

# HTML5

이 승

# 개발 환경

- 크롬 웹 브라우저
- vscode

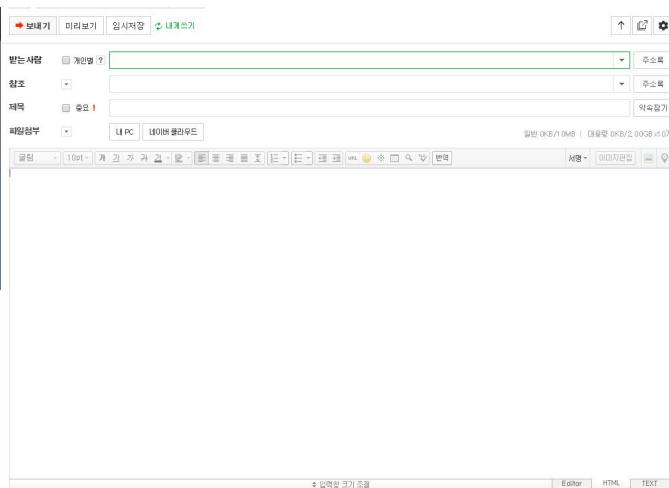
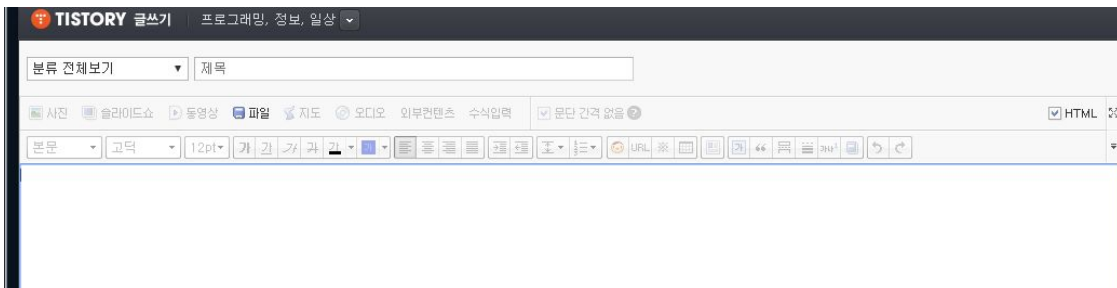
# HTML이란?

- **Hyper Text Markup Language** 웹 페이지를 위한 마크업 언어
  - 마크업 언어란? 태그 등을 이용하여 문서나 데이터의 구조를 명기하는 언어의 한 가지.
  - 하이퍼 텍스트란? 웹 사이트에서 링크를 클릭해 다른 문서나 사이트로 이동하는 기능.
- 제목, 단락 등의 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법 제공

```
<!doctype html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

# HTML은 우리 근처에

- 블로그 게시물을 작성할 때 원하는 대로 게시글을 작성할 수 있다.
- 이메일을 보낼 때도 **html**으로 작성할 수 있다.



# HTML tag 규칙

- <>를 사용해 구분
- 소문자로 사용 (대소문자 구분 x, 표준 명세에서 모두 소문자로 사용할 것을 권장)
  - ex) <div></div> <DIV></DIV> 모두 같은 태그를 의미
- 여는 태그가 있다면 닫는 태그를 정확히 써줘야 한다.
- <img>, <br>태그처럼 닫는 태그가 없는 경우도 있다.
- 태그에 속성 값을 넣을 수 있다.
  - <tag attribute="attribute" attribute2="attribute2"></tag>
- 포함 관계를 명확히 해야한다.
  - <tag1><tag2></tag2></tag1>

# 간단한 html파일 만들기

1. html 파일을 저장할 임시 폴더 생성
2. html 파일 생성 후 코드 작성

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4      <meta charset="UTF-8">
5      <title>sample</title>
6  </head>
7  <body>
8      hello world
9  </body>
10 </html>
```

# html 구조

- `<!DOCTYPE html>` - 현재 문서가 HTML5 언어로 작성된 웹 문서. 웹 브라우저에게 해당 문서가 어떤 유형의 문서인지 알려줌
- `<html lang="ko"></html>` - 웹 문서의 시작을 알리는 태그. lang속성으로 문서에서 사용할 언어를 지정할 수 있다. (참고: [https://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes))
- `<head>`  
    `<meta charset="UTF-8">`  
    `<title>sample</title>`  
    `</head>`  
    **head tag** - 브라우저에 정보를 주는 태그. 화면에 보이지는 않지만 웹 브라우저가 알아야 할 정보들은 모두 **head**에 입력. 외부 파일등도 입력.  
  
