

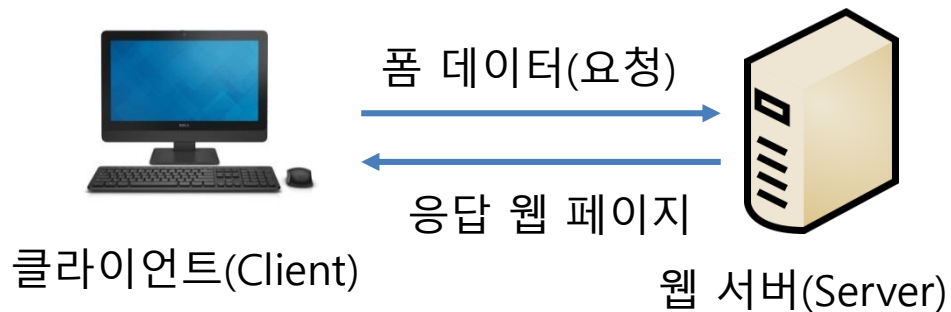
4강. 폼(Form) 및 고급 선택자



폼(Form) 만들기

웹에서 만나는 폼(form)

- 폼은 사용자가 웹 브라우저를 통해 입력한 데이터를 웹 서버로 전송하는 양식이다.
- 폼에 입력한 사용자 정보를 처리하는 것은 Java(JSP), Python, PHP와 같은 서버 측 웹 프로그래밍 언어이다.



Pybo

로그인 [또는 계정을 만드세요](#)

사용자ID

비밀번호

로그인

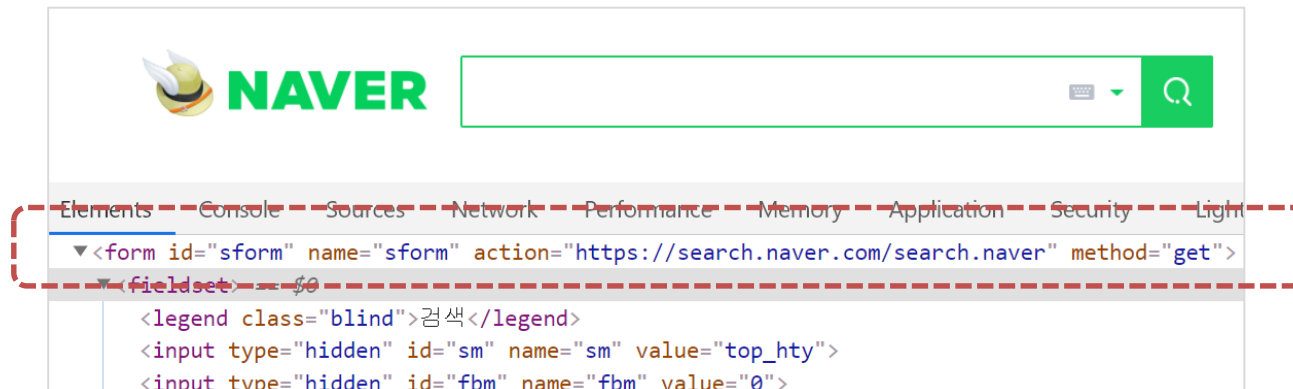


폼 만들기

● <form> 태그

- <form 속성="속성 값">여러 폼 요소 </form>
- 몇 가지 속성을 통해 사용자가 입력한 자료들을 서버 쪽으로 넘기는데 이용

속성	설명
action	<form> 태그 안의 내용들을 처리해 줄 서버 상의 경로를 지정한다.
method	get – 주소 표시줄에 사용자가 입력한 내용이 그대로 드러난다.
	post – 사용자가 입력한 내용이 드러나지 않는다.(보안이 필요한 경우)
name	폼의 이름을 지정합니다.



사용자 입력을 위한 <input> 태그

● <input> 태그

type 속성 값에 따라 함께 사용할 수 있는 속성들도 달라진다.

유형	설명
text	한 줄짜리 텍스트를 입력할 수 있는 텍스트 상자를 넣는다.
password	비밀번호를 입력할 수 있는 필드를 넣는다.
submit	사용자 입력 내용을 서버로 전송
reset	사용자 입력 내용 전부 삭제
number	숫자 입력 필드(value, min, max, step 속성)
date	날짜를 선택할 수 있도록 표시
radio	여러 항목 중 하나만 선택 가능
checkbox	여러 항목 중 다중 선택 가능함
file	파일을 첨부할때 사용



폼 만들기

● 검색 폼 만들기



A visual representation of the search form. It consists of a rectangular text input field on the left and a button with the Korean text '검색' (Search) on the right.

