

HTML

Hyper text markup language : 하이퍼 텍스트 마크업 랭귀지

> 마크업과 마크다운 차이점 설명:

https://samsara-ku.dev/common_sense/difference-between-markup-and-markdown/ / <https://blog.cordelia273.space/15>

마크업 - 마크로 둘러싸인 언어. 태그(tag)로 둘러싸여있음. 마크의 형식과 규칙을 정의한 언어. 문서의 골격에 해당하는 부분을 작성하는데 사용.

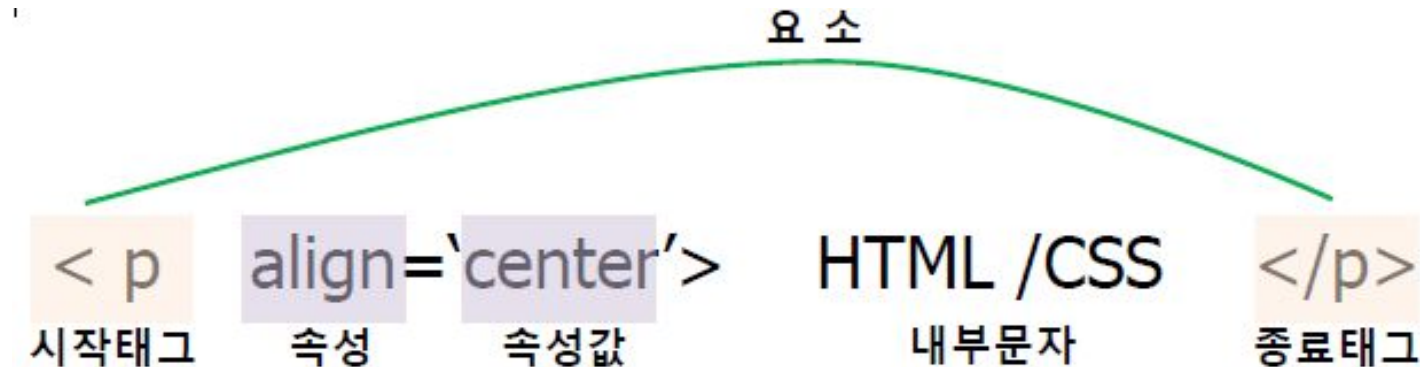
마크다운-마크업 언어의 일종으로 존 그루버, 아론 스워츠가 만듦. 읽고 쓰기 쉽다.(마크다운 파일 ex: readme.md..md확장자 파일임)

웹페이지를 작성하기 위해 사용되는 언어로 웹 브라우저에게 보일 문자열과 이걸 감싸는 태그(<div>같은...)로 이뤄져있음.

- 마크업(태그): 문서의 논리적인 구조를 정의하고 출력장치에 어떤 형태로 보여질건지 지시하는 역할

HTML 5가지 특징 & 구성요소

1. 구조적 설계 지원(시멘틱)
2. 그래픽 및 멀티미디어 기능 강화
3. CSS3, JavaScript 지원
4. 다양한 API 제공
5. 모바일 웹 지원, 장치 접근 가능(배터리 정보, 카메라, GPS 등)
6. 웹 브라우저 서버와 양방향 통신 가능
7. 인터넷이 연결되지 않은 상태에서도 애플리케이션이 동작함.