    **title tag** - 문서 제목을 표시하는 태그. 웹 브라우저 상에서 제목 표시줄에 나오는 문자열을 입력하는 태그.  
  
    **meta tag** - 문자 인코딩 및 문서 키워드, 요약 정보등을 입력. 웹 문서 정보를 검색 엔진에게 전달해 주는 데 사용할 수 있다.
- `<body>Hello html</body>` **body tag** - 실제 브라우저에 표시 될 내용.

# 특수 기호 사용하기

- HTML 소스에서 공백이나 태그에 사용되는 괄호등을 소스 기호로 인식하지 않고 그대로 표시하기 위해 사용.
- 공백을 특수기호로 사용하지 않고 여러 번 사용할 경우 한 번만 입력 된다. 이 때 공백 특수기호를 사용해서 띄어쓰기를 여러 번 할 수 있다.
- 특수 기호 참조 사이트 ( [https://www.w3schools.com/html/html\\_entities.asp](https://www.w3schools.com/html/html_entities.asp) )



# 텍스트 tag

- 블록 tag - 태그를 적용시 줄바꿈이 됨
  - `<h1></h1>` - 제목 표시하기. n에 들어가는 것은 1~6까지의 숫자. 1이 가장 크고 6이 가장 작은 크기를 표현한다.
  - `<p></p>` - 단락 태그. 텍스트를 표시할 때 자주 사용하는 tag. 텍스트를 한 줄로 표시하는데 브라우저 창의 너비보다 길어질 경우 자동으로 줄바꿈이 된다.
  - `<br>` - 줄바꿈 태그. html file에서 줄바꿈을 할 경우 공백으로 입력되어 제대로 표시되지 않음. 이때 `<br>`태그를 사용하여 줄바꿈을 할 수 있다. 달는 태그는 없어도 된다.
  - `<pre></pre>` - 입력하는 그대로 화면에 표시. 이 태그 안에서는 공백과 줄바꿈 모두 그대로 화면에 나온다.

# 실습

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8">
5   <title>block level tag</title>
6 </head>
7 <body>
8   <h1>h1 tag</h1>
9   <h2>h2 tag</h2>
10  <h3>h3 tag</h3>
11  <h4>h4 tag</h4>
12  <h5>h5 tag</h5>
13  <h6>h6 tag</h6>
14  <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum
    has been the industry's standard dummy text ever since the 1500s, when an unknown printer
    took a galley of type and scrambled it to make a type specimen book. It has survived not
    only five centuries, but also the leap into electronic typesetting, remaining essentially
    unchanged. It was popularised in the 1960s with the release of Letraset sheets containing
    Lorem Ipsum passages, and more recently with desktop publishing software like Aldus
    PageMaker including versions of Lorem Ipsum.</p>
15  <br><br><br><br>
16  <pre>
17    function example () {
18      const text = 'hello world';
19      console.log(text);
20    };
21    example();
22  </pre>
23 </body>
24 </html>
```

# 텍스트 tag

- 인라인 tag - 태그를 적용해도 줄바꿈이 발생하지 않음
  - `<strong></strong>`, `<b></b>` - 굵게 표시하기, 두 개의 차이는 화면 낭독기에서의 차이  
`strong`태그는 굵게 표시하며 강조의 의미를 내포, `b`태그는 굵게 표시만 한다.
  - `<em></em>`, `<i></i>` - 이탤릭체로 표시하기, 두 개의 차이는 화면 낭독기에서의 차이  
`em`태그는 이탤릭 폰트로 표시하며 강조의 의미를 내포, `i`태그는 이탤릭 폰트로 표시만 한다.
  - `<span></span>` - 줄바꿈 없이 영역 묶기. 특정 텍스트 영역에 스타일을 입히거나 할 때 가장 많이 사용한다.

# 실습

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4      <meta charset="UTF-8">
5      <title>inline level tag</title>
6      <style>
7          .bold {
8              color: ■ #FF0000;
9          }
10
11          .italic {
12              color: ■ #00FF00;
13          }
14      </style>
15  </head>
16  <body>
17      <p>
18          <strong class="bold">Lorem Ipsum</strong> is <b class="bold">simply</b> dummy <em
19              class="italic">text</em> of the <i class="italic">printing</i> and typesetting industry.
20          <span style="background-color: ■ #0000FF">Lorem Ipsum</span> has been the industry's
21              standard dummy text ever since the 1500s, when an unknown printer took a galley of type
                and scrambled it to make a type specimen book. It has survived not only five centuries,
                but also the leap into electronic typesetting, remaining essentially unchanged. It was
                popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum
                passages, and more recently with desktop publishing software like Aldus PageMaker
                including versions of Lorem Ipsum.
            </p>
        </body>
    </html>
```

# 목록을 만드는 태그

- 순서가 필요없는 목록<ul></ul>(unordered list)
  - 순서가 필요없는 목록을 만들기 위해선 <ul></ul>태그 안에 <li></li>태그를 넣으면 된다.
  - list-style-type을 이용하여 목록을 표시하는 불릿을 수정할 수 있다.

```
<ul>  
  <li>순서가</li>  
  <li>필요없는</li>  
  <li>목록</li>  
</ul>
```

# 목록을 만드는 태그

- 순서가 필요한 목록<ol></ol>(ordered list)
  - 순서가 필요한 목록을 만들기 위해선 <ol></ol>태그 안에 <li></li>태그를 넣으면 된다.
  - **type** 속성으로 순서를 나타내는 문자를 변경할 수 있다. (css의 list-style-type으로도 가능)
  - **start** 속성을 사용하여 시작하는 숫자를 임의로 변경할 수 있다.
  - **reversed**속성을 사용하여 항목을 역순으로 표시할 수 있다.

```
<ol>  
  <li>순서가</li>  
  <li>필요한</li>  
  <li>목록</li>  
</ol>
```

## type

Indicates the numbering type:

- 'a' indicates lowercase letters,
- 'A' indicates uppercase letters,
- 'i' indicates lowercase Roman numerals,
- 'I' indicates uppercase Roman numerals,
- and '1' indicates numbers (default).

# 목록을 만드는 태그

- 목록을 만드는 태그는 중첩해서 사용할 수 있다.
- 바깥 목록 태그의 <li></li>태그 안에 추가 할 목록 태그를 입력하면 된다.

```
<ul>  
  <li>  
    <ol>  
      <li>순서가</li>  
      <li>필요한</li>  
      <li>목록</li>  
    </ol>  
  </li>  
  <li>  
    <ul>  
      <li>순서가</li>  
      <li>필요없는</li>  
      <li>목록</li>  
    </ul>  
  </li>  
</ul>
```

# 실습

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8">
5   <title>list tag</title>
6 </head>
7 <body>
8   <h3>unordered tag</h3>
9   <ul>
10     <li>순서가</li>
11     <li>필요없는</li>
12     <li>목록</li>
13   </ul>
14   <h3>ordered tag</h3>
15   <ol>
16     <li>순서가</li>
17     <li>필요한</li>
18     <li>목록</li>
19   </ol>
20   <h3>중첩된 목록</h3>
21   <ul>
22     <li>
23       <ol>
24         <li>순서가</li>
25         <li>필요한</li>
26         <li>목록</li>
27       </ol>
28     </li>
29     <li>
30       <ul>
31         <li>순서가</li>
32         <li>필요없는</li>
33         <li>목록</li>
34       </ul>
35     </li>
36   </ul>
37 </body>
38 </html>
```



# 목록을 만드는 태그

- <dl><dt><dd> - 설명 목록(description list) 만들기

