

## 2. HTML 태그

### 1) HTML 태그 소개

- HTML 버전이 업그레이드 되면서 태그가 새로 추가되기도 하고 삭제되기도 합니다. 현재 태그의 개수는 대략 130여 개 정도입니다. 관련 통계 사이트를 보면, 실제로 대다수의 웹 페이지는 대략 25개 정도의 다른 태그가 사용된다고 합니다. <https://www.advancedwebranking.com/html/#overview>

- html, head, body, title 등 기본적으로 들어가는 태그까지 포함되기 때문에, 실제로 주로 사용하는 태그는 이보다 더 적을 것으로 예상합니다.

<http://html5doctor.com/element-index/>

<https://www.w3schools.com/tags/default.asp>

### 2) 제목과 단락요소

- 제목 태그(Heading) : 문서 내에 제목을 표현할 때 사용하는 태그입니다. 태그 이름은 heading을 줄여서 h로 쓰며, 제목의 레벨에 따라서 <h1>~<h6>까지 있습니다. 보통 <h1>은 해당 페이지를 대표하는 큰 제목으로 주로 사용되며, 숫자가 올라갈수록 조금 더 낮은 수준의 소제목이 나타내게 됩니다. 하지만, 현재 웹 페이지의 내용은 제목과 본문 식의 문서 형태보다는 주로 이미지나 그림처럼 시각적인 형태로 표현되고 있어서 제목 태그를 <h6>까지 쓰는 경우는 거의 없습니다.

<h1>역사</h1>

<h2>개발</h2>

1980년, 유럽 입자 물리 연구소(CERN)의 계약자였었던 물리학자 팀 버너스리가 HTML의 원형인 인콰이어를 제안하였다.

... 이하 생략

<h2>최초 규격</h2>

HTML 최초의 일반 공개 설명은 1991년 말에 버너스리가 처음으로 인터넷에서 문서를 "HTML 태그"(HTML tag)로 부르면서 시작되었다.

... 이하 생략

-> 예시 글에서는 역사가 가장 큰 제목이기 때문에 <h1>를 사용했고, 개발과 최초 규격에는 그보다 아래 단계인 <h2>를 사용했습니다. 원문을 참고하여 적절하게 태그를 사용했습니다. (<https://ko.wikipedia.org/wiki/HTML> 중 역사 섹션) 제목 태그를 사용하면, 일반 텍스트보다 더 강조되는 스타일이 적용되는데 이는 브라우저가 기본적으로 제목 태그에 제공하는 스타일입니다. 제목이라는 의미에 맞게 기본적으로 좀 더 굵고 크게 표현이 됩니다.

[https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading\\_Elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements)

<https://www.w3.org/MarkUp/html3/headings.html>

- 단락 태그(paragraph)는 paragraph를 줄여서 p로 씁니다. 제목 태그에 이어 본문에 해당하는 내용을 단락 태그를 이용해 작성해보겠습니다.

<h1>역사</h1>

<h2>개발</h2>

<p>

1980년, 유럽 입자 물리 연구소(CERN)의 계약자였었던 물리학자 팀 버너스리가 HTML의 원형인 인콰이어를 제안하였다.

... 이하 생략

</p>

<h2>최초 규격</h2>

<p>

HTML 최초의 일반 공개 설명은 1991년 말에 버너스리가 처음으로 인터넷에서 문서를 "HTML 태그

"(HTML tag)로 부르면서 시작되었다.

... 이하 생략

</p>

-> 화면에는 별다른 변화는 없지만, 이전보다 훨씬 의미에 맞게 잘 짜인 마크업 구조라고 볼 수 있습니다.

- 개행 : <p>를 사용해서 단락으로 구분하면 자연스럽게 개행이 됩니다. HTML 문법 중 공백 시간에 배운 대로, html은 한 칸 이상의 공백 및 개행을 무시하기 때문에 실제 코드창에 개행을 하더라도 화면에 나타나지는 않습니다. 따라서 개행을 위해 쓰이는 태그가 바로 <br>입니다. (linebreak 를 줄여서 br 이라고 합니다.) <br>은 닫기 태그와 내용이 존재하지 않는 빈 태그입니다. 개행하고자 하는 곳에서 <br>을 선언하면 개행이 됩니다. 이처럼 <p>를 이용하면 태그 자체에서 자연스럽게 개행이 되지만, <p> 내부에서 강제로 개행을 하기 위해서는 <br>을 이용해야 합니다.