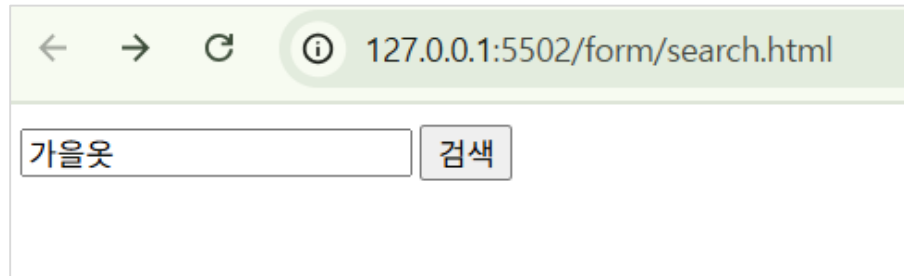
search.html

```
<!-- action = 이동경로(서버페이지), method=보내는 방법 -->  
<!-- method: get(주소표시줄에 노출), post(주소표시줄에 노출안됨) -->  
<!-- submit(폼을 서버에 전송), text: 입력상자 -->  
<form action="search.jsp" method="get">  
    <input type="text" name="sform">  
    <input type="submit" value="검색">  
</form>
```

<form>태그에서 action="search.jsp"로 지정했기 때문에 내용을 입력하고 검색버튼을 누르면 그 값이 웹 서버에 있는 search.jsp 파일로 전송되어 검색이 처리된다.

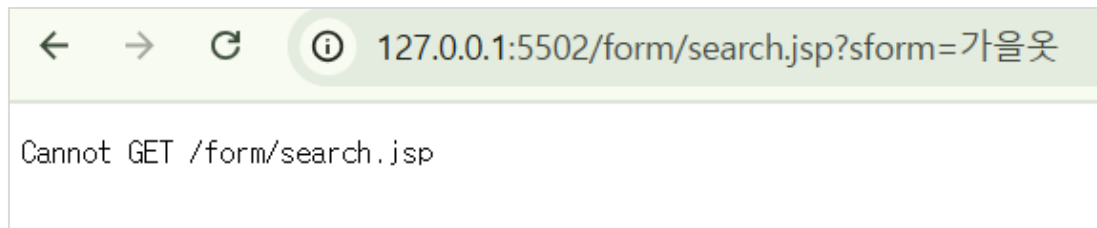
폼 만들기

● 검색 폼 만들기



← → ↻ ⓘ 127.0.0.1:5502/form/search.html

가을옷 검색



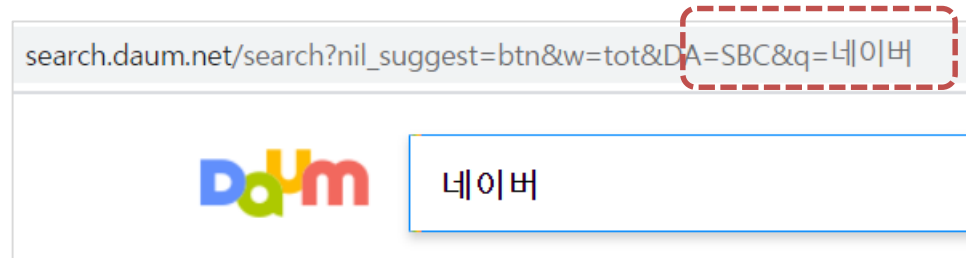
← → ↻ ⓘ 127.0.0.1:5502/form/search.jsp?sform=가을옷

Cannot GET /form/search.jsp

폼 만들기

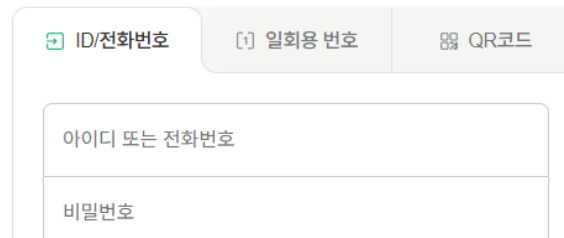
- method - get 과 post

get 방식 - 주소표시줄에 입력 내용이 표시됨



The screenshot shows a web browser's address bar with the URL `search.daum.net/search?nil_suggest=btn&w=tot&DA=SBC&q=네이버`. A red dashed box highlights the `DA=SBC&q=네이버` part of the URL. Below the address bar, the Daum logo is visible on the left, and a search input field on the right contains the text "네이버".

post 방식 - 주소표시줄에 입력한 내용이 표시되지 않음



The screenshot shows a login form with three tabs: "ID/전화번호" (selected), "일회용 번호", and "QR코드". Below the tabs, there are two input fields: "아이디 또는 전화번호" (ID or Phone Number) and "비밀번호" (Password).

아이디, 비밀번호를 입력하고 로그인을 요청하면 주소 표시줄에 나타나지 않음

```
<form id="frmNIDLogin" name="frmNIDLogin" target="_top" autocomplete="off" action="https://nid.naver.com/nidlogin.login" method="POST"> == $0
  <input type="hidden" id="localechange" name="localechange" value>
  <input type="hidden" name="dynamicKey" id="dynamicKey" value="X0-vDYZgA4-W29m0J150H_aPol__mQdteZgtVWo4ofujC5tF1TMwKhM7-MuILtKhN9f8PRFrCuCZD48124B4h9my47o4tkH8R9Ar5LxUns">
  <input type="hidden" name="eccpw" id="eccpw" value>
```



〈label〉 태그

● <label> 태그

- 입력 창 옆에 '아이디' 나 '비밀번호' 처럼 붙여 놓은 텍스트
- 스마트폰 처럼 화면이 작은 기기에서 **라디오 버튼**이나 **체크박스** 부분을 정확히 터치하는 것이 어려울 때 사용하면 더욱 유용하다.