HTML 구성요소

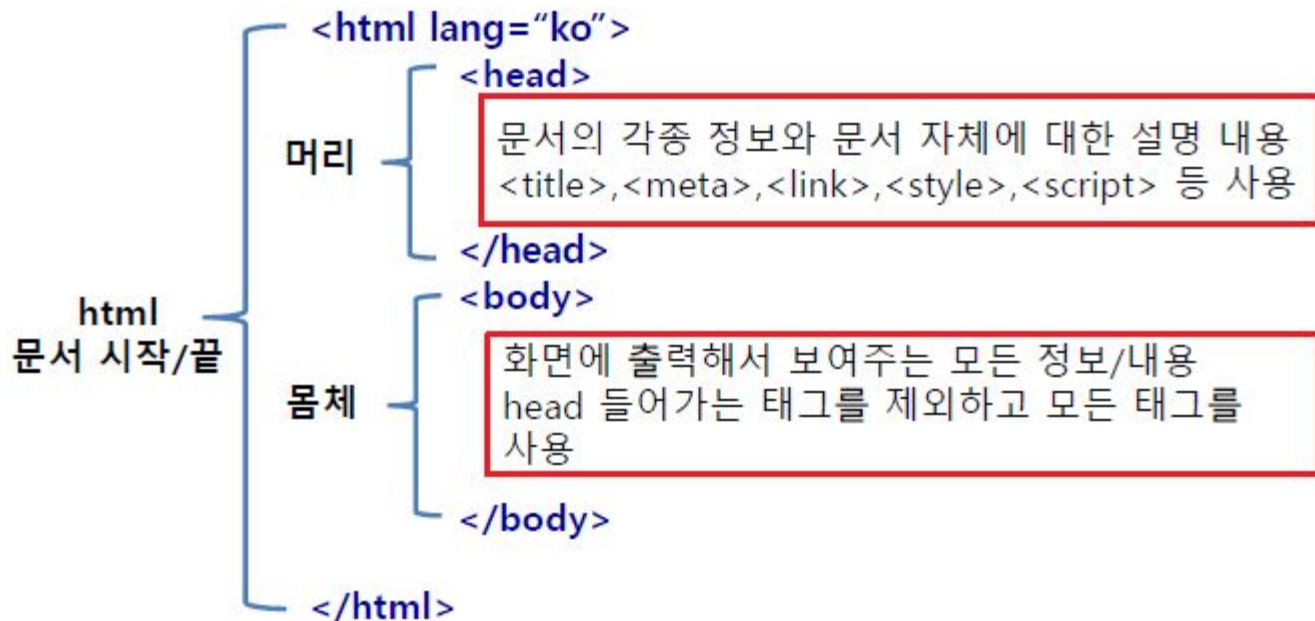
1. 태그: '<'랑 '>'로 묶인 명령어. 시작태그와 종료태그를 쓸때 이용됨 (ex: `<div></div>`)
2. 요소: 시작태그와 종료태그로 이뤄진 모든 명령어. 하나의 HTML문서는 요소의 집합임.(그냥 시작태그~종료태그의 모든걸 요소라고 함)
3. 속성: 요소의 시작태그에만 사용. 명령어 구체화 역할. (ex: `style`)
4. 변수/속성값: 속성이 가지는 값. 입력할때 ""이나 "를 이용한다. (ex: 스타일의 " "; 이걸 생각하면 될듯)

HTML 작성시 주의사항

1. 태그는 대소문자를 구분하지 않지만 소문자를 권장함
2. 시작태그로 시작하면 반드시 종료태그로 종료함
3. 파일 확장자는 꼭 `html`, `htm`으로 설정
4. 문자 공백은 한개만 인식함. 공백을 추가하기 위해선 ` `; 개행을 하기 위해선 `
`을 쓰면 됨
5. 주석은 `<!-->`로 여러줄이나 한줄 사용 가능.
6. ! 누르고 탭을 하면 `<head>`에 대해 자동작성됨.

HTML 기본 구조

문서유형 `<!doctype html>`



목록을 만드는 법 3가지

1. `내용` : unordered list의 약자. 순서가 필요없는 목록을 만든다.

블록요소임. life-style로 스타일 지정가능

2. `내용` : ordered list의 약자. 숫자나 알파벳 등 순서가 있는 목록을 만드는데 사용 (1,2,3,4....)

3. `<dl>내용</dl>` : definition list의 약자. 사전처럼 용어를 설명하는 목록을 만든다.