### 3) 텍스트를 꾸며주는 요소

- 텍스트 표현 태그 : 웹 표준화가 대두하면서 웹 문서의 구조와 표현을 분리하였습니다. 그 과정에서 많은 표현용 태그들이 사라졌고, 지금은 표현용 태그가 얼마 남지 않았습니다.

<b> : bold 태그는 글자를 굵게 표현하는 태그입니다.

<i> : italic 태그는 글자를 기울여서 표현하는 태그입니다.

<u> : underline 태그는 글자의 밑줄을 표현하는 태그입니다.

<s> : strike 태그는 글자의 중간선을 표현하는 태그입니다. (예전에 존재했던 strike 태그와는 다른 태그로, strike 태그는 폐기되어 더는 사용할 수 없습니다.)

-> 위 태그들은 의미가 없는 표현용 태그이기 때문에 사용하실 때는 각별히 주의를 하셔야 합니다.

살펴본 태그 이외에도 텍스트 관련 태그는 많습니다.

<p>

<b>Lorem</b> <i>ipsum</i> dolor sit amet<br>

<u>Lorem</u> <s>ipsum</s> dolor sit amet

</p>

[https://developer.mozilla.org/en-US/docs/Web/HTML/Element#inline\\_text\\_semantics](https://developer.mozilla.org/en-US/docs/Web/HTML/Element#inline_text_semantics)

### 4) 앵커 요소

- <a> : <a>(anchor 태그)는 a태그, 앵커, 링크 등 여러 이름으로 불립니다.

<a href="http://www.naver.com/" target="\_blank">네이버</a>

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

- href 속성 : 링크를 만들기 위해 <a>는 반드시 href(hypertext reference) 속성을 가지고 있어야 합니다. href 속성의 값은 링크의 목적지가 되는 URL입니다.

- target 속성 : target 속성은 링크된 리소스를 어디에 표시할지를 나타내는 속성입니다. 속성값으로는 \_self, \_blank, \_parent, \_top이 있습니다. \_self는 현재 화면에 표시한다는 의미로, target 속성이 선언되지 않으면 기본적으로 self와 같이 동작합니다. \_blank는 새로운 창에 표시한다는 의미로 외부 페이지가 나타나게끔 하는 속성입니다. \_parent와 \_top은 프레임이라는 특정 조건에서만 동작하는 속성으로, 최근에는 프레임을 잘 쓰지 않기 때문에 따로 다루지 않고 넘어가겠습니다.

- 기타 속성 : <a>에는 이 외에도 많은 속성이 있습니다. 아래 링크를 참고해 다루지 않은 나머지 속성을 직접 공부하기를 권장합니다.

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

(참고로 속성 이름 옆 html5 레이블로 표시된 것은 html5 버전이 나오면서 새로 생긴 속성이라는 뜻입니다. Obsolete attributes로 표시된 부분은 폐기된 속성들이므로 보지 않으셔도 됩니다.)

- 내부링크 : <a>를 통해 만들어진 링크가 꼭 외부 페이지로만 이동하는 것은 아닙니다. <a>를 통해 페이지 내부의 특정 요소로 초점을 이동할 수도 있는데, 이를 내부링크라고 합니다. 내부링크를 사용할 때는 href 속성값에 #을 쓰고 그 뒤에 페이지 내에서 이동하고자 하는 요소의 id 속성값을 적으면 됩니다.

```
<a href="#some-element-id">회사 소개로 이동하기</a>
```

... 중략.

```
<h1 id="some-element-id">회사 소개</h1>
```

보통 페이지에 내용이 많아 스크롤이 길어질 경우, 빠르게 화면 최상단으로 이동하고자 할 때 내부 링크를 주로 사용합니다. 웹페이지에서 화면 하단에 있는 'top' 또는 '맨 위로 이동하기' 버튼이 이에 해당합니다.

## 5) 의미가 없는 컨테이너 요소

- 의미가 없는 컨테이너 요소 : 태그 자체에 아무 의미가 없으며, 단순히 요소들을 묶기 위해 사용되는 태그입니다. 스타일을 주거나 서버에 보내는 데이터를 담기 위한 용도로 이런 의미 없는 요소들이 사용됩니다. 이런 의미 없는 태그의 사용빈도는 매우 높습니다. 그 이유는 html 태그들은 문서를 웹에 나타내기 위해 제작되어 기본적으로 문서에 최적화되어있는의미를 지니는데, 현재 웹의 형태는 문서 형태에서 많이 벗어났기 때문입니다. 다행히 HTML 버전이 업그레이드되면서 웹에 알맞은 태그들이 많이 생겼습니다. 가장 대표적으로 많이 쓰이는 의미가 없는 태그는 <div>, <span>입니다.