로그인

아이디

비밀번호

로그인

〈label〉 태그

● <label> 태그

```
<h2>로그인</h2>
<form action="login.jsp" method="post">
  <ul>
    <li>
      <label for="userid">아이디</label>
      <input type="text" name="userid" id="userid">
    </li>
    <li>
      <label for="userpw">비밀번호</label>
      <input type="password" name="userpw" id="userpw">
    </li>
    <li>
      <button type="submit">로그인</button>
    </li>
  </ul>
</form>
```



〈label〉 태그

● <label> 태그

```
h2{
  margin: 20px;
  color: #369;
}
form{
  width: 350px;
  border: 1px solid #ccc;
  padding: 10px;
  margin: 20px;
}
form ul{
  padding: 0;
  margin: 0;
}
ul li{
  list-style: none;
  margin: 20px;
}
```

```
label{
  display: inline-block;
  width: 80px;
  text-align: right;
  margin-right: 10px;
}
button{
  margin-left: 100px;
  padding: 5px 15px;
  background: #ccc;
  border: none;
  cursor: pointer;
}
```

사용자 입력을 위한 <input> 태그

● radio버튼 & checkbox 버튼

- radio 버튼 – 1개의 항목만 선택 가능, name 속성의 값이 같아야 함
- checkbox – 여러 개의 항목 선택 가능

신청 과목

이 달에 신청할 과목을 선택하세요(1과목만 가능함)

☐ 회화 ☒ 문법 ☐ 작문

메일링

메일로 받고 싶은 뉴스 주제를 선택해 주세요(복수 선택 가능함)

☐ 해외단신 ☒ 5분 회화 ☒ 모닝팝스

사용자 입력을 위한 <input> 태그

● <fieldset>, <legend> 태그 – 폼 요소 그룹으로 묶기

<fieldset>태그는 폼의 요소들을 하나의 영역으로 묶어 외곽선을 그려주고,
<legend>태그는 <fieldset>태그로 묶은 그룹에 제목을 붙여 준다.

```
<form action="" method="post">
  <!-- <fieldset>은 폼의 요소를 하나의 영역으로 묶어서 외곽선을 그려줌
       <legend>은 묶은 그룹에 제목을 붙여준다.
       <label>로 감싸면 글자를 선택해도 적용됨 -->
  <fieldset>
    <legend>신청 과목</legend>
    <p>이 달에 신청할 과목을 선택하세요(1과목만 가능함)</p>
    <!-- name의 subject는 동일하게 작성 -->
    <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>
  </fieldset>
  <fieldset>
    <legend>메일링</legend>
    <p>메일로 받고 싶은 뉴스 주제를 선택해 주세요(복수 선택 가능함)</p>
    <label><input type="checkbox" name="mailing" value="news">해외단신</label>
    <label><input type="checkbox" name="mailing" value="dialog">5분 회화</label>
    <label><input type="checkbox" name="mailing" value="pops">모닝팝스</label>
  </fieldset>
</form>
```

radio-check.html



사용자 입력을 위한 <input> 태그

- <fieldset>, <legend> 태그 – 폼 요소 그룹으로 묶기

```
form{
  width: 500px;
  border: 1px solid #ccc;
  padding: 10px;
}
fieldset{
  margin-bottom: 16px;
  padding: 10px;
  border: 1px solid #ccc; /* 외곽선 */
}
legend{
  font-weight: bold;
  padding: 0 5px; /* 좌우 여백 */
}
```

여러 줄 입력하는 텍스트 영역

● <textarea> 태그 - 여러 줄 입력하는 텍스트 영역 만들기

- 폼에서 여러 줄의 텍스트를 입력할 때 사용하는 폼이다.
- cols 속성 - 몇 글자를 쓸지 지정
- rows 속성 - 화면에 몇 줄을 표시할 지 지정
- Placeholder 속성 - 힌트 안내문

배송 업무

배송시 요청사항을 적어주세요

전송

여러 줄 입력하는 텍스트 영역

● <textarea> 태그 - 여러 줄 입력하는 텍스트 영역 만들기

```
<!-- 여러줄 입력하는 기능, 멀티라인 태그  
cols(칼럼) 영어기준 글자수, rows-보이는 줄수  
placeholder-힌트글 -->  
<form action="" method="post">  
  <fieldset>  
    <legend>배송 업무</legend>  
    <textarea cols="60" rows="5" name="memo"  
              placeholder="배송시 요청사항을 적어주세요"></textarea>  
  </fieldset>  
  <input type="submit" value="전송">  
</form>
```

textarea.html

```
form{  
  width: 400px;  
}  
textarea{  
  /* 크기 조정 불가 */  
  resize: none;  
}  
/* input 태그의 속성에 스타일을 줄때 [속성='속성값']를 사용함 */  
input[type='submit']{  
  margin-top: 10px;  
  padding: 5px 10px;  
}
```