1. ul의 속성으로 **disc**, **circle**, **square**가 있음(기본값은 **disc**)
2. li의 경우 속성값 (**1**,**a**, **A**, **i** 등)에서 기본값은 **1**.
3. **start** : li의 시작값. **start = ""**; 안에 숫자를 하면 숫자의 시작이 **1**이 아닌 **5**부터 시작됨
4. **reversed** : 이게 있으면 항목순서가 거꾸로됨

1. 다른파일로(다른 페이지) 넘어가게 됨. 예를들어 문자관련 를 쓰고, Ctrl + 클릭 "01_text.html" 을 하면, 파일을 생성할 수 있음
2. 바뀐 페이지에서 맨 첫줄에 인덱스로 돌아가기 를 작성하면 돌아가기 가능. (인덱스 파일로 돌아가는 코드) 여기서 index.html은 돌아가고자 하는 파일명 이름이다.

<head></head>

1. head 태그는 페이지 설정정보. 화면엔 보이지 않는다. 이게 보이려면 <body>에 적어야함.
2. head에 올 수 있는건 만약 table(표)를 만든다면 그 표에 대한 줄 설정과 색깔을 어떻게 할건지 설정을 할

```
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>table</title>
9   <style>
10     table {
11       border: 1px solid ■black;
12       border-collapse: collapse;
13       padding: 10px;
14     }
15
16     th,
17     td {
18       border: 1px solid ■black;
19     }
20   </style>
21 </head>
```

head의 테이블 border 속성(테두리)

```
/*1px :보더 라인의 두께  
solid :보더 유형  
black :보더 색*/  
</style>
```

1. border 속성은 테이블의 테두리를 설정해주는 속성. ex (border="1") 숫자가 높을수록 테두리가 두꺼워짐
2. background-color는 배경색 지정이 가능.
3. border-collapse: - 표의 테두리와 셀의 테두리 사이의 간격을 없앤것. 겹치는 부분은 한줄로. (이걸 하지 않으면 그냥 2줄선의 표가 탄생)
4. width와 height는 테이블의 크기를 결정해주는 속성입니다. ex(width=50px width=100%)픽셀단위로 크기,비율로 줄 수 있음
5. padding - 셀 패딩. 셀 내용과 테두리 사이의 공간을 말함.

table 태그 정리

1. `<th>내용</th>`: table head 약자로, 표의 제목을 쓰는 역할
2. (기본값은 굵은 글씨체에, 중앙 정렬)
3. `<tr> 내용 </tr>` : table row 약자로, 가로줄을 만드는 역할
4. (기본값은 보통 글씨체에, 왼쪽 정렬)
5. `<td>내용 </td>`: table data 약자로, 셀을 만드는 역할
(기본값은 보통 글씨체에, 왼쪽 정렬)

1. table ✎

- 표를 나타내는 HTML 태그
- 내부에 행을 나타내는 tr과 셀을 나타내는 th, td 태그 등이 사용됨

2. tr ✎

- "table row"
- 표 내부에서 행을 나타내는 태그

3. th ✎

- "table header"
- 헤더 칸(cell)을 나타내는 태그
- 기본 스타일: 중앙 정렬(text-align:center), 두껍게(font-weight:bold)

4. td ✎

- "table data"
- 일반적인 칸(cell)을 나타내는 태그
- 기본 스타일: 왼쪽 정렬(text-align:left)

thead, tbody

[개요]

제목 줄을 가지고 있는 경우 사용하는 태그로 <thead>, <tbody>, <th>를 사용하여 구현합니다.

[속성]

- <thead> : 표의 제목을 나타내는 태그로 <table>하위, <tr> 상위에 위치합니다.
- <tbody> : 표의 본문 영역을 나타내는 태그로 <thead>와 같은 위치에 존재합니다.
- <th> : 제목 셀을 나타내는 태그로 <td>태그 대신 사용됩니다.