- <div>태그와 <span>태그 : div(division) 태그는 블록 레벨 태그입니다. 블록 레벨 요소는 기본적으로 한 줄을 생성해서 내용을 표현합니다. 반면, span 태그는 인라인 레벨 태그입니다. 인라인 레벨 요소들은 블록 레벨 요소의 한 줄 안에서 표현되는 요소들입니다. 이전에 배운 <p>는 블록 레벨 태그이며, 그 안에서 텍스트를 꾸며주는 <b>, <i>, <u>, <s>는 모두 인라인 레벨 태그입니다.

```
<div>
```

```
    <span>Lorem</span> ipsum dolor sit.
```

```
</div>
```

<div>, <span>는 모두 아무 의미가 없으므로 실제 브라우저도 별다른 스타일을 적용하지 않습니다.

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

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

## 6) 리스트 요소

- <ul> : ul(unordered list) 태그는 순서가 없는 리스트를 표현할 때 사용합니다.

```
<ul>
```

```
    <li>콩나물</li>
```

```
    <li>파</li>
```

```
    <li>국 간장</li>
```

```
    ...
```

```
</ul>
```

-> 콩나물국에 들어가는 일부 재료들을 나열한 리스트입니다. 각 재료는 나오는 순서가 바뀌어도 상관없으므로 이러한 것은 순서가 없는 리스트로 표현할 수 있습니다. <ul>을 선언한 후 그 안에서 <li>를 사용해 각 항목을 나타내서 사용합니다.

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

- <ol> : ol(ordered list) 태그는 순서가 있는 리스트를 표현할 때 사용합니다.

```
<ol>
```

```
<li>냄비에 국물용 멸치를 넣고 한소끔 끓여 멸치 육수를 7컵(1,400ml) 만든다.</li>
<li>콩나물을 넣고 뚜껑을 덮어 콩나물이 익을 때까지 끓인다.</li>
<li>뚜껑을 열고 대파, 마늘, 고춧가루를 넣고 끓인다.</li>
```

...

```
</ol>
```

-> 콩나물국을 끓이는 순서를 나열한 리스트입니다. 이 순서는 서로 바뀌면 안 되기 때문에 순서가 있는 <ol>을 사용해야 합니다. <ol>을 선언한 후 그 안에서 <li>를 사용해 각 항목을 나타내서 사용합니다.

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

- <dl> : dl(definition/description list) 태그는 용어와 그에 대한 정의를 표현할 때 사용합니다. <dl>은 앞서 배운 <ul>, <ol>과는 구조가 조금 다릅니다. <ul>, <ol>은 항목을 단순히 나열하는 구조지만, <dl>은 용어와 설명이 하나의 세트로 항목을 이루고 하나 이상의 항목으로 리스트가 이루어지는 구조입니다.

```
<dl>
```

```
<dt>리플리 증후군</dt>
```

```
<dd>허구의 세계를 진실이라 믿고 거짓된 말과 행동을 상습적으로 반복하는 반사회적 성격장애를 뜻하는 용어</dd>
```

```
<dt>피그말리온 효과</dt>
```

```
<dd>타인의 기대나 관심으로 인하여 능률이 오르거나 결과가 좋아지는 현상</dd>
```

```
<dt>언더독 효과</dt>
```

```
<dd>사람들이 약자라고 믿는 주체를 응원하게 되는 현상</dd>
```

```
</dl>
```

```
<dt> : 용어를 나타내는 태그
```

```
<dd> : 용어에 대한 정의 또는 설명을 나타내는 태그
```

용어 하나에 여러 정의가 들어갈 때, <dd>를 한 개 이상 쓰는 것이 가능합니다.

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

## 7) 이미지 요소

- <img> : <img>는 문서에 이미지를 삽입하는 태그로, 닫는 태그가 없는 빈 태그입니다.

```

```

- src 속성 : <img>의 필수 속성으로 이미지의 경로를 나타내는 속성입니다.