여러 데이터 나열해 표시하기

● <select> ~<option> 태그

– 목록(리스트) 형태로 여러 데이터를 나열해 표시하고 선택하는 태그

```
<select>  
  <option value="c1">내용1 </option>  
  <option value="c2">내용2 </option>  
  <option value="c3">내용3 </option>  
</select>
```

수강 신청

학과번호	<input type="text"/>
이름	<input type="text" value="홍길동"/>
학과	<div>소프트웨어학과 ▼ 소프트웨어학과 기계공학과 전기전자공학과 컴퓨터공학과</div>

여러 데이터 나열해 표시하기

```
<div id="wrap">
  <form action="process.jsp" method="post">
    <fieldset>
      <legend>수강 신청</legend>
      <!-- value: 입력상자에 표시될 값
      |   readonly: 읽기 전용 -->
      <ul>
        <li>
          <label for="cid">학과번호</label>
          <input type="text" id="cid" name="cid"
            |   autofocus required>
        </li>
        <li>
          <label for="sname">이름</label>
          <input type="text" id="sname" name="sname"
            |   value="홍길동" readonly>
        </li>
      </ul>
    </fieldset>
  </form>
</div>
```

select_option.html



여러 데이터 나열해 표시하기

select.html

```
<li>
  <label for="major">학과</label>
  <select id="major" name="major">
    <option value="software">소프트웨어학과</option>
    <option value="mechanic">기계공학과</option>
    <option value="electric">전기전자공학과</option>
    <option value="computer">컴퓨터공학과</option>
  </select>
</li>
</ul>
</fieldset>
<div>
  <input type="submit" value="신청하기">
  <input type="reset" value="취소하기">
</div>
</form>
</div>
```



여러 데이터 나열해 표시하기

```
#wrap{
    width: 800px;
    margin: 50px auto;
}
form{width: 500px;}
fieldset{margin-bottom: 16px;}
ul li{
    list-style: none;
    margin: 16px;
}
ul li label{
    width: 100px;
    float: left;
}
ul li input{width: 140px; height: 20px;}
select{width: 150px; height: 25px;}
input[type='submit'], input[type='reset']{
    padding: 5px 10px;
}
```

major.css



여러 데이터 나열해 표시하기

● <select> <optgroup> <option> 태그

- <optgroup> : 관련 옵션을 그룹화한 태그

optgroup.html

차를 선택하세요.

- 현대 자동차
 - IONIQ5
 - SONATA
- 기아 자동차
 - EV6
 - K7

```
<form action="" method="post">
  <label for="cars">차를 선택하세요.</label>
  <select name="cars" id="cars">
    <optgroup label="현대 자동차">
      <option value="IONIQ5">IONIQ5</option>
      <option value="SONATA">SONATA</option>
    </optgroup>
    <optgroup label="기아 자동차">
      <option value="EV6">EV6</option>
      <option value="K7">K7</option>
    </optgroup>
  </select>
  <br><br>
  <input type="submit" value="Submit">
</form>
```

파일 첨부 버튼

● <input type="file"> - 파일을 첨부할 때 사용

반품 정보

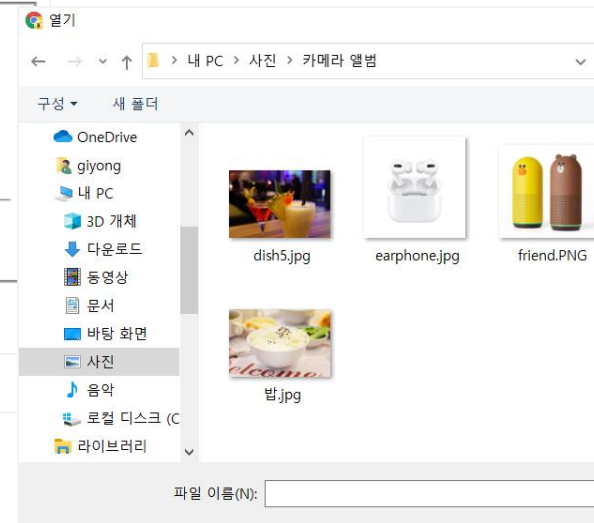
만일 수령한 상품에 문제가 있다면 즉시 **반품 신청**해 주세요.
반품 신청시 상품의 상태를 사진으로 첨부해 주세요.