```
<dl>  
  <dt>설명 목록</dt>  
  <dd>설명 목록 태그는 사전 형식처럼 사용할 수 있다.</dd>  
  <dt>두 번째 설명 목록</dt>  
  <dd>list 처럼 여러 개의 dt, dd태그를 사용할 수 있다.</dd>  
</dl>
```

# 이미지 태그

- <img>태그 사용

- 
- src속성을 사용해 이미지 파일이 있는 경로를 알려줘야 한다.
- 지원하는 이미지 포맷 (<https://developer.mozilla.org/ko/docs/Web/HTML/Element/img> 참조)
- width, height 속성으로 이미지의 크기를 조절할 수 있음.
- alt 속성으로 이미지에 설명을 추가할 수 있음.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>sample</title>
</head>
<body>
  <h2>image tag</h2>
  
</body>
</html>
```

# 링크 만들기

- <a>태그
  - <a href="주소" [속성="값"]> text </a>
  - href에 링크할 주소를 입력
  - 이미지 태그와 함께 사용하면 이미지 링크가 된다.
  - href외 다른 속성 <https://developer.mozilla.org/ko/docs/Web/HTML/Element/a> 참조

# 실습

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4      <meta charset="UTF-8">
5      <title>sample</title>
6  </head>
7  <body>
8      <h2>link tag</h2>
9      <ul>
10         <li><a href="http://www.naver.com">naver로 이동</a></li>
11         <li><a href="http://www.nba.com"></a></li>
12         <li><a href="inline-level.html">inline-tag</a></li>
13     </ul>
14 </body>
15 </html>
```

## a 태그 앵커로 사용하기

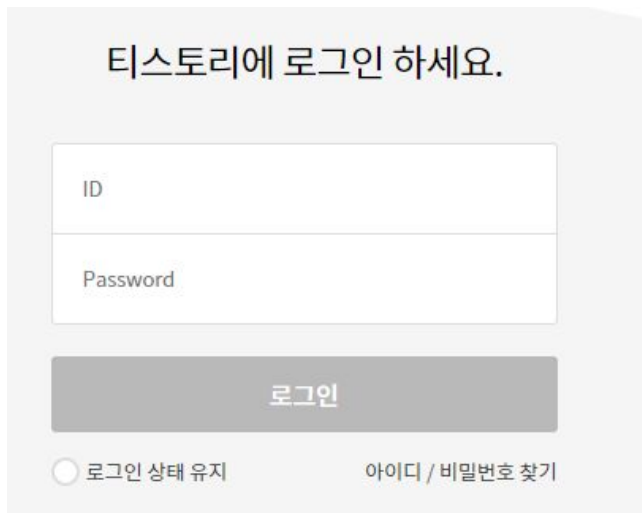
- 한 문서 안에 다른 곳으로 이동하고 싶으면 a태그와 id속성으로 앵커 기능을 만들 수 있다.
- 이 때 href에 주소가 아니라 # + id명으로 이동할 태그의 id를 지정해준다.
- `<tag id="id명"></tag>`
- `<a href="#id명"></a>`

# 실습

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4      <meta charset="UTF-8">
5      <title>anchor tag</title>
6  </head>
7  <body>
8      <h2>anchor</h2>
9      <ul>
10         <li><a href="#html">html</a></li>
11         <li><a href="#css">css</a></li>
12         <li><a href="#javascript">javascript</a></li>
13     </ul>
14     <p id="html">
15         HTML은 하이퍼텍스트 마크업 언어(HyperText Markup Language, 문화어: 초본문표식말기언어,
16         하이퍼본문표식말기언어)라는 의미의 웹 페이지를 위한 지배적인 마크업 언어다. HTML은 제목, 단락,
17         목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로
18         구조적 문서를 만들 수 있는 방법을 제공한다. 그리고 이미지와 객체를 내장하고 대화형 양식을
19         생성하는 데 사용될 수 있다. HTML은 웹 페이지 콘텐츠 안의 각쇠 괄호에 둘러싸인 "태그"로
20         되어있는 HTML 요소 형태로 작성한다. HTML은 웹 브라우저와 같은 HTML 처리 장치의 행동에 영향을
21         주는 자바스크립트와 본문과 그 밖의 항목의 외관과 배치를 정의하는 CSS 같은 스크립트를 포함하거나
22         불러올 수 있다. HTML과 CSS 표준의 공동 책임자인 W3C는 명확하고 표상적인 마크업을 위하여 CSS의
23         사용을 권장한다. (https://ko.wikipedia.org/wiki/HTML)
24     </p>
25     <p id="css">
26         종속형 시트 또는 캐스케이딩 스타일 시트(Cascading Style Sheets, CSS)는 마크업 언어가 실제
27         표시되는 방법을 기술하는 언어로[1], HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다.
28         W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다. (https://ko.wikipedia.org/wiki/%EC%A2%85%EC%86%8D%ED%98%95\_%EC%8B%9C%ED%8A%B8)
29     </p>
30     <p id="javascript">
31         자바스크립트(영어: JavaScript)는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹
32         브라우저 내에서 주로 사용하며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고
33         있다. 또한 Node.js와 같은 런타임 환경과 같이 서버 사이드 네트워크 프로그래밍에도 사용되고
34         있다. 자바스크립트는 본래 넷스케이프 커뮤니케이션스 코퍼레이션의 브렌던 아이크(Brendan Eich)가
35         처음에는 모카(Mocha)라는 이름으로, 나중에는 라이브스크립트(LiveScript)라는 이름으로
36         개발하였으며, 최종적으로 자바스크립트가 되었다. 자바스크립트가 썬 마이크로시스템즈의 자바와 구문
37         (syntax)이 유사한 점도 있지만, 이는 사실 두 언어 모두 C 언어의 기본 구문에 바탕을 뒀기
38         때문이고, 자바와 자바스크립트는 직접적인 관련성이 없다. 이름과 구문 외에는 자바보다 셸프와
39         유사성이 많다. (https://ko.wikipedia.org/wiki/%EC%9E%98%B8%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8)
40     </p>
41 </body>
42 </html>
```