- alt 속성 : 이미지의 대체 텍스트를 나타내는 속성입니다. 대체 텍스트는 이미지를 대체하는 글을 뜻하며, 웹 접근성 측면에서 중요한 속성입니다. src 속성과 더불어 반드시 들어가야 하는 속성입니다.

- width/height 속성 : 이미지의 가로/세로 크기를 나타내는 속성입니다. 값의 단위는 필요하지 않으며, 값을 입력하면 자동으로 픽셀 단위로 계산됩니다. width/height는 선택적인 속성이지만 이미지의 크기가 고정적이라면 width/height 속성을 선언하는 것이 성능적인 측면에서 좋습니다. width/height 속성이 없으면 이미지는 원본 크기대로 노출되며, 둘 중 하나만 선언하면 나머지 한 속성은 선언한 속성의 크기에 맞춰 자동으로 비율에 맞게 변경됩니다. 반면 두 속성 모두 선언하면 이미지는 비율과 무관하게 선언한 크기대로 변경됩니다.

- 상대경로와 절대경로 : src 속성에는 이미지의 경로가 들어가며, 이미지의 상대경로와 절대경로가 있습니다. 상대경로는 현재 웹 페이지를 기준으로 상대적으로 이미지의 위치를 나타내는 경로이고, 절대경로는 실제 그 이미지가 위치한 곳의 전체 경로입니다.

```
<!-- 상대경로 -->
```

```

```

```
<!-- 절대경로 -->
```





-> 상대 경로에서의 './'는 페이지가 있는 현재 폴더를 나타냅니다.

- 이미지 파일 형식

gif : 제한적인 색을 사용하고 용량이 적으며 투명 이미지와 애니메이션 이미지를 지원하는 형식

jpg : 사진이나 일반적인 그림에 쓰이며 높은 압축률과 자연스러운 색상 표현을 지원하는 형식(투명을 지원하지 않는다.)

png : 이미지 손실이 적으며 투명과 반투명을 모두 지원하는 형식

## 8) 테이블 요소 1

- 표의 구성 요소 : 표는 셀(내용이 들어가는 하나의 칸)로 이루어져 있습니다. 표의 행(가로 방향)을 row, 열(세로 방향)을 column이라 합니다.

<table> : 표를 나타내는 태그

<tr> : 행을 나타내는 태그

<th> : 제목 셀을 나타내는 태그

<td> : 셀을 나타내는 태그

하나의 <table>은 하나 이상의 <tr>로 이루어져 있으며, <tr>은 셀을 나타내는 <th>, <td>를 자식으로 가지게 됩니다. 표를 구성할 때는 위에서 밑으로, 좌측에서 우측으로 작성하면 됩니다.

<table>

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

</tr>

<tr>

<td>5</td>

<td>6</td>

<td>7</td>

<td>8</td>

</tr>

<tr>

<td>9</td>

<td>10</td>

<td>11</td>

<td>12</td>

</tr>

<tr>

<td>13</td>

<td>14</td>

<td>15</td>

<td>16</td>

</tr>

</table>

-> 위는 1부터 16을 표현한 4행 4열의 표를 나타내는 코드입니다. 브라우저 화면을 확인해보면 테두리가 없어 표가 어색해 보일 수 있습니다. 브라우저에서 제공하는 테이블의 기본 스타일에는 테두리가 없기 때문입니다. 확인을 위해 아래의 CSS 코드를 <head> 안에 입력하면 테두리가 나타나는 걸 확인할 수 있습니다.

```
<style>
```

```
th, td { border: 1px solid; }
```

```
</style>
```

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

- 표의 구조와 관련된 태그 : 표가 복잡해지면 표를 해석해서 음성으로 전달해야 하는 스크린 리더기와 같은 보조 기기를 통해서는 표의 내용을 이해하는 것이 더욱 어려워질 것입니다. 따라서 표를 구조적으로 파악하기 위해 도움이 되는 태그를 사용해야 합니다.