태그요소

1. **block** 요소 : 화면 끝까지 영역을 차지. **br**을 쓰지 않아도 다음칸이 생김
 - 블록요소는 **padding**과 **margin** 값을 지정해서 상하좌우 여백을 넣는게됨.
 - 블록요소는 너비를 작게 설정해도 가로영역을 모두 차지함. 다른요소가 옆으로 오는게 불가
 - 화면 너비전체를 차지하는 요소. **div, p, h1~h6, table...**
2. **inline** 요소 : **content**(문장)만큼만 영역차지.
 - 너비와 높이가 내부 컨텐츠 크기만큼 설정됨
 - **width: 값; height: 값;** 이런 속성이 **default**나 다름없음
 - 하지만 **width, height**로 크기를 지정하진 못함
 - **padding, margin**으로 좌우에만 여백 지정가능
 - 여러개 인라인 요소가 있으면 수평으로 쌓임. 보통 **text**를 작성할때 사용됨
 - **content**만큼만 영역을 차지하는 요소. **span, strong, em, del, img, a...**

태그요소

3. 인라인 블록 요소: 너비와 높이가 내부 콘텐츠 크기만큼 설정

- width, height로 너비 높이 설정 가능
- padding, margin으로 상하좌우여백 지정가능
- 인라인요소처럼 수평으로 쌓임
- 인라인 블록 요소를 생성하려면 css의 display: "inline-block";으로 스타일 적용 필요

참고 사이트: <https://memostack.tistory.com/275>

문자관련태그

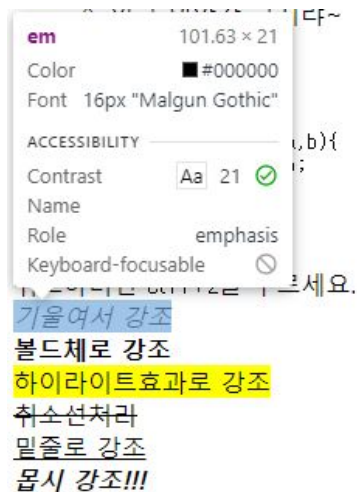
1. <h1><h2>.... ~ <h6>까지
heading이란 뜻. 블록요소
2. <p>태그
paragraph라는 뜻. 문단태그이며 블록요소임
3. 태그,
, 공백문자
은 짧은 문장 태그. 인라인 요소임. br은 개행. 는 한개마다
한칸의 공백.

특수문자들, 기타 태그 설명

1. 특수문자들 : <http://kor.pe.kr/util/4/charmap2.htm> 에서 확인가능
2. <pre>태그 : pre-formatted 태그. 블록요소임
3. <hr>태그 : 수평선 태그
4. <blockquote>태그 : 인용구 태그. screen reader, 포탈 검색봇.
5. 축약어: acronym 태그. 아래 밑줄이 그어짐...
6. <code> 태그: 코드 작성이 가능한 태그
7. <kbd> 키보드 값을 나타낼때 쓰는 태그 <kbd>Ctrl</kbd><kbd>Z</kbd>
8. pharse tag : 강조용 (기울이기, 볼드체, <mark> 하이라이트, 취소선 처리, <ins> 밑줄강조 태그가 있음)

추가로 알아두면 좋을 것들

- div{div\$}*3 : <div>div1</div><div>div2</div><div>div3</div>
-> 이렇게 div가 3개가 나옴.
- 태그{태그\$}*개수: 동일 태그를 여러개 만드는 것



 태그, 표 관련 태그 추가설명

1. 블록요소. list style로 스타일 지정가능. 시작숫자 지정가능함.
2. <dl> <dt><dd> 태그 - 셋 다 설명목록 태그이다. dl = description list , dt = definition title , dd = define description. 블록요소임
3. 표 관련 태그

```
table
  thead 헤더행 묶음
    tr
      th | td
  tbody 본문행 묶음
    tr
      th | td
  tfoot 정리행 묶음
    tr
      th | td
```

표 관련 태그 설명

- | | |
|--------------------------------|--------------------------|
| 1. border : 1px solid black; | 헤더에 표 1px짜리 검정색 고정길이를 표시 |
| 2. border-collapse : collapse; | 표 실선 사이의 공백 없음처리 |
| 3. padding : 10 px; | 표 주변을 10 px 만큼 공백처리 |

테이블 조절 응용

1. colspan = " n " -> 테이블 행 길이 n으로 지정
2. rowspan = " n " -> 테이블 열 길이 n으로 지정
3. width = " n px " -> 테이블 가로길이 n픽셀로 지정
4. height = " n px " -> 테이블 높이 n픽셀로 지정

 설명

- 이미지는 body에 들어감.
- img 시작과 종료태그엔 구분없이 하나의 태그만 작성 가능
- 가 기본 품임
- src 다음에 들어가는 건 이미지 경로(절대경로/상대경로 존재)
- alt는 대체 텍스트 이미지를 못찾거나, 시각적 확인이 불가능한 경우 텍스트를 적음

절대경로

1. 프로토콜주소 (<http://192.168.0.13~:5500/이미지주소.png>)
2. ~파일명/이미지/이미지명.png

상대경로

1. 홈 상대주소
2. ./ 현재 디렉토리(자식디렉토리) , ../부모디렉토리

단, 파일이 위치한 경로복사를 한 주소(컴퓨터 경로)로는 이동불가.

base64 인코딩

1. 참고주소 : <https://velog.io/@byeol4001/Base-64%EC%99%80-base64-img-%EC%82%AC%EC%9A%A9%ED%95%98%EA%B8%B0>
2. 형식으로 이뤄짐.

절대크기와 상대크기

절대크기: px

- 픽셀(화소)

상대크기: %, em, rem, rh, vw

- **em**: 현재 요소의 **font-size**에 비례한 크기 조절
> 2.0em 일 때 현재 요소에 설정된 **font-size**의 2배가 됨
- **rem**: 현재 페이지 html태그에 설정된 **font-size**에 비례한 크기 조절
> 2.0rem일 때 현재 html태그에 설정된 **font-size**의 2배가 됨
- **vw (viewport width)**: 현재 웹 페이지를 보고있는 프로그램의 너비에 기반
> 1vw일 때 현재 viewport width의 1%를 의미함
- **vh (viewport height)**: viewport의 높이에 기반한 크기 조절
> 1vh일 때 현재 viewport height의 1%를 의미함

참고 사이트: <https://baek.dev/post/38/>

절대크기와 상대크기

절대크기: px

- 픽셀(화소)

상대크기: %, em, rem, rh, vw

- em: 현재 요소의 font-size에 비례한 크기 조절
> 2.0em 일 때 현재 요소에 설정된 font-size의 2배가 됨
- rem: 현재 페이지 html태그에 설정된 font-size에 비례한 크기 조절
> 2.0rem일 때 현재 html태그에 설정된 font-size의 2배가 됨
- vw (viewport width): 현재 웹 페이지를 보고있는 프로그램의 너비에 기반
> 1vw일 때 현재 viewport width의 1%를 의미함
- vh (viewport height): viewport의 높이에 기반한 크기 조절
> 1vh일 때 현재 viewport height의 1%를 의미함

참고 사이트: <https://baek.dev/post/38/>,

<https://velog.io/@conqueror-g/%EB%B6%80%EB%AA%A8-%EC%9A%94%EC%86%8CParent-Element-%EC%9E%90%EC%8B%9D-%EC%9A%94%EC%86%8CChild-Element>

링크관련(target)

- a 태그의 href 속성(hyper reference)에 명시된 문서로 이동 가능함
- 절대 주소: / 또는 protocol
- 상대 주소: 현재 페이지 기준으로 경로 평가

target 속성

- _self : 현재 페이지에서 이동
- _blank : 빈 페이지에서 오픈함

```
<a href="https://naver.com">네이버로</a>  
<a href="/01_html/index.html">인덱스로(절대주소)</a>  
<a href="./index.html" target="_self">인덱스로(상대주소)</a>  
<a href="./index.html" target="_blank">인덱스로(상대주소2)</a>
```

#bookmark, #anchor

- lorem으로 텍스트를 랜덤으로 붙일수있음.
- <https://post.naver.com/viewer/postView.nhn?volumeNo=25195497>

```
<h2>한 페이지의 특정 부분으로 이동 #bookmark #anchor</h2>
<ul>
  <li><a href="#header1">Hello World1</a></li>
  <li><a href="#header2">Hello World2</a></li>
  <li><a href="#header3">Hello World3</a></li>
  <li><a href="#::~text=Hello%20World3-,Lorem%20ipsum%20dolor%20sit%20amet%20consectetur,-adipisicing%20elit.>
</ul>
```

한 페이지의 특정 부분으로 이동 #bookmark #anchor

- [Hello World1](#)
- [Hello World2](#)
- [Hello World3](#)
- [내용참조](#)

lorem, top

- lorem으로 텍스트를 랜덤으로 붙일수있음.
- <https://post.naver.com/viewer/postView.nhn?volumeNo=25195497>
- (h3#header\${hello world}+p>Lorem*3)*3 이러면 랜덤으로 lorem의 문장이 3문장 3개가 복사됨
- a href="#top" : 맨 위로 올라간다.

```
<!-- (h3#header${hello world}+p>Lorem*3)*3-->
<h3 id="header1">Hello World1</h3>
<p><span>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sunt laboriosam minima molestiae ullam providen
<h3 id="header2">Hello World2</h3>
<p><span>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Temporibus quod dignissimos consequatur sit te
<h3 id="header3">Hello World3</h3>

<p><span>Lorem ipsum dolor sit amet consectetur adipisicing elit. Et voluptas laboriosam officiis mollitia aspern
<h3 id="header1"></h3>
<p><span>Lorem ipsum dolor sit amet consectetur adipisicing elit. Harum, minima soluta. Fugit ea nostrum, rerum c
```

[illegible]

멀티미디어(노래, 비디오) 관련

- audio 태그, video 태그, src 속성
- controls 제어기능 시각화 여부
- autoplay 자동재생, muted 음소거

홈페이지에 음악, 동영상 삽입

1. 음악 삽입: `<audio src="~.mp3"></audio>`
2. 여기에 `controls="controls"`를 넣는다면 플레이어 버튼이나 정지버튼과 같이 비디오의 실행을 제어할 수 있는 제어가 표시됨.
3. `controls` 속성은 `boolean` 속성임(해당속성을 명시하지 않으면 자동으로 값은 `false`)
4. `<video src="~.mp4"></video>` 안에 `autoplay`와 `muted`를 넣으면 자동재생이 되는 영상의 노래가 자연히 음소거되어 나옴. `width, height`로 영상의 크기를 정할 수 있다.

멀티미디어(노래, 비디오) 관련

- `iframe(<iframe></iframe>)`으로 유튜브에서 자유롭게 영상을 골라 올릴 수 있음.
(유튜브의 임베드 링크를 갖고 오는 것.)

```
<audio src="../../sample/audio/major.mp3" controls="controls"></audio>
<audio src="../../sample/audio/major.mp3" controls="controls autoplay muted"></audio>

<video src="../../sample/video/small.mp4" controls="controls"></video>
<video src="../../sample/video/small.mp4" controls="controls autoplay muted width="300px"></video>
<iframe
width="560"
height="315"
src="https://www.youtube.com/embed/uRZDg9zHahM"
title="youtube video player"
frameborder="0"
allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
```

