

## HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>텍스트 태그 연습</title>
  </head>
  <body>
    <h1>텍스트 태그 연습</h1>
    <p>텍스트를 표시하고 있습니다.</p>
    <hr/>
    <h2>작은 제목</h2>
    <p>제목의 크기가 바뀌었습니다.</p>
    <p>p 태그는 하나의 문단을 나타냅니다.</p>
    <p>따라서 하나의 p 태그는 자기만의 영역을 가집니다.</p>
  </body>
</html>
```

-----

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>텍스트 태그 연습</title>
  </head>
  <body>
    <h1>텍스트 태그 연습</h1>
    <p>텍스트를 표시하고 있습니다.</p>
    <hr/>
    <h2>작은 제목</h2>
    <p>제목의 크기가 바뀌었습니다.</p>
```

<p>p 태그는 하나의 문단을 나타냅니다.</p>

<p>따라서 하나의 p 태그는 자기만의 영역을 가집니다.

지금 영상을 만들고 있는데... 저희 집 고양이가

너무 방해를 해서 참 힘드네요...^^ 그래도 저는  
제 고양이를 사랑합니다 하하하하하

</p>

</body>

</html>

-----

#### 내용정리

- p 태그는 문단을 표시하는 태그이다.
- h1~h6 태그는 헤드라인을 표시하는 태그이며, 숫자가 작을수록 크다.
- 수평선 hr 은 주제 변경이나 내용 구분 시에 사용하는 태그이다.
- HTML 은 텍스트의 줄바꿈, 공백 등을 엔터와 스페이스로 처리하지 않는다.
- 스페이스는 한 번씩만 허용된다(연달아 입력할 경우 한 번 외엔 무시).
- 태그 br 는 줄바꿈을, 문자조합 &nbsp; 는 공백을 나타낸다.

-----

<!DOCTYPE html>

<html>

<meta charset="utf-8">

<title>태그의 구분과 텍스트 태그</title>

<body>

<h1>제목을 표시하는 큰 텍스트</h1>

<p>문단을 나타내는 작은(?) 텍스트!

```
        이 태그는 블록 요소를 만들까요?  
        아니면 인라인 요소를 만들까요?  
    </p>  
</body>  
</html>
```

-----

```
<!DOCTYPE html>  
<html>  
    <meta charset="utf-8">  
    <title>태그의 구분과 텍스트 태그</title>  
    <body>  
        <h1>제목을 표시하는 큰 텍스트</h1>  
        <p>문단을 나타내는 작은(?) 텍스트!  
            이 태그는 블록 요소를 만들까요?  
            아니면 인라인 요소를 만들까요?  
        </p>  
        <strong>굵은 글자를 만들고 싶어요</strong>  
        <em>기울여진 글자를 만들고 싶어요</em>  
        <mark>형광색으로 칠한 글자를 만들고 싶어요</mark>  
    </body>  
</html>
```

-----

```
<!DOCTYPE html>  
<html>  
    <meta charset="utf-8">  
    <title>태그의 구분과 텍스트 태그</title>
```

```
<body>
  <h1>제목을 표시하는 큰 텍스트</h1>
  <p>문단을 나타내는 작은(?) 텍스트!
    <strong>p 태그 안에서 굵게 표시하기</strong>
    이 태그는 블록 요소를 만들까요?
    <mark>p 태그 안에서 밝게 표시하기</mark>
    아니면 인라인 요소를 만들까요!?
  </p>
</body>
</html>
```

-----

```
<!DOCTYPE html>
<html>
  <meta charset="utf-8">
  <title>태그의 구분과 텍스트 태그</title>
  <body>
    <h1>제목을 표시하는 큰 텍스트</h1>
    <p>문단을 나타내는 작은(?) 텍스트!
      <strong>p 태그 안에서 굵게 표시하기</strong>
      이 태그는 블록 요소를 만들까요?
      <p>블록 안에 블록을 넣으면 블록 안에 층이 생겨버려요!</p>
      <mark>p 태그 안에서 밝게 표시하기</mark>
      아니면 인라인 요소를 만들까요!?
    </p>
  </body>
</html>
```

- HTML 문서의 요소는 블록 레벨 요소와 인라인 요소로 구분할 수 있다.
- 블록 레벨 요소는 너비(공간)를 모두 차지하여 블록을 형성한다.
- 인라인 요소는 콘텐츠를 표시하기 위해 필요한 공간만 차지한다.