<caption>: 표의 제목을 나타내는 태그

<thead>: 제목 행을 그룹화하는 태그

<tfoot>: 바닥 행을 그룹화하는 태그

<tbody>: 본문 행을 그룹화하는 태그

```
<table>
```

```
  <caption>Monthly Savings</caption>
```

```
  <thead>
```

```
    <tr>
```

```
      <th>Month</th>
```

```
      <th>Savings</th>
```

```
    </tr>
```

```
  </thead>
```

```
  <tbody>
```

```
    <tr>
```

```
      <td>January</td>
```

```
      <td>$100</td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>February</td>
```

```
      <td>$80</td>
```

```
    </tr>
```

```
  </tbody>
```

```
  <tfoot>
```

```
    <tr>
```

```
      <td>Sum</td>
```

```
      <td>$180</td>
```

```
    </tr>
```

```
  </tfoot>
```

```
</table>
```

-> <tfoot>은 <thead> 다음에 있지만, 실제 화면에서는 표의 맨 아래에 위치하게 된다는 점을 주의해야 합니다.

## 9) 테이블 요소 2

- 복잡한 표 만들기

```
<table>
  <caption>Specification values</caption>
  <thead>
    <tr>
      <th rowspan="2">Grade.</th>
      <th rowspan="2">Point.</th>
      <th colspan="2">Strength.</th>
      <th rowspan="2">Percent.</th>
    </tr>
    <tr>
      <th>kg/mm</th>
      <th>lb/in</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Hard</td>
      <td>0.45</td>
      <td>56.2</td>
      <td>80,000</td>
      <td>20</td>
    </tr>
    <tr>
      <td>Medium</td>
      <td>0.45</td>
      <td>49.2</td>
      <td>70,000</td>
      <td>25</td>
    </tr>
    <tr>
      <td>Soft</td>
      <td>0.45</td>
      <td>42.2</td>
      <td>60,000</td>
      <td>30</td>
    </tr>
  </tbody>
</table>
```

- <colgroup>, <col>, scope 속성, header 속성

## 10) 폼 요소 1

- 폼 요소는 서버에 데이터를 전달하기 위한 요소이며 <input>은 대표적인 폼 요소입니다. <input>은 내용이 없는 빈 요소이며 type 속성을 통해 여러 종류의 입력 양식으로 나타나게 됩니다. 본 강의에서는 자

주 사용되는 type들만 다루니 나머지는 아래 링크를 참고하시기 바랍니다.

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

- type="text"

<input type="text" placeholder="o o o">

-> 주로 아이디, 이름, 주소, 전화번호 등 단순한 텍스트를 입력할 때 사용합니다. type="text"에는 placeholder 속성이 존재합니다. placeholder 속성은 사용자가 입력하기 전 미리 화면에 노출하는 값으로, 입력하는 값의 양식을 표현할 수 있습니다.

- type="password"

<input type="password">

-> 암호와 같이 공개할 수 없는 내용을 입력할 때 사용합니다. 화면에는 type="text"와 같게 나타나지만 실제로 입력할 때 값을 노출하지 않습니다.

- type="radio"

<input type="radio" name="gender"> 남자

<input type="radio" name="gender"> 여자

-> 라디오 버튼을 만들 때 사용합니다. 라디오 버튼은 중복 선택이 불가능하며, 하나의 항목만을 선택해야 합니다.

- type="checkbox"

<input type="checkbox" name="hobby"> 등산

<input type="checkbox" name="hobby"> 독서

<input type="checkbox" name="hobby"> 운동

-> 체크박스를 만들 때 사용합니다. 체크박스는 중복 선택이 가능합니다.

- 라디오 버튼과 체크박스에는 checked, name 속성이 존재합니다. checked 속성은 값이 별도로 존재하지 않는 boolean 속성입니다. boolean 속성은 true/false 둘 중 하나의 값을 가지며 속성이 존재하면 true, 속성이 존재하지 않으면 false를 가집니다. name 속성은 라디오 버튼과 체크박스를 그룹화시켜주는 속성입니다. 아직 name 속성을 다루기 전이므로 실제 버튼을 클릭했을 경우 동작하는 방식에 대해서만 알아두시기 바랍니다.

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

## 11) 폼 요소 2

- type="file"

<input type="file">

-> 파일을 서버에 올릴 때 사용합니다. 브라우저에 따라 표현되는 형태는 조금씩 다르지만, 모두 같은 역할을 합니다.