파일 선택

선택된 파일 없음

반품 신청

취소하기



<h1>레드향 주문하기</h1>

<form>

<fieldset>

<legend>반품 정보</legend>

<p>만일 수령한 상품에 문제가 있다면 즉시 반품 신청해 주세요.</p>

<p>반품 신청시 상품의 상태를 사진으로 첨부해 주세요.</p>

<hr>

<input type="file">

</fieldset>

<div>

<input type="submit" value="반품 신청">

<input type="reset" value="취소하기">

</div>

</form>

```
form{
  width: 500px;
}
input[type="submit"], input[type="reset"]{
  margin: 20px 3px;
  padding: 10px;
}
```



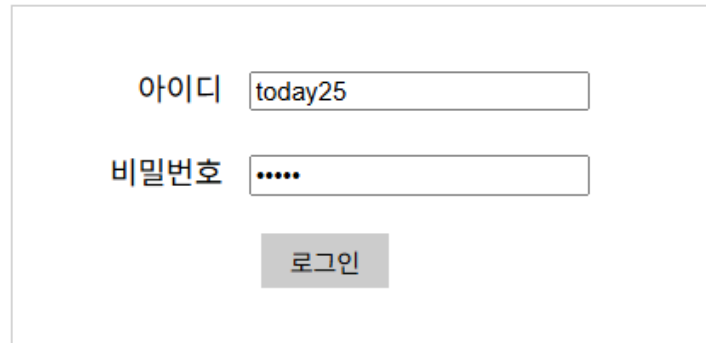
hidden으로 서버에 보내기

- 히든 필드를 만들때 사용하는 **type="hidden"**

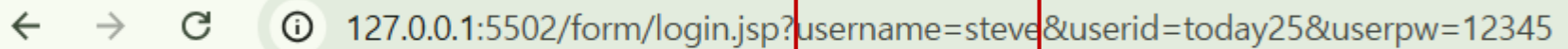
사용자에게는 보이지 않고 서버 페이지로 데이터는 전송됨

`<input type="hidden" name="필드 이름" value="서버로 넘길 값">`

로그인



A login form with two input fields and a submit button. The first field is labeled '아이디' (ID) and contains the text 'today25'. The second field is labeled '비밀번호' (Password) and contains four dots. Below the fields is a button labeled '로그인' (Login).



A browser address bar showing a URL: `127.0.0.1:5502/form/login.jsp?username=steve&userid=today25&userpw=12345`. The query string part is highlighted with a red box.



hidden으로 서버에 보내기

```
<h2>로그인</h2>
<form action="login.jsp" method="get">
  <ul>
    <input type="hidden" name="username" value="steve">
  </ul>
  <li>
    <label for="userid">아이디</label>
    <input type="text" name="userid" id="userid">
  </li>
  <li>
    <label for="userpw">비밀번호</label>
    <input type="password" name="userpw" id="userpw">
  </li>
  <li>
    <button type="submit">로그인</button>
  </li>
</ul>
</form>
```



버튼 <button> 태그

- <button>태그 - 버튼 만들기

<button> 이름 </button> : submit 처럼 서버 페이지로의 전송기능은 없고, **onclick** 속성을 사용하여 주로 자바스크립트 함수와 함께 사용한다.
(단, type 속성에 submit을 명시하면 서버 페이지 전송이 가능함)

로그인

새탭 열기

사이트 이동

127.0.0.1:5500 내용:
아이디나 비밀번호가 일치하지 않습니다.

확인

버튼 <button> 태그

● <button>태그 - 버튼 만들기

```
<!-- alert('msg') : 알림, 경고창, window.open(url) -->
<p>
  <button type="button" onclick="alert('숫자를 입력해 주세요')">전송</button>
</p>
<!-- confirm(): 확인/취소 창 -->
<p>
  <button onclick="confirm('정말로 삭제하시겠습니까?')">삭제</button>
</p>

<p>
  <button onclick="window.open('https://www.gg.go.kr')">경기도청 가기</button>
</p>
```

자바스크립트 window 객체의 open()함수

버튼 <button> 태그

- <button>태그 - 버튼 만들기

```
<p>
  <button onclick="greet()">확인</button>
</p>

<script>
  /*function greet(){
    alert('안녕하세요!')
  }*/

  /*const greet = function(){
    alert('안녕하세요!');
  }*/

  //화살표 함수
  const greet = () => {
    alert('안녕하세요!')
  }
```



〈input〉 태그의 다양한 속성

readonly 속성 – 읽기 전용 필드 만들기

- 사용자가 입력하지 못하고 읽게만 만들 수도 있다.

required 속성 – 필수 필드 만들기

- 내용을 폼에 입력한 후 submit 버튼을 클릭하면 폼을 서버로 전송하는데 이때 필수 필드에 필요한 내용이 모두 채워졌는지 검사해야 한다.
- 필수적으로 입력해야 하는 내용에는 required 속성을 지정해 필수 필드로 만든다.

배송 정보

주문 상품	<input type="text" value="상품용 3KG"/>	readonly – 읽기 전용
이름	<input type="text" value="김기용"/>	
배송 주소	<input type="text"/>	required – 필수 필드
이메일	<input type="text" value="! 이 입력란을 작성하세요."/>	
연락처	<input type="text" value="하이픈 빼고 입력해 주세요"/>	

〈input〉 태그의 다양한 속성

autofocus 속성 – 입력 커서 표시하기

- 폼의 요소 중에서 원하는 곳에 마우스 커서를 표시할 수 있다.

placeholder 속성 – 힌트 표시하기

- 사용자가 텍스트를 입력할 때 도움이 되도록 입력란에는 적당한 힌트 내용을 표시하고 있다가 그 필드를 클릭하면 내용이 사라지도록 만들 수 있다.

배송 정보

주문 상품	<input type="text" value="상품용 3KG"/>
이름	<input type="text"/>
배송 주소	<input type="text"/>
이메일	<input type="text"/>
연락처	<input type="text" value="하이픈 빼고 입력해 주세요"/>

autofocus – 입력 커서

placeholder – 힌트 표시

상품 주문서 만들기

레드향 주문하기

상품 선택

주문할 상품을 선택해 주세요.

- ☐ 선물용 3kg 개(최대 5개)
- ☐ 선물용 5kg 개(최대 3개)
- ☐ 가정용 3kg 개(최대 5개)
- ☐ 가정용 5kg 개(최대 3개)

포장 선택

- ☐ 선물 포장 ☒ 선물 포장 안함

배송 정보

주문 상품

이름

이메일

연락처

배송 주소

배송 지정 (주문일로부터 최소 3일 이후)

메모

주문하기

취소하기



상품 주문서 만들기

```
<div id="container">
  <h2>레드향 주문하기</h2>
  <form action="order.jsp" method="post">
    <fieldset>
      <legend>상품 선택</legend>
      <p>주문할 상품을 선택해 주세요.</p>
      <ul>
        <!-- type=number : 숫자 표시 -->
        <li>
          <label>
            <input type="checkbox" name="ck_gift3" value="g_3">선물용 3kg
          </label>
          <input type="number" name="num_gift3" min="0" max="5">개(최대 5개)
        </li>
        <li>
          <label>
            <input type="checkbox" name="ck_gift5" value="g_5">선물용 5kg
          </label>
          <input type="number" name="num_gift5" min="0" max="3">개(최대 3개)
        </li>
      </ul>
    </fieldset>
  </form>
</div>
```

order.html



상품 주문서 만들기

```
<ul>
  <!-- type=range : 범위 (슬라이드바) 표시 -->
  <li>
    <label><input type="checkbox" name="ck_fa3" value="g_3">가정용 3kg</label>
    <input type="range" min="0" max="5" name="num_fa3">개 (최대 5개)
  </li>
  <li>
    <label><input type="checkbox" name="ck_fa5" value="g_5">가정용 5kg</label>
    <input type="range" min="0" max="3" name="num_fa5">개 (최대 3개)
  </li>
</ul>
<p>포장 선택</p>
<ul>
  <li>
    <label><input type="radio" name="gift" value="yes">선물 포장</label>
    <label class="size"><input type="radio" name="gift"
      value="no" checked>선물 포장 안함</label>
  </li>
</ul>
</fieldset>
```



상품 주문서 만들기

```
<fieldset>
  <legend>배송 정보</legend>
  <ul>
    <!--
      value: 입력상자에 표시될 값
      readonly: 읽기 전용
      autofocus: 페이지 로드 시 커서 위치 지정
      placeholder: 입력상자에 힌트 문구 표시
      required: 필수 입력 제한
    -->
    <li>
      <label for="prod">주문 상품</label>
      <input type="text" id="prod" name="prod"
        value="선물용 3KG" readonly>
    </li>
    <li>
      <label for="cname">이름</label>
      <input type="text" id="cname" name="cname"
        autofocus required>
    </li>
    <li>
      <label for="email">이메일</label>
      <input type="email" id="email" name="email"
        required>
    </li>
  </ul>
</fieldset>
```

order.html



상품 주문서 만들기

```
<li>
  <label for="phone">연락처</label>
  <input type="text" id="phone" name="phone"
    placeholder="하이픈빼고 입력해주세요" required>
</li>
<li>
  <label for="address">배송 주소</label>
  <input type="text" id="address" name="address"
    required>
</li>
<!-- type=date (날짜)-->
<li>
  <label for="d-day">배송 지정</label>
  <input type="date" id="d-day" name="d-day">
  <small>(주문일로부터 최소 3일 이후)</small>
</li>
<li>
  <label for="memo">메모</label>
  <textarea cols="40" rows="3" id="memo"
    name="memo" placeholder="배송시 요청사항을 적어주세요"></textarea>
</li>
</ul>
</fieldset>
<div>
  <input type="submit" value="주문하기">
  <input type="reset" value="취소하기">
</div>
</form>
```

order.html



상품 주문서 만들기

order.css

```
#container{
  width: 960px;
  margin: 20px auto;
}
h2{text-align: center;}
form{
  width: 600px;
  margin: 0 auto; /* 수평 가운데 정렬 */
}
fieldset{
  margin-bottom: 16px;
  padding: 16px; /* 안쪽 여백 */
}
legend{
  font-weight: bold;
  font-size: 1.2em;
}
```

```
ul li{
  list-style: none;
  margin: 16px;
}
ul li label{
  width: 120px;
  float: left;
}
textarea{resize: none;}
input[type='submit'], input[type='reset']{
  cursor: pointer;
  padding: 8px 16px;
  margin-right: 8px;
}
```

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;  
}
```

3. 형제 선택자

- 인접 형제 선택자(+) - 가장 가까운 형제 요소에 스타일 적용
- **기본형** : 상위요소 + 하위요소 (상위 요소와 하위 요소를 '+'를 사용하여 연결)
- 형제 선택자 (~) : 부모가 같은 형제 요소 전 범위를 선택

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

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



연결 선택자

예약 방법 & 사용 요금

아직 온라인 예약 신청이 준비되어 있지 않습니다.
전화(031-1234-0000)로 문의 바랍니다.

펜션 객실 안내

크리스탈 : 60,000원/일

루비 : 75,000원/일

자식 선택자 사용

예약 방법 & 사용 요금

아직 온라인 예약 신청이 준비되어 있지 않습니다.
전화(031-1234-0000)로 문의 바랍니다.

펜션 객실 안내

크리스탈 : 60,000원/일

루비 : 75,000원/일

형제 선택자 사용



연결 선택자

selector.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>
```

```
  <p>펜션 객실 안내</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 요소의 자식 p 요소에 적용 */
```

```
/* section > p {
```

```
  color:blue;
```

```
} */
```

```
/* 형제 요소중 가장 가까운 요소에 적용 */
```

```
section h1 + p{
```

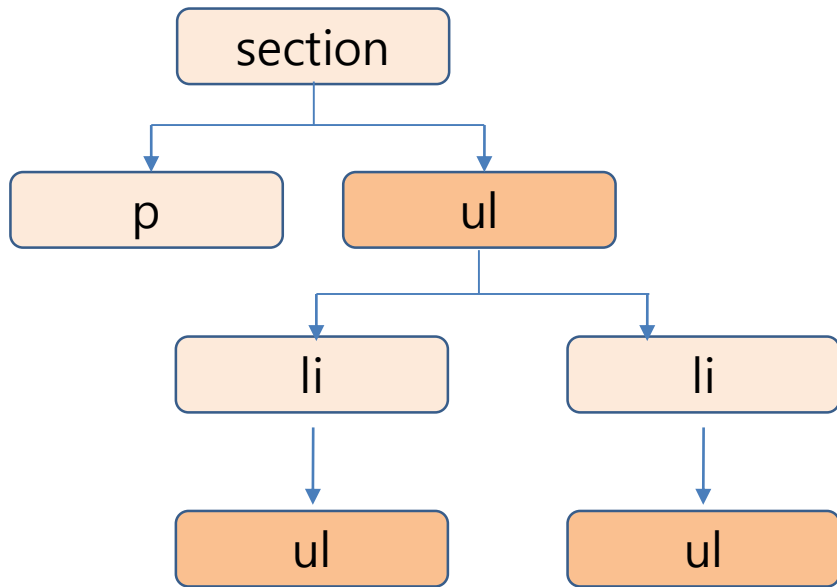
```
  color: red;
```

```
}
```

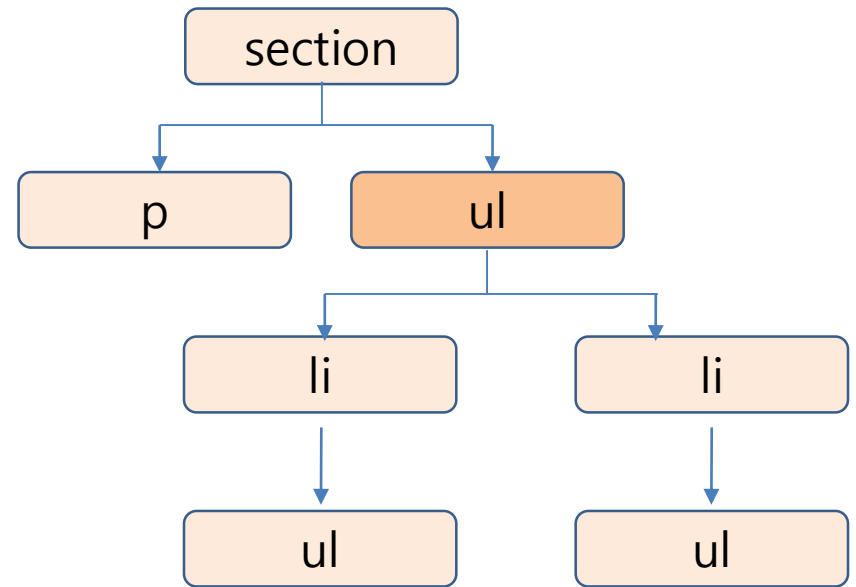


연결 선택자

연결 선택자 비교도.