# form 태그

- 로그인이나 회원가입 등 사용자가 웹 사이트로 정보를 보낼 수 있는 요소들은 모두 폼이라고 할 수 있다.
- 폼은 서버에 입력한 데이터를 전달하고 처리는 서버 프로그래밍을 이용해서 처리한다.



티스토리예 로그인 하세요.

ID

Password

로그인

☐ 로그인 상태 유지      [아이디 / 비밀번호 찾기](#)

# form 태그

- <form> 태그

- <form [속성="값"]>form 요소</form> 의 형식으로 사용한다.
- method 속성을 통해 서버에 어떤 방식으로 넘길것인지 등을 지정한다. (post, get)
  - get - 주소 표시줄에 사용자가 입력한 내용이 그대로 드러난다. 보안상의 문제가 있을 수 있다.
  - post - 대부분 이 방식을 사용. 입력 내용의 제한을 받지 않고 입력한 내용이 주소표시줄에 드러나지 않음.
  - <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form>
- name - 폼의 이름을 지정
- action - 입력된 내용을 처리해 줄 프로그램을 지정



form태그

이 외의 태그들

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

# 웹 개발에 관심이 있다면

- html 및 css를 이용한 웹 페이지 만들기
- vanilla javascript
- 각종 프레임워크 (react, vue, angular 등)
- 웹팩
- 백엔드 (spring, nodejs 등)
- graphql
- 등등..

## 참고하면 좋은 사이트

- <https://caniuse.com/>
- <http://gs.statcounter.com/>
- <https://webclub.tistory.com/491> (html4 vs html5)
- <https://developer.mozilla.org/ko/>
- <https://github.com/Knowre-Dev/WebDevCurriculum>