- type="submit|reset|image|button"

<form action="/test.html">

메시지: <input type="text" name="message"><br>

<input type="submit">

<input type="reset">

<input type="image" src="http://placeholder.it/50x50?text=click" alt="click" width="50" height="50">

<input type="button" value="버튼">

</form>

-> submit, reset, image, button 타입은 모두 클릭 가능한 버튼을 만듭니다.

submit : form의 값을 전송하는 버튼



reset : form의 값을 초기화하는 버튼  
 image : 이미지를 삽입할 수 있는 버튼 (submit과 동작이 동일함)  
 button : 아무 기능이 없는 버튼  
 이미지 버튼은 이미지 관련 속성인 src, alt 속성이 반드시 필요하며 width/height 속성을 적용할 수도 있습니다.

## 12) 폼 요소 3

- <select> : <select>는 선택 목록 상자 또는 콤보박스라고도 합니다. 몇 개의 선택지를 리스트 형태로 노출하고 그중 하나를 선택할 수 있게 하는 태그입니다. (multiple 속성을 사용하면 다중 선택도 가능합니다.)

```
<select>
  <option>서울</option>
  <option>경기</option>
  <option>강원</option>
  ...
```

```
</select>
```

-> <select>내부의 <option>으로 각 항목을 나타냅니다. <option>의 속성으로는 selected가 있으며 이는 선택된 항목을 의미합니다.

- <textarea> : 한 줄만을 입력할 수 있는 <input type="text">와 달리 여러 줄의 텍스트를 입력할 때 사용합니다.

```
<textarea rows="5" cols="30">
```

```
...
```

```
</textarea>
```

<textarea>에는 텍스트 상자의 크기를 조절하는 rows, cols 속성이 있습니다.

cols : 가로 크기를 조절하는 속성(한 줄에 들어가는 글자의 수, 수치의 의미는 평균적인 글자의 너비로 정확히 글자 수를 나타내지는 않습니다.)

rows : 세로 크기를 조절하는 속성(화면에 보여지는 줄 수)

- <button> : 버튼을 만들 때 사용하며 submit, reset, button 3가지의 타입이 있습니다.

```
<button type="submit|reset|button">○○○</button>
```

-> 각 버튼은 이전에 배웠던 input 타입의 submit, reset, button과 모두 같은 기능을 가진 버튼입니다.

다만, 빈 태그가 아니며 내용을 안에 직접 넣을 수 있으므로 좀 더 자유로운 스타일 표현이 가능합니다.

## 13) 폼 요소 4

- <label> : <label>은 form 요소의 이름과 form 요소를 명시적으로 연결시켜주기 위해 사용합니다. 모든 form 요소에 사용할 수 있습니다.

```
<label for="name">이름</label>: <input type="text" id="name"><br>
```

```
<label for="nickname">이름</label>: <input type="text" id="nickname"><br>
```

```
<label for="address">이름</label>: <input type="text" id="address"><br>
```

-> form 요소의 id 속성값과 <label>의 for 속성값을 같게 적어주어야 합니다. <label>을 사용하면 이를 클릭했을 경우 해당 form 요소를 클릭한 것처럼 동작합니다. 또한, 스크린 리더기를 통해 듣게 되면 해당 form 요소에 접근 시 <label>을 함께 읽어주게 됩니다. <label>은 사용성, 접근성적인 측면으로 중요한 역할을 하므로 반드시 써주는 것이 좋습니다.

- <fieldset>, <legend> : <fieldset>, <legend>는 form 요소를 구조화 하기 위해 필요한 태그입니다.

<fieldset> : 여러 개의 폼 요소를 그룹화하여 구조적으로 만들기 위해 사용

<legend> : 폼 요소의 제목으로 <fieldset> 내부에 작성

<fieldset>은 보통 form의 성격에 따라 구분합니다. <legend>는 <fieldset>의 자식으로 반드시 최상단에 위치해야 합니다.

<fieldset>

    <legend>기본 정보</legend>

    ... 폼 요소들 ...

</fieldset>

<fieldset>

    <legend>부가 정보</legend>

    ... 폼 요소들 ...

</fieldset>

- <form> : <form>은 form 요소들을 감싸는 태그로 데이터를 묶어서 실제 서버로 전송해주는 역할을 하는 태그입니다. 만약 <fieldset>으로 구조화되어있다면 <fieldset>도 함께 감싸는 역할을 합니다.

<form action="" method="">

    <fieldset>

        <legend>기본 정보</legend>

        ... 폼 요소들 ...

    </fieldset>

    <fieldset>

        <legend>부가 정보</legend>

        ... 폼 요소들 ...