폼 관련

1. (form에 대해 제대로 들어가기 전에 미리 적어둔다) **margin**에 대한 설명 >
<https://abooooks.tistory.com/78>
2. form은 사용자 입력 양식 태그. 서버로 사용자 입력값 전송이 가능.
 - **action**은 폼을 제출할 서버 **url**임
 - **method**는 전송방식 **GET**(기본값)과 **POST**가 있음
 - > **GET** : 사용자입력값을 **url**에 직렬화해서 전송. 단순조회시 사용
 - > **POST** : 사용자 입력값을 **http** 메시지 **body**에 작성함. (단, **url**에 입력값이 노출되지 않음) 데이터 입력, 수정, 삭제 용도로 사용
 - form 태그 하위에 **name** 속성이 있는 **input** 태그값, **select** 태그값, **textarea** 값 등을 제출함
 - **global** 속성
 - > **id**: 페이지 내에 태그 식별자. 페이지 내에서 중복이 있음 안됨
 - 폼 관련 속성
 - > **name**: 폼 전송시 사용할 속성명. **name** 속성이 없으면 전송x
 - > **value**: 미리 입력값을 적어두는 것

폼 관련

- form의 기본 태그 형식.
- form을 적으면 네모난 칸이 생기는 거나 다름없음. 거기에 label(라벨)을 붙이면 네모칸의 이름을 말하는 것.
- label의 for 속성은 id와 연결됨(동일해야함)

```
<form action="" method="post">
  <label for="username">이름</label>
  <input type="text" name="username" id="username" placeholder="이름을 입력하세요" value="신사임당">
  <br>

  <!--label의 for 속성은 id와 연결됨-->
  <label for="email">이메일</label>
  <input type="email" name="email" id="email" placeholder="이메일을 입력하세요.">

  <br><br>
  <!--<input type="submit" value="">-->
  <!--input[type=submit]-->
  <input type="submit" value="등록">
</form>
```

폼관련

이름

이메일

폼 관련

```
<form action="" method="get">
  <fieldset>
    <legend>text관련</legend>
    <label for="userId">아이디 : </label>
    <input type="text" name="userId" id="userId" value="honggd" required>
    <br>
    <!--Label+input:password-->
    <label for="password">비밀번호 : </label>
    <input type="password" name="password" id="password" maxLength="12">
    <br>
    <label for="search">검색 : </label>
    <input type="search" name="search" id="search">
    <br>
    <label for="email">이메일 : </label>
    <input type="email" name="email" id="useremail" placeholder="이메일을 입력하세요.">
    <br>
    <label for="url">홈페이지 : </label>
    <input type="url" name="url" id="url" value="http://">
    <br>
    <label for="tel">전화번호 : </label>
    <input type="tel" name="url" id="tel">
    <br>
    <label for="memo">메모 : </label>
    <textarea name="memo" id="memo" cols="30" rows="10" style="resize : none;">a
```

폼 관련

- 위의 그림처럼 **legend** 태그를 선언하면 작은 칸과 이름을 둘러싼 테두리 생성.
- **textarea**는 여러줄의 문장을 적을 수 있고, 큰 네모로 넓이와 높이를 지정할 수 있음
- 제출버튼은 `<input type="submit" value="제출">` (`<button></button>` 태그도 있음. 둘이 별 차이는 없다.)

```
<form action="">
  <fieldset>
    <legend>숫자</legend>
    <label for="amount">수량 : </label>
    <input type="number" name="amount" id="amount"
      min="0" max="10" step="2" value="4">
    <br>
    <label for="score">점수 : </label>
    <input type="range" name="score" id="score"
      oninput="console.log(this.value);" value="100">
  </fieldset>
  <button type="submit">제출</button>
</form>
```

숫자

수량 : 4

점수 :

제출

폼 관련

- date, time, datetime-local로 날짜, 시각, 날짜-시각을 한번에 보게 만들 수 있다.

```
<from>
<fieldset>
  <legend>날짜/시간</legend>
  <!--label+input:date-->
  <label for="date">날짜</label>
  <input type="date" name="date" id="date"
    value="1999-09-09"
    placeholder="날짜를 입력하세요.">
  <br>
  <label for="time">시간</label>
  <input type="time" name="time" id="time"
    value="21:38"
    placeholder="시간을 입력하세요.">
  <br>
  <label for="datetime-local">시각</label>
  <input type="datetime-local" name="datetime-local" id="datetime-local"
    value="2022-05-19T22:27"
    placeholder="날짜/시각을 입력하세요.">
</fieldset>
<button type="submit">제출</button>
```

날짜/시간

날짜	1999-09-09	📅
시간	오후 09:38	🕒
시각	2022-05-19 오후 10:27 📅	

제출

폼 관련

- radio와 checkbox는 각각 이렇게 생겼다.

```
<form action="">
  <fieldset>
    <legend>선택관련</legend>
    <!--radio 제시된 옵션중 택일해야 하는 경우. 취소는 불가능하다.-->
    <span>성별 : </span>
    <input type="radio" name="gender" id="gender0" value="남"><label for="gender0">남</label>
    <input type="radio" name="gender" id="gender1" value="여"><label for="gender1">여</label>
    <br>
    <!--checkbox는 제시된 옵션 중 복수개 선택이 가능한 경우 사용할-->
    <span>취미 : </span>
    <!------>
    <input type="checkbox" name="hobby" id="hobby1" value="축구"><label for=""hobby1">축구</label>
    <input type="checkbox" name="hobby" id="hobby2" value="농구"><label for=""hobby2">농구</label>
    <input type="checkbox" name="hobby" id="hobby3" value="야구"><label for=""hobby3">야구</label>
    <br>
```

성별 : ○남 ○여

취미 : □축구 □농구 □야구

폼 관련

- option을 하면 아래로 쭉 내려서 고를 수 있게 할 수 있다.
- selected로 하면 처음 뜨는 항목을 지정할 수 있다.
- datalist : option에 없는 목록 값도 사용자 입력으로 받을 수 있다.
- select : 사용자가 선택할 수 있는 선택항목을 제공하고 그 범위내에서만 선택 가능.
- 참고 주소: <https://hianna.tistory.com/439>

```
<label for="nation">국적 : </label>
<select name="nation" id="nation">
  <option value="kor" selected>한국</option>
  <option value="jpn">일본</option>
  <option value="usa">미국</option>
</select>
<!--multiple 속성을 추가하면 복수개 선택 가능-->

<br>
<label for="pet">반려동물 : </label>
<input type="text" name="pet" id="pet" list="pets">
<datalist id="pets">
  <option value="강아지">
  <option value="고양이">
  <option value="가재">
```

국적 :

반려동물 :

- 색 변경을 하지 못하게 하려면 ? 을 이용하면 변경되지 못한다.
- `input type="reset" value="초기화"` 하면 초기화버튼 생성&초기값으로 돌려줌
- `input`과 `button` 타입의 차이점: <https://cocoder16.tistory.com/18>

```
<form action="">
  <fieldset>
    <legend>기타</legend>
    <label for="color">컬러 : </label>
    <input type="color" name="color" id="color" value="#30ab5f">
    <br>
    <label for="file">파일 : </label>
    <input type="file" name="file" id="file">
    <br>
    <label for="">안보이지만 전송 : </label>
    <input type="hidden" name="abc" value="가나다">
    <br>
    <input type="button" value="버튼">
    <input type="submit" value="제출">
    <input type="reset" value="초기화"><!-- 초기값으로 돌려줌-->
    <br>
    <button type="button">버튼</button>
    <button type="submit">제출</button> <!-- button 태그의 type 속성 기본값은 submit 이다-->
    <button type="reset">초기화</button>
  </fieldset>
  <button>제출</button>
</form>
```