-----

<이미지를 표시해보자 이미지 태그를 사용하는 방법>

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>이미지를 표시해보자</title>
  </head>
  <body>
    <h1>이미지를 표시해볼게요!</h1>
    
    <p>이미지 표시에는 img 태그</p>
  </body>
</html>
```

-----

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>이미지를 표시해보자</title>
  </head>
  <body>
    <h1>이미지를 표시해볼게요!</h1>
    
    <p>이미지 표시에는 img 태그</p>
  </body>
</html>
```

내용 정리

- 이미지를 표시할 때는 닫는 태그가 없어도 되는 img 태그를 사용한다.
- src 속성은 이미지의 url 을 입력받는 속성이다.
- alt 속성은 대체 텍스트를 입력받는다. 이미지 유실 상황에 대응할 수 있다.
- width 와 height 는 각각 이미지의 너비와 높이를 입력받는 속성이다.

-----

## <영역을 구분해보자 - 컨테이너 요소 & 전역 속성>

콘텐츠나 레이아웃에 아무런 영향도 주지않고, 단지 다른 요소 여럿을 묶어

관리하기 편하게 만드는 역할을 하는 태그를 `컨테이너`라고 한다.

콘텐츠 내용을 구분하거나, 공통적인 스타일을 적용하고자 할 때 개발자는 컨테이너를 사용하는 것이 좋다.

다음 두 가지 태그가 컨테이너 역할을 담당하는 태그들이다.

- <div></div>:블록 레벨 컨테이너
- <span></span>:인라인 컨테이너

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>영역을 구분해보자</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>일단 문서를 만들자</h1>
```

```
    <hr>
```

```
    <div>
```

```
      <h2><span>공부</span> 다짐하기</h2>
```

```
      
```

```
      <p>펭귄군은 공부를 열심히 하기로 했다!</p>
```

```
    </div>
```

```
    <hr/>
```

```
<div>
  <h2><span>운동</span> 열심히 하기</h2>
  <p>건강을 위해 운동을 열심히 하자!</p>
</div>
</body>
</html>
```

-----

- id:요소에 고유한 이름을 부여하는 식별자 역할 속성.
- class:요소를 그룹 별로 묶을 수 있는 식별자 역할 속성.
- style:요소에 적용할 CSS 스타일을 선언하는 속성.
- title:요소의 추가 정보를 제공하는 텍스트 속성. 사용자에게 툴팁 제공.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>영역을 구분해보자</title>
  </head>
  <body>
    <h1>일단 문서를 만들자</h1>
    <hr>
    <div id="study">
      <h2 class="title"><span>공부</span> 다짐하기</h2>
      
      <p>펭귄군은 공부를 열심히 하기로 했다!</p>
    </div>
    <hr/>
    <div id="workout">
      <h2 class="title"><span>운동</span> 열심히 하기</h2>
      <p>건강을 위해 운동을 열심히 하자!</p>
    </div>
  </body>
```

</html>

## 내용 정리

- 콘텐츠나 레이아웃에 영향을 주지 않는 태그 요소를 컨테이너라 한다.
- 컨테이너는 영역을 묶어서 관리하고자 할 때 사용한다.
- 모든 태그에 공통적으로 지정 가능한 속성을 전역 속성이라 한다.
- id 와 class 는 태그명이 같은 요소들을 식별하기 위해 사용하는 속성들이다.
- id 는 태그당 하나씩만 지정해야 하나, class 는 다중 지정 및 중복 가능하다.
- 특정 태그에만 지정 가능한 속성들도 물론 존재한다.

-----

2022/11/24

## #9 링크 만들기

링크(anchor)란 현재 문서에서 다른 문서로 이동할 수 있는 수단이다.

```
<a href="https://www.naver.com">
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>링크 만들기</title>
```

```
  <body>
```

```
    <a href="https://www.naver.com/">
```

```
      <네이버로 갑시다!>
```

```
    </a>
```

```
    <br>
```

```
    <a href="https://www.google.com/">
```



```
        구글로 갑시다!
    </a>

</body>
</head>
</html>
```

-----

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>링크 만들기</title>
    <body>

      <a href="https://www.naver.com/" target="_self">
        네이버로 갑시다!
      </a>
      <br>
      <a href="https://www.google.com/" target="_blank">
        구글로 갑시다!
      </a>

      <br><br>

      <a href="tel:010-1234-5678">010-1234-5679</a>
      <br>
      <a href="mailto:asdfjkl@naver.com">
        asdfjkl@naver.com
      </a>

    </body>
  </head>
```

</html>

전화걸기, 메일 보내기

내용 정리

- a 태그는 페이지를 다른 위치로 이동시킬 수 있는 링크를 만든다.
- href 속성에는 링크가 참조하는(이동하고자 하는)url 을 입력한다.
- target 속성에는 새 페이지가 열릴 대상(탭)을 지정할 수 있다.
- href 속성을 통해 전화 걸기, 이메일 보내기 등을 지정할 수도 있다.

-----

## #10 목록 표시하기

목록은 연관 있는 항목(item)들을 나열한 것을 의미한다.

HTML 목록은 '순서 없는 목록'과 '순서 있는 목록'으로 구분된다.

순서 없는 목록 Unordered List <ul></ul>

순서 있는 목록 Ordered List <ol></ol>

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>목록 표시하기</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>귀여운 동물 목록</h1>
```

```
    <ul>
```

```
      <li>강아지</li>
```

```
      <li><strong>고양이</strong></li>
```

```
      <li>두더지</li>
```

```
      <li>햄스터</li>
```

```
    </ul>
```

```
<hr/>
<h1>프로그램 공부순서</h1>
<ol>
  <li><mark>HTML</mark></li>
  <li>CSS</li>
  <li>JAVASCRIPT</li>
  <li>NODEJS</li>
</ol>
</body>
</html>
```

## 내용정리

- 목록은 연관 있는 항목들을 나열한 것이다.
- 목록 요소는 '순서 없는 목록'과 '순서 있는 목록'으로 구분할 수 있다.
- 목록 안에 들어가는 항목 요소를 표시할 때는 li 태그를 사용한다.
- ol,ul,li 태그는 모두 블록 레벨 요소를 표시하는 태그이다.

-----

## #11 입력 요소 만들기

input <input />

사용자로부터 값을 입력받을 수 있는 대화형 컨트롤(또는 '필드')을 나타낸다.  
기본적으로 인라인 요소이며, 단일 태그이다.

input의 핵심, type 속성

type의 값에 따라 입력 요소의 형태나 입력 데이터 유형 등이 달라진다.  
사용 가능한 type은 20여 가지이며, 기본값은 text이다.

모든 속성에 대한 정보

<https://developer.mozilla.org/ko/docs/Web/HTML/Element/Input>

```
<input type="text" name="nickname" />
```

```
<input type="text" name="job" />
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>입력 요소 만들기</title>
  </head>
  <body>
    <input type="text"> <br><br>
    <input type="button"> <br><br>
    <input type="color"> <br><br>
    <input type="range"> <br><br>
    <input type="date"> <br><br>
  </body>
</html>
```

-----

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>입력 요소 만들기</title>
  </head>
  <body>
    <input type="text" maxlength="10"
    placeholder="메시지 입력..!"> <br><br>
    <input type="button" value="PUSH"> <br><br>
    <input type="color"> 색을 골라 보세요!<br><br>
    <input type="range" max="100" min="0"
    step="10"> <br><br>
    <input type="date"> <br><br>
    <input type="radio"> <br><br>
```

```
        <input type="checkbox"> <br><br>
    </body>
</html>
```

-----

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>입력 요소 만들기</title>
  </head>
  <body>
    <input name="text" type="text" maxlength="10"
    placeholder="메시지 입력..!"> <br><br>

    <input name="push" type="button"
    value="PUSH"> <br><br>

    <input name="color" type="color">
    색을 골라 보세요!<br><br>

    <input name="score" type="range"
    max="100" min="0"
    step="10"> <br><br>

    <input name="birthday" type="date"> <br><br>
  </body>
</html>
```

## 내용 정리

- input 태그는 입력 요소를 만드는 태그이다.
- input 태그는 type 속성을 통해 형태와 입력 값 유형을 결정한다.
- type 값에 따른 사용 방법이 다양하다.  
(웹페이지 내에서의 역할, 추가 속성 등)

- name 속성으로 input 요소를 구별할 수 있다.

## #12 select 그리고 textarea

### input 외 입력 요소들

select 는 다수의 옵션(선택지)을 포함할 수 있는 선택 메뉴이다.  
메뉴 안에 포함되는 옵션은 option 태그를 사용해 표시한다.

select 에는 input 과 마찬가지로 name 을 지정할 수 있으며, 각각의 option 에는 value 속성을 지정할 수 있다. value 는 실제로 처리될 값을 나타낸다.

```
<select name="cafe">
  <option value="starbucks">스타벅스</option>
  <option value="co">스타벅스</option>
  <option value="starbucks">스타벅스</option>
  <option value="starbucks">스타벅스</option>
</select>
```

textarea 는 여러 줄의 일반 텍스트를 입력할 수 있는 입력 요소이다.  
textarea 역시 name 을 추가하여 구별해줄 수 있는 입력 요소이다.

```
<textarea name="letter"></textarea>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>다양한 입력 요소 만들기</title>
  </head>
```

```

<body>
  <h1>키우고 싶은 동물 고르기</h1>
  <select name="pet" multiple>
    <option value="dog">강아지</option>
    <option value="cat" selected>고양이</option>
    <option value="hamster">햄스터</option>
    <option value="parrot">앵무새</option>
  </select>
  <br>

  <textarea name="contents" cols="10" rows="10">기본적으로 쓰여 있는
  텍스트</textarea>
</body>
</html>

```

## 내용 정리

- select 는 선택 메뉴를 나타내는 태그이다.
- select 의 각 선택지는 option 태그를 통해 표시한다.
- 각각의 option 에는 value 를 지정하여 추후 처리를 준비한다.  
(자바스크립트를 통한 입력 값 처리에 value 가 사용될 수 있다)
- textarea 는 여러 줄의 일반 텍스트를 입력하는 요소를 나타낸다.

-----

## #13 폼(form)Part1 서버와 클라이언트

form 을 학습하기 전에

form 은 사용자가 입력한 데이터(입력값)를 서버로 보내기 위해 사용하는 태그.  
그런데 서버가 뭐지?  
서버란, 정보를 제공하는 호스트(host)이다!

## #14 폼(form)Part2 폼 태그

form 은 입력 요소들을 감싸며, 입력 값을 서버 측으로 제출(submit)할 수 있다.

```
<form>
  <input type="text" placeholder="아이디">
  <br>
  <input type="text" placeholder="비밀번호">
</form>
```

form 의 내용(입력값)을 제출하기 위해 input 태그의 submit 타입 사용 가능!

```
<form>
  <input type="text" placeholder="아이디">
  <br>
  <input type="text" placeholder="비밀번호">
  <br>
  <input type="submit" value="로그인">
</form>
```

로그인 버튼을 누르면 입력된 아이디와 비밀번호가 서버로 전송되고(요청), 서버 측에서 데이터를 처리한 결과를 클라이언트에게 보내준다(응답).

form 의 속성

action : 입력값을 전송할 서버의 url

method : 클라이언트가 입력한 데이터를 어떤 식으로 전송할지(GET or POST)

```
<form action="exmaple.php" method="POST">
</form>
```

위 예제는

example.php 라는 서버 프로그램으로 입력값을 전송하여 요청할 것이다.

POST 방식으로 전송할 것이다.



## GET vs POST

GET : 서버에 요청을 보내어 응답을 받아낸다.  
서버로부터 정보를 '가져오겠다'는 성격의 요청이다.

POST : 서버에 요청을 보내어 작업을 수행한다.  
서버에 있는 데이터를 추가/수정/삭제 한 후에 응답을 받아낸다.  
서버의 정보를 '조작하겠다'는 성격의 요청이다.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>서버에 요청을 보내자, 폼!</title>
  </head>
  <body>
    <h1>키우고 싶은 동물 고르기</h1>
    <form action="exam.php" method="POST">
      <input type="text" placeholder="NAME" name="name">
      <br>
      <select name="pet">
        <option value="dog">강아지</option>
        <option value="cat">고양이</option>
        <option value="parrot">앵무새</option>
      </select>
      <br>
      <input type="submit" value="전송">
    </form>
  </body>
</html>
```

내용 정리

- form 은 사용자가 입력 요소에 입력한 데이터를 전송해준다.
- 서버란 데이터를 제공하는 호스트이다.
- 클라이언트란 데이터를 제공받아 이용하는 고객(사용자)이다.
- form 은 입력 요소(input, select, textarea 등)를 감싸고, 데이터를 제출(submit)한다.
- form 의 action 은 서버 측 주소를 지정하는 속성이다.
- form 의 method 는 데이터 전송 방식을 지정하는 속성이다.

## -----

## #15 메타 태그(meta)

### meta 태그

meta 태그는 HTML 문서에 대한 메타데이터를 정의한다.

메타데이터란 데이터에 대한 데이터, 즉 '정보'를 의미한다.

meta 태그는 항상 head 태그의 안에 들어가며, 일반적으로 문자 세트, 페이지 설명, 키워드, 문서의 작성자 및 뷰포트 설정을 지정하는 데 사용된다.

meta 태그를 사용하는 이유?

웹페이지에 대한 정보를 제공 하므로

검색엔진이 페이지를 검색할 때 참고할 수 있고, 검색 결과에도 반영할 수 있다.

meta 태그가 제공하는 메타데이터의 유형&속성은?

- charset:문자 세트
- http-equiv:콘텐츠 속성 정보에 대한 http 헤더
- name:문서 정보
- content:메타데이터 내용

### charset

문자 인코딩에 대한 요약 정보를 기입하는 속성이다. 문자 인코딩이란 한글을 표시하기 위해 문자 세트를 지정하는 작업으로, 영문과 한글을 모두 사용하기 위해

utf-8 방식을 사용하는 것이 좋다.

코드 예

```
<meta charset="utf-8">
```

http-equiv

콘텐츠 속성의 정보/값에 대한 HTTP 헤더를 제공한다.

HTTP 란 인터넷에서 데이터를 주고 받을 수 있는 프로토콜이다.

코드 예

```
<meta http-equiv="x-ua-compatible" content="IE=edge">
```

브라우저의 최신 버전의 엔진을 사용하라는 뜻

```
<meta http-equiv="refresh" content="10">
```

10 초마다 페이지 새로고침하라는 뜻

name

name 속성을 이름으로, content 속성을 값으로 하여 문서 정보를 이름 + 값 쌍의 형태로 제공할 때 사용할 수 있다.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <meta http-equiv="x-ua-compatible" content="IE=edge">
```

```
    <meta name="author" content="유노코딩">
```

```
    <meta name="keywords" content="메타태그, 학습, 코딩교육">
```

```
    <meta name="description" content="메타태그를 추가하는 방법">
```

```
    <title>메타태그 추가하기</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>메타태그 추가하기</h1>
```

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

## 내용 정리

- meta 태그는 HTML 문서의 메타데이터(정보)를 표시하는 태그다.
- 검색엔진 최적화에 기여하며, 검색 결과에도 영향을 끼친다.
- charset 는 문자 인코딩에 대한 요약 정보를 기입하는 속성이다.
- http-equiv 는 콘텐츠 속성의 정보/값에 대한 HTTP 헤더를 제공한다.
- name 속성은 문서의 다양한 정보를 제공할 수 있는 메타데이터의 이름이다.
- content 는 메타데이터의 구체적인 내용을 기입하는 속성이다.

-----

## #16 뷰포트 (viewport)

뷰포트(viewport)란 현재 화면에 보여지고 있는 영역을 의미한다.

기기 별로 뷰포트가 다르기 때문에, 동일한 웹 페이지라도 기기에 따른 배율 조정이 발생해 화면의 크기가 다르게 보이는 현상이 나타난다.

아래 그림은 동일한 HTML 문서를 각각 PC 와 모바일로 확인한 결과이다.

왜 이런 현상이?

태블릿, 스마트폰 등 모바일 기기가 등장하기 전에는 웹 페이지가 컴퓨터 화면만을 위해 설계 되었다.

컴퓨터 화면에서의 웹 페이지는 웹브라우저라는 소프트웨어를 통해 페이지 크기를 조절해가며 웹을 조회할 수 있지만, 모바일 기기에서는 고정된 사이즈(스크린 크기)로 조회해야 하므로 웹의 모든 콘텐츠를 표시하기 위해서는 배율 조정을 해야만 한다.

때문에 PC 용 웹페이지와 모바일 웹페이지를 따로 만드는 경우도 있다!

name="viewport"

기기 별로 뷰포트가 다르기 때문에 발생하는 배율 문제에 대응하기 위해 meta

태그를 통해 뷰포트 관련 설정을 추가할 수 있다.

코드 예

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

뷰포트의 너비를 단말기 너비에 맞추고, 초기 배율을 1로 한다!

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <meta name="viewport"
```

```
    contents="width=device-width, initial-scale=1.0">
```

```
    <title>뷰포트 체험하기</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>안녕하세요!</h1>
```

```
    <h1>감사해요!</h1>
```

```
    <h1>잘있어요!</h1>
```

```
    <h1>다시만나요!</h1>
```

```
  </body>
```

```
</html>
```

내용 정리

- 뷰포트는 현재 화면에 보여지고 있는 영역을 의미한다.
- 웹 문서는 기본적으로 PC 화면에 맞춰 설계 되었기 때문에, 다른 기기에서 배율 조정이 일어난다.
- 배율 조정 현상에 대응하기 위해 개발자는 메타 태그를 사용해 뷰포트 관련 설정을 추가할 수 있다.