    </fieldset>

</form>

action: 데이터를 처리하기 위한 서버의 주소

method: 데이터를 전송하는 방식을 지정

method 속성값에는 get/post 2가지 방식이 존재합니다. get 방식은 데이터가 전송될 때 주소창에 파라미터 형태로 붙어 데이터가 노출됩니다. 반면, post 방식은 데이터가 전송될 때 데이터가 노출되지 않습니다.

- 코드 실습

<!DOCTYPE html>

<html lang="ko">

<head>

    <meta charset="UTF-8">

    <title>form</title>

</head>

<body>

    <form action="">

        <h1>Form 관련 요소</h1>

        <fieldset>

            <legend>기본 정보</legend>

            <label for="userid">아이디 : </label> <input type="text" placeholder="영문으로만 써주세요" id="userid"><br>

            <label for="userpw">비밀번호 : </label> <input type="password"

```

id="userpw"><br>
        성별 : <label for="male">남자</label> <input type="radio" name="gender"
id="male" checked>, <label for="female">여자</label> <input type="radio" name="gender"
id="female"><br>
        </fieldset>
        <fieldset>
            <legend>부가 정보</legend>
            취미 : 영화 감상 <input type="checkbox" name="hobby" checked>, 음악 감
상 <input type="checkbox" name="hobby">, 독서 <input type="checkbox" name="hobby"><br>
            프로필 사진 업로드 : <input type="file"><br>
            사는 지역 : <select>
                <option>서울</option>
                <option>경기</option>
                <option>강원</option>
                <option selected>제주</option>
            </select><br>
            자기소개 : <textarea cols="30" rows="5" placeholder="자기소개는 짧게 해주
세요."></textarea><br>
            <button type="submit">전송</button>
            <button type="reset">취소</button>
        </fieldset>
    </form>
</body>
</html>

```

### 3. 콘텐츠 모델, 시멘틱 마크업, 블록&인라인

#### 1) 콘텐츠 모델

- Content Models의 7 분류

1. Metadata Content
2. Flow Content
3. Sectioning Content
4. Heading Content
5. Phrasing Content
6. Embedded Content
7. Interactive Content

- Metadata : " base, link, meta, noscript, script, style, title " Metadata에는 콘텐츠의 스타일, 동작을 설정하거나 다른 문서와의 관계 등 정보를 포함하는 요소들이 포함됩니다. 메타 태그, 타이틀 태그, 스타일 태그, 링크 태그가 이에 해당하며 대부분 <head>내에 들어간다는 것이 특징입니다.

- Flow : " a, abbr, address, map>area, article, aside, audio, b, bdo, blockquote, br, button, canvas, cite, code, datalist, del, details, dfn, div, dl, em, embed, fieldset, figure, footer, form, h1 ~ h6, header, hgroup, hr, i, iframe, img, input, ins, kbd, keygen, label, map, mark, math, menu, meter, nav, noscript, object, ol, output, p, pre, progress, q, ruby, samp, script, section, select, small, span, strong, style[scoped], sub, sup, svg, table, textarea, time, ul, var, video, wbr " Flow에는 문서의 자연스러운 흐름에 의해 배치되는 요소들이 포함됩니다. Metadata에 해당하는 일부 태그들만 Flow에서 제외되며 요소 대부분이 Flow에 포함됩니다.

- Sectioning : " article, aside, nav, section " Sectioning에는 문서의 구조와 관련된 요소들이 포함됩니다. HTML5에서 새로 생긴 <article>, <aside>, <nav>, <section> 등이 포함되며 이 태그들은 문서의 구조, 아웃라인에 영향을 주게 됩니다.

- Heading : " h1, h2, h3, h4, h5, h6 " Heading에는 각 section의 header를 정의하는 heading 태그가 포함됩니다.

- Phrasing : "a, abbr, map>area, audio, b, bdo, br, button, canvas, cite, code, datalist, del, dfn, em, embed, i, iframe, img, input, ins, kbd, keygen, label, map, mark, math, meter, noscript, object, output, progress, q, ruby, samp, script, select, small, span, strong, sub, sup, svg, textarea, time, var, video, wbr" Phrasing에는 문서의 텍스트 또는 텍스트를 꾸며주는 문단 내부 레벨로 사용되는 요소들이 포함됩니다.

- Embedded : " audio, canvas, embed, iframe, img, math, object, svg, video " Embedded에는 외부 콘텐츠를 표현하는 요소들이 포함되며 오디오나 비디오, 이미지 등 멀티미디어 관련 요소들이 이에 해당합니다.

