list-style: none;
```

```
  margin: 15px;
```

```
}
```

```
ul li a{
```

```
  text-decoration: none;
```

```
}
```

```
ul li a[target]{
```

```
  background: url("images/newwindow.png") no-repeat center right;
```

```
  padding-right: 30px;
```

```
}
```



# 속성 선택자

## [속성=속성값] 선택자 (selector)

지정한 속성과 속성값을 가진 요소를 찾아 스타일 적용

### 회원 가입

—개인 정보—

이름	<input type="text"/>
메일 주소	<input type="text"/>
연락처	<input type="text"/>
블로그/홈페이지	<input type="text"/>

```
li label {  
    width:120px;  
    float:left;  
    text-align:right;  
    margin-right:8px;  
}  
input[type="submit"],[type="reset"] {  
    height:30px;  
}
```

# 가상 선택자

**가상 (클래스) 선택자**란 지정하기 어려운 웹 요소를 선택할 수 있는 기능을 하는데 클래스 이름앞에 콜론(:)을 붙인다.

## 사용자 동작에 반응하는 가상 선택자

- 웹 요소를 클릭하거나 마우스 커서를 올려놓는 등 특정 동작을 할때 사용함
- **콜론(:)**을 클래스 선택자 이름 앞에 붙인다.

표기	설명
:link	방문하지 않은 링크에 스타일 적용
:visited	방문한 링크에 스타일 적용
:hover	웹 요소에 마우스 커서를 올려 놓았을때 스타일 적용
:active	웹 요소를 활성화했을 때의 스타일 적용( <b>마우스 클릭</b> )
:focus	웹 요소에 초점이 맞추어졌을 때의 스타일 적용( <b>탭키 누름</b> )

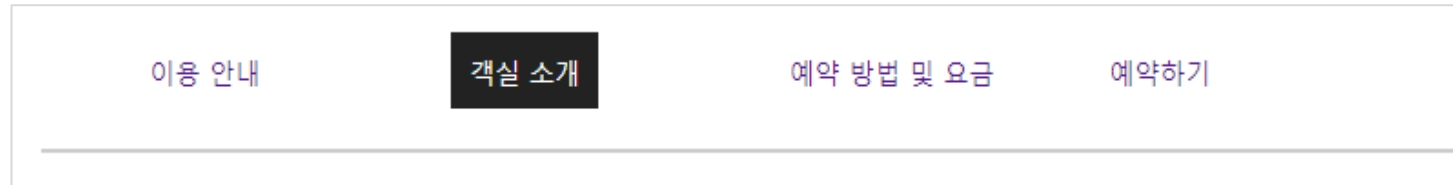
a{ }와  
같음

※ 2개 이상의 선택자를 사용할 경우 위에서 아래로 순서대로 해야 적용된다.



# 가상 클래스 선택자

## 링크에서 가상 선택자 사용하기



```
<div class="container">  
  <nav>  
    <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>  
</div>  
</body>
```





# 가상 클래스 선택자

nav.css

```
*{margin: 0; padding: 0;}
.container {width: 960px; margin : 0 auto;}
nav{height: 80px; text-align: center;}
nav ul{list-style: none; border-bottom:2px solid #ccc; padding: 15px; }
nav ul li {display: inline-block; margin: 10px 20px;}
nav a:link, .navi a:visited{
    text-decoration: none;
    color: #000;
    padding: 10px;
    font-size:14px;
}
nav a:hover, a:focus {
    background-color: #222;
    color: #fff;
}
nav a:active {background-color: #f00;}
```



# 가상 클래스 선택자

## UI 요소 상태에 따른 가상 선택자

사용자의 동작 뿐만 아니라 **웹 요소의 상태**에 따라 스타일을 지정할 때도 가상 선택자 사용

표기	설명
:enabled 와 :disabled	요소를 사용할 수 있을 때와 없을 때의 스타일 지정
:checked	라디오박스나 체크 박스에서 해당 항목을 선택했을때

사용자 정보

이름

신청 과목

이 달에 신청할 과목을 선택하세요

☐ 회화 ☐ 문법 ☒ 작문

# 가상 클래스 선택자

```
<form>
  <fieldset>
    <legend>사용자 정보</legend>
    <label>이름 <input type="text" disabled></label>
  </fieldset>
  <fieldset>
    <legend>신청 과목</legend>
    <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>
  </fieldset>
</form>
```

ui-element.html

```
form fieldset{margin-bottom: 20px;}
input:disabled{ /*가상 선택자(:disabled)로 스타일 적용*/
  background: #ddd;
  border: 1px solid #ccc;
}
input:checked + span{
  /*가상 선택자(:checked)로 스타일 적용
  | '+'는 연결 연산자임.
  */
  color: blue;
}
```



# 가상 클래스 선택자

## 구조 가상 선택자

웹 문서 구조를 기준으로 특정 위치에 있는 요소를 찾아 스타일을 지정할 때 사용

종류	설명
:first-child	부모 안에 있는 모든 요소 중에서 첫 번째 자식 요소를 선택
:last-child	부모 안에 있는 모든 요소 중에서 마지막 자식 요소를 선택
:nth-child(n)	부모 안에 있는 모든 요소 중에서 n번째 자식요소를 선택
:nth-last-child(n)	부모 안에 있는 모든 요소 중에서 끝에서 n번째 자식요소를 선택
A:nth-of-type(n)	부모 안에 있는 A요소 중에서 n번째 요소를 선택
A:nth-last-of-type(n)	부모 안에 있는 A요소 중에서 끝에서 n번째 요소를 선택



# 가상 클래스 선택자

## 본문 내용1

구조 가상 클래스란 웹 문서의 구조를 기준으로 특정 위치에 있는 요소를 찾아 스타일을 적용할 때 사용하는 가상 클래스 선택자이다.

요소가 여러개 나열되어 있는 경우에는 class나 id를 사용하지 않고도 스타일을 지정할 요소가 몇 번째인지를 따져서 스타일을 적용할 수 있다.

## 본문 내용2

구조 가상 클래스란 웹 문서의 구조를 기준으로 특정 위치에 있는 요소를 찾아 스타일을 적용할 때 사용하는 가상 클래스 선택자이다.

## 본문 내용3

요소가 여러개 나열되어 있는 경우에는 class나 id를 사용하지 않고도 스타일을 지정할 요소가 몇 번째인지를 따져서 스타일을 적용할 수 있다.



# 가상 클래스 선택자

nth.html

```
<div class="container">
  <div class="contents">
    <h2>본문 내용1</h2>
    <p>구조 가상 클래스란 웹 문서의 구조를 기준으로 특정 위치에 있는 요소를 찾아
    | 스타일을 적용할 때 사용하는 가상 클래스 선택자이다.</p>
    <p>요소가 여러개 나열되어 있는 경우에는 class나 id를 사용하지 않고도 스타일을
    | 지정할 요소가 몇 번째인지를 따져서 스타일을 적용할 수 있다.</p>
    <h2>본문 내용2</h2>
    <p>구조 가상 클래스란 웹 문서의 구조를 기준으로 특정 위치에 있는 요소를 찾아
    | 스타일을 적용할 때 사용하는 가상 클래스 선택자이다.</p>
    <h2>본문 내용3</h2>
    <p>요소가 여러개 나열되어 있는 경우에는 class나 id를 사용하지 않고도 스타일을
    | 지정할 요소가 몇 번째인지를 따져서 스타일을 적용할 수 있다.</p>
  </div>
</div>
```



# 가상 클래스 선택자

nth.css

```
.container {width: 960px; margin: 0 auto;}
.contents {
  width:400px;
  margin: 0 auto;
  border:1px solid ■ #222;
  border-radius:5px;
  padding:20px;
}
.contents :first-child{
  /* 모든 자식 요소중 첫번째 자식 요소
  | | 선택자 앞에 1칸 띄어줌 */
  color: ■ red;
}
.contents :nth-child(2) { /* 자식 요소중 3번째 요소*/
  background-color: ■ #ffff00;
}
.contents p:nth-of-type(3) { /*p 태그 중 3번째 위치*/
  background-color: ■ #00ff00;
}
.contents :last-child{ /* 모든 자식 요소중 마지막 자식 요소 */
  color: ■ blue;
}
```



# 가상 클래스 선택자

**:nth-child(n)** – 앞에서부터 n번째 자식 요소에 스타일 적용  
위치를 나타낼 때는  $an+b$  처럼 수식을 사용할 수 있음

건강에 좋은 건강 식품	
블루베리	
귀리	
토마토	
시금치	
적포도주	
견과류	
브로콜리	
연어	
마늘	
녹차	



# 가상 클래스 선택자

nth-child.html

```
<div id="container">
  <h1>건강에 좋은 건강 식품</h1>
  <table>
    <tr>
      <td>블루베리</td>
    </tr>
    <tr>
      <td>귀리</td>
    </tr>
    <tr>
      <td>토마토</td>
    </tr>
    <tr>
      <td>시금치</td>
    </tr>
    <tr>
      <td>적포도주</td>
    </tr>
    <tr>
      <td>견과류</td>
    </tr>
```

```
h1{text-align: center;}
table{
  width: 200px;
  margin: 0 auto;
  border-collapse: collapse;
}
td{
  border: 1px solid #ccc;
  text-align: left;
  padding: 10px 10px 10px 20px;
}
tr:nth-child(2n+1){background: #eee;}
/* tr:first-child{background: #eee;} */
/* tr:last-child{background: #eee;} */
```



## 가상 선택자 실습 문제

아래와 같은 조건을 이용하여 표를 작성하고 table-nth.html로 저장하세요

이 름		연락처	
주 소			
자기소개			

- ① 열 합치기는 <table>의 colspan 속성을 사용한다.
- ② 가상 선택자 nth-child()를 사용하여 열의 너비와 배경색을 지정한다.



# 가상 선택자 실습 문제

table-nth.html

```
<table>
  <tr>
    <td>이 름</td>
    <td></td>
    <td>연락처</td>
    <td></td>
  </tr>
  <tr>
    <td>주 소</td>
    <td colspan="3"></td>
  </tr>
  <tr>
    <td>자기소개</td>
    <td colspan="3"></td>
  </tr>
</table>
```

```
table, td{
  border: 1px solid #ccc;
  border-collapse: collapse;
  padding: 10px;
  text-align: center;
}
td:nth-child(2n+1){
  width: 100px;
  font-weight: bold;
  background: #eee;
}
td:nth-child(2n){
  width: 200px;
}
```



# 가상 선택자 실습 문제

first-of-type.html



```
  
  

```

```
img {  
  margin-right: 50px;  
  border: 1px solid #ccc;  
  box-shadow: 2px 2px 5px #ccc;  
}  
img:first-of-type {  
  border: 2px solid #f00;  
}
```

# 가상 요소

## 가상 요소

가상요소를 만들어 사용하는 이유는 특별히 화면에 보이는 부분을 꾸밀 때 태그의 사용을 줄일 수 있다.

### 내용 앞 뒤에 콘텐츠를 추가하는 **::before** 요소 **::after** 요소

지정한 요소의 앞 뒤에 텍스트나 이미지 등을 추가할 수 있다.

가상 클래스와 구분하기 위해 콜론 2개 (::)를 붙여서 표시한다.

#### 상품 목록

- 라면 **NEW**
- 생수
- 두부
- 커피 **NEW**



# 가상 요소

after.html

```
<div class="container">
  <h2>상품 목록</h2>
  <ul>
    <li class="new">라면</li>
    <li>생수</li>
    <li>두부</li>
    <li class="new">커피</li>
  </ul>
</div>
```

```
.goods {width: 600px; margin : 0 auto;}
ul li {margin:15px;}

/* 가상요소 ::after */
/* content: 'NEW' 사용 */
ul .new::after{
  content: "NEW";
  background-color: ■ #f00;
  border-radius: 2px;
  font-size: x-small;
  color: □ #fff;
  margin: 0 6px;
}
```

# 가상 요소

[모델](#) > [승용](#) > SONATA

## SONATA

다시 한번 성장하다.

SONATA 2021년형 출시

```
<div id="container">
  <section>
    <nav>
      <ol>
        <li><a href="#">모델</a></li>
        <li><a href="#">승용</a></li>
        <li>SONATA</li>
      </ol>
    </nav>
    <h1>SONATA</h1>
    <p>다시 한번 성장하다.</p>
    <p>SONATA 2021년형 출시</p>
  </section>
</div>
```

## 가상 요소

```
#container{width: 700px; margin: 0 auto;}
nav {border-bottom: 1px solid black;}
nav ol {
  list-style: none;
  padding-left: 0;
}
nav ol li {display: inline-block;}
nav a{text-decoration: none;}
nav a::after {
  content: '>';
  color: #000;
  font-size: 80%;
  font-weight: bold;
  padding: 0 5px;
}
```



# 서브메뉴(submenu)

기업정보

제품소개

투자정보

인재채용

메모리1

메모리2

메모리3

메모리4



# 서브메뉴(submenu)

```
<div id="container">
  <nav>
    <ul id="main-menu">
      <li><a href="#">기업정보</a></li>
      <li><a href="#">제품소개</a>
        <ul id="sub-menu">
          <li><a href="#">메모리1</a></li>
          <li><a href="#">메모리2</a></li>
          <li><a href="#">메모리3</a></li>
          <li><a href="#">메모리4</a></li>
        </ul>
      </li>
      <li><a href="#">투자정보</a></li>
      <li><a href="#">인재채용</a></li>
    </ul>
  </nav>
</div>
```

sebmenu.html



# 서브메뉴(submenu)

style.css

```
*{margin:0; padding:0;}
#container{width: 1000px; margin:0 auto;}
nav{height: 60px; background: ■rgb(1, 1, 168);}
#main-menu, #sub-menu{list-style: none; text-align: center; }
#main-menu > li{display: inline-block; background: ■rgb(1, 1, 168); margin: 10px;}
#main-menu > li > a{
    display: block; padding: 10px 30px; text-decoration: none;
    background: ■rgb(1, 1, 168); color: □#fff; font-size: 0.9em;
}

/* sub-menu 스타일 */
#sub-menu{position: absolute; top: 60px; background: ■#012881; visibility: hidden;}
#sub-menu > li{padding: 15px 30px;}
#sub-menu > li > a{text-decoration: none; color: □rgba(255, 255, 255, 0.6);}
#main-menu > li:hover #sub-menu{visibility: visible;}
```