하위 선택자(후손)



자식 선택자

연결 선택자

child.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

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 */  
}
```

예약 방법 및 요금

행복펜션에 예약하려면?

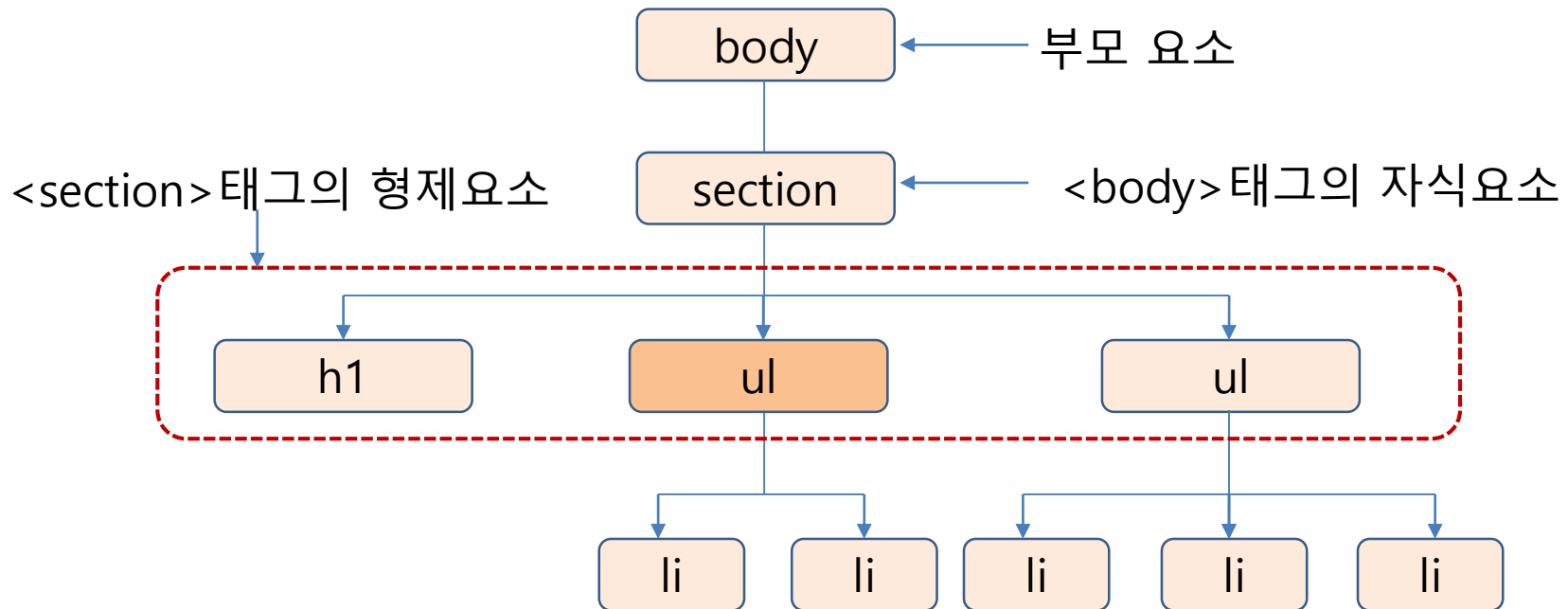
- 예약 방법
 - 직접 통화
 - 문자 남기기
- 요금
 - 1인 : 40,000원
 - 2인 : 60,000원
 - 3인 : 80,000원
 - 4인 : 100,000원

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



연결 선택자

연결 선택자 계층도



연결 선택자

naver1.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>
```

```
/*인접 형제 선택자*/
h1 + ul{
  color: ■blue;
  font-weight: bold;
}
```



연결 선택자

naver2.html

```
<!-- 본문 -->
<section id="wrap">
  <h1>행복 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>
```

```
#wrap{width: 600px;}
/* 형제 요소중 가장 가까운 요소 선택 */
/* #wrap h1 + ul{
  color: #ff00ff;
  border: 1px solid #ccc;
} */
/* 형제 요소는 모두 스타일 적용 */
#wrap h1 ~ ul{
  color: ■#ff00ff;
}
```



속성 선택자

[속성] 선택자

- 주어진 속성과 속성 값이 일치하는 요소를 찾아 스타일 적용
- 태그[속성] 형태로 사용함 - 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) 선택자란 지정하기 어려운 웹 요소를 선택할 수 있는 기능을 하며 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(마우스 클릭)
:focus	웹 요소에 초점이 맞추어졌을 때의 스타일 적용(탭키 누름)

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, a:visited{
    text-decoration: none;
    color: #000;
    padding: 10px;
    font-size: 14px;
}
/* focus - 탭키 누름 */
nav a:hover, a:focus {
    background-color: #222;
    color: #fff;
}
/* 마우스 클릭 */
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

```
/*가상 선택자(:disabled)로 스타일 적용*/
input:disabled{
  background: #ddd;
  border: 1px solid #ccc;
}
/*가상 선택자(:checked)로 스타일 적용*/
input:checked + span{
  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:first-of-type	A요소 중에서 첫번째 요소를 선택
A:nth-of-type(n)	A요소 중에서 n번째 요소를 선택
A:nth-last-of-type(n)	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;
}
table td{
  border: 1px solid #ccc;
  text-align: left;
  padding: 10px 10px 10px 20px;
}
/* n=0을 대입, 짝수인 경우 2n */
table tr:nth-child(2n+1){background: #eee;}
/* tr:first-child{background: #eee;} */
/* tr:last-child{background: #eee;} */
```