- Interactive : " a, audio[controls], button, details, embed, iframe, img[usemap], input, keygen, label, menu, object[usemap], select, textarea, video[controls] " Interactive에는 사용자와 상호작용을 하는 요소들이 포함되며 대표적으로 form 요소들이 이에 해당합니다.

[https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content\\_categories](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories)

#### 2) 시멘틱 마크업

- 시멘틱 마크업이란? 시멘틱 마크업은 종종 POSH(Plain Old Semantic HTML)라고도 불리는데, 단어 그대로 평범하고 오래된 의미론적인 HTML이라는 뜻입니다. 시멘틱은 즉, 기계(컴퓨터, 브라우저)가 잘 이해할 수 있도록 하는 것을 뜻합니다. 애초에 프로그래밍 언어는 사람과 기계와의 정해진 약속이며 HTML 역시 마찬가지입니다. 시멘틱 마크업은 적절한 HTML 요소를 올바르게 사용하는 것에서 시작합니다.

- 시멘틱 마크업 하기 : 마크업 할 때는 의미에 맞는 태그, 요소를 사용하는 것이고 문서를 표현할 때는

구조화를 잘 해주는 것입니다. 정해진 약속대로 코드를 작성하게 되면 결국 기계뿐 아니라 사람도 이해하기 쉬운 코드가 됩니다.

<b>굵은</b> vs <strong>중요한</strong>

<i>기울어진</i> vs <em>강조하는</em>

<u>밑줄친</u> vs <ins>새롭게 추가된</ins>

<s>중간선이 있는</s> vs <del>삭제된</del>

<b>는 의미 없이 단순히 텍스트를 굵게 표현하는 태그지만, <strong>은 중요하다는 의미를 지닙니다. <strong>은 중요하다는 의미에 맞춰 브라우저에 의해 굵은 스타일로 표현된 것입니다. 따라서 중요하다는 의미를 포함할 때는 <b>가 아닌 <strong>을 사용하는 것이 적절하고 시멘틱한 마크업입니다. 이외에 <i>는 단순히 기울어진 글자를 표현하지만, <em>은 글자의 특정 부분을 강조하는 의미를 지닙니다. <u>와 <s>는 단순히 글자에 선을 그은 것이고, <ins>와 <del>은 내용이 새롭게 추가되거나 삭제가 되었다는 의미를 지닙니다.

<https://developer.mozilla.org/en-US/docs/Glossary/Semantics>

[https://www.w3schools.com/html/html5\\_semantic\\_elements.asp](https://www.w3schools.com/html/html5_semantic_elements.asp)

<https://nuli.navercorp.com/seminar/s03th>

3) HTML5 시멘틱 요소 : <article> <aside> <figcaption> <figure> <footer> <header> <main> <mark> <nav> <section> <time>

<https://developer.mozilla.org/en-US/docs/Glossary/Semantics>

#### 4) 블록&인라인

- 블록 레벨 요소 : 부모 요소의 가로 영역에 맞게 꽉 채워져 표현되는 요소입니다. 양옆으로 다른 요소가 배치되지 않게 박스를 생성하므로 박스의 위아래로 줄 바꿈이 생기게 됩니다. 블록 레벨 요소는 일반적인 모든 요소(블록, 인라인 레벨 등)를 포함할 수 있습니다. " div, h1~h6, p, ul, li, table ..."

[https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level\\_elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements)

- 인라인 레벨 요소 : 하나의 라인 안에서 자신의 내용만큼의 박스를 만드는 요소입니다. 라인의 흐름을 끊지 않고 요소 앞 뒤로도 줄 바꿈이 되지 않아 다른 인라인 요소들이 자리할 수 있습니다. 인라인 레벨 요소는 블록 레벨 요소의 자식으로 분류되기 때문에 자손으로 블록 레벨 요소를 가질 수 없습니다. 즉, 인라인 레벨 요소는 블록 레벨 요소를 포함할 수 없습니다. " span, i, img, em, strong, a ..." 다만, HTML5 버전에서 생겨난 한 가지 예외 경우가 있는데 <a>는 인라인 레벨 요소지만 자손으로 블록 레벨 요소를 가질 수 있습니다.

[https://developer.mozilla.org/en-US/docs/Web/HTML/Inline\\_elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements)