

1.1 Introducing WEB

WEB

웹(WEB)이란, 인터넷에 연결된 컴퓨터들을 통해 사람들이 정보를 공유할 수 있는 전 세계적인 정보 공간이다. 웹을 통해 공유되는 정보들은 대개 웹 페이지(web page)의 형태로 공유되며, 웹 페이지는 특수한 양식을 갖춘 텍스트로 구성된다. 각 웹페이지는 일반적인 텍스트(plain text)뿐만 아니라 이미지, 동영상, 다른 웹페이지로 연결되는 하이퍼링크(hyperlink) 등의 웹 자원(web resource)으로 다양하게 구성되고, 하나의 주제, 하나의 영역을 공유하는 여러 웹 자원과 웹 페이지는 웹 사이트(website)를 구성한다. 웹 사이트는 웹 서버(web server)라는 디바이스에서 동작하는 웹 애플리케이션(web application)이라고 하는 프로그램에 의해 구현되어 작동하며, 이 웹 애플리케이션으로 인해 권한만 충분하다면 전 세계 어디서든 웹 사이트에 접속할 수 있다.

본 교재에서 다루는 웹의 영역은 웹 애플리케이션과 관련된 영역이며, 흔히 웹 개발자가 다루는 웹의 영역도 이와 상당수 일치한다. 앞으로 여러분은 웹 애플리케이션의 작동 원리와 구조를 이해하고, HTML과 CSS, Javascript를 이용하여 웹을 디자인하고 설계하는 학습을 할 것이다.

Front-end and Back-end

웹 서버, 혹은 웹 사이트가 작동하는 원리를 생각해보자. 예를 들어, 사용자가 블로그 형태의 웹 사이트에 접속했을 때 사용자에게 보여지는 영역에는 어떤 것이 있는가? 블로그의 블로거, 블로거에 대한 상세 정보, 게시물, 게시물에 대한 상세 정보, 댓글, 댓글에 대한 상세 정보 등이 있을 것이다. 뿐만 아니라 블로그를 예쁘게 꾸민 디자인 등도 사용자에게 보여지는 영역이다. 이렇듯 사용자에게 보여지는 영역을 **front-end**라고 부르며, 사용자(client) 쪽에서 동작하는 영역이라고 하여 **client-side**라고도 부른다.

반대로, 사용자에게 직접적으로 보여지지 않는 영역도 있다. 사용자의 요청에 맞게 적절한 웹 페이지를 구성하여 전달해주는 로직, 게시물이나 댓글, 블로거의 정보를 데이터베이스에 저장하고 데이터베이스로부터 읽어오는 기능, 회원에 따라 글을 작성할 권한을 부여할 것인지, 댓글을 수정할 권한을 부여할 것인지 결정하는 로직 등이 있을 것이다. 이렇게 사용자에게 직접적으로 보여지지 않는 영역을 **back-end**라고 부르고, 이러한 로직은 서버상에서 동작하기 때문에 **server-side**라고도 부른다.

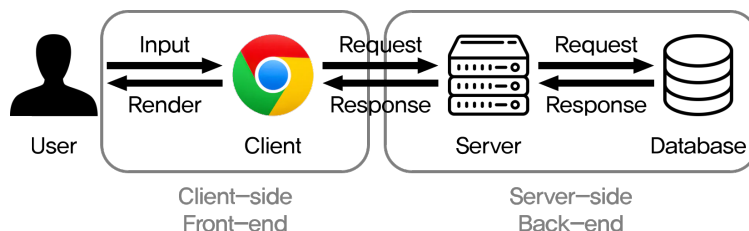


Figure 1.1 How Typical Web Application Works

본 교재에서는 front-end 분야를 다루며, front-end에서 기본적으로 사용되는 HTML, CSS, Javascript를 학습한다. 이를 응용하여 실무에서는 React.js, Vue.js, Angular.js 등의 프레임워크를 사용하기도 한다.

Chapter 2

HTML: The Basic Structure

Contents

2.1	Introducing HTML	8
2.2	Basic Structure of HTML	9
2.3	Commonly Used HTML Tags	11
2.4	Class and Id Attributes	17
2.5	HTML Exercises	19

2.1 Introducing HTML

HTML은 HyperText Markup Language의 약자로, 웹 브라우저에 웹 페이지의 구조를 체계적으로 표현하는 컴퓨터 언어이다. HTML은 웹 페이지의 가장 기본적인 뼈대를 이루며, 어떤 front-end framework를 사용하든 HTML을 필수적으로 이해하고 있어야 한다.

HTML is not a Programming Language?

소프트웨어 개발과 관련된 밈(meme)을 접하다 보면 가장 자주 듣는 밈 중 하나가 바로 “HTML is not a programming language”, 즉 HTML은 프로그래밍 언어가 아니라는 밈이다. HTML은 프로그래밍 언어가 아닌, 구조를 서술하기 위한 언어로, 쉽게 접할 수 있는 프로그래밍 언어인 C, Java, Python 등과 달리 조건문, 반복문, 변수 선언 등의 기능이 전혀 없다. 구조를 서술하기 위해 존재하는 언어이니만큼, 미리 정해진 규칙에 따라 HTML을 작성하여 구조를 표현한다는 개념으로 접근하면 HTML이라는 언어를 쉽게 이해할 수 있을 것이다.

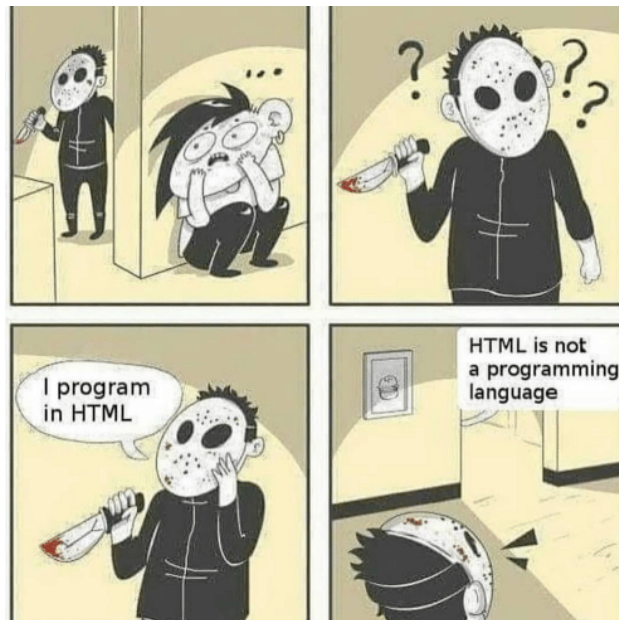


Figure 2.1 HTML is not a programming language

2.2 Basic Structure of HTML

Tags and Elements

HTML은 HyperText Markup Language, 즉 마크업 언어이며, 마크업 언어는 구조를 서술할 때 정해진 마크로 시작하여 마크로 끝나는 언어이다.

Code 2.1 Example of HTML

```
<!doctype html>
<html>
<head>
  <title>Example Domain</title>
  <meta charset="utf-8">
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
  <div>
    <h1>Example Domain</h1>
  </div>
</body>
</html>
```

Code 2.1을 관찰하여 HTML의 구조를 이해해보자. Example Domain이라는 내용을 `<title>`과 `</title>`이 감싸고 있는데, 이러한 문자열을 태그(tag)라고 한다. 두 태그에서 `title`을 **태그 이름(tag name)**이라고 하며, 태그 내부의 부분(Example Domain)을 **내용(content)**이라고 한다. 유사하게, 이 `title` 태그를 `<head>`와 `</head>`가 감싸고 있다. 이렇게 태그와 그 태그 사이에 감싸져 있는 내용이 하나의 **HTML 요소(element)**를 구성하며, HTML 요소의 앞쪽에 위치하여 시작을 알리는 태그를 시작 태그, 뒤쪽에 위치하여 끝을 알리는 태그를 끝 태그라고 한다. 이처럼 HTML은 기본적으로 내용의 양 끝단을 태그로 감쌈으로써 구조를 표현한다.¹

Code 2.1을 더 관찰해보면 `head` 태그의 요소와 `body` 태그의 요소가 `html` 태그의 요소 일부분인 것을 확인할 수 있다. 즉, HTML 요소는 내부에 또 다른 HTML 요소를 포함할 수 있고, 다른 HTML 요소의 일부가 될 수 있으며, 이와 같은 구조를 **nested 구조**라고 한다. HTML에서는 요소 A와 요소 B가 요소 C의 일부분을 구성할 때, 요소 A와 B를 요소 C의 **하위 요소(sub element)** 혹은 **자식 요소(child element)**라고 하고, 요소 C를 요소 A와 요소 B의 **상위 요소(super element)** 혹은 **부모 요소(parent element)**라고 한다. 또한, 요소 A를 요소 B의 **형제 요소(sibling element)**라고 한다.

또한, `meta` 태그에서는 태그 내에 태그 이름 외의 다른 문자열을 확인할 수 있다. 이는 HTML 요소의 속성과 관련된 문자열로, 첫 번째 `meta` 태그는 `charset` 속성의 값이 `utf-8`이며, 두 번째 `meta` 태그는 `http-equiv`의 값이 `Content-type`, `content`의 값이 `text/html; charset=utf-8`이라는 뜻이다.² 이렇듯 HTML의 속성은

¹ 다만, `meta` 태그와 같이 시작 태그만 존재하는 요소도 존재하며, 이러한 종류의 태그를 단일 태그라고 한다. 단일 태그는 그 자체만으로 하나의 요소가 된다.

² 이렇게 데이터의 정의를 이름과 데이터의 값으로 이루어진 데이터 집합을 **키/값 쌍(key-value pair)**라고 하며, 데이터의 정의를 키(key), 데이터의 값을 값(value)라고 한다. 예를 들어, 어떠한 학생의 인적 정보를 저장한 데이터 집합이 있고, 그 내부에는 학생의 이름이 홍길동, 학번이 2021990999, 성별이 남성, 거주지는 서울특별시 성북구 등의 정보가 포함되어 있다고 가정하자. 그러면 학생의 “이름”, “학번”, “성별”, “거주지” 등은 key가 되고, “홍길동”, “2021990999”, “남성”, “서울특별시 성북구” 등은 각 key에 대한 value가 된다. 웹 분야 뿐만 아니라 컴퓨터과학 분야 전반에서 사용되는 대부분의 데이터는 이러한 key-value pair 형태로 표현될 수 있다.

key-value pair 형태의 데이터 집합이며, 속성(attribute)은 key, 속성값은 value에 대응된다. 속성값은 기본적으로 큰따옴표(") 내에 작성되며, 속성과 속성값은 등호(=)로 연결되며, 각 key-value pair는 공백을 사이에 두고 나열된다. HTML 요소의 속성은 태그나 내용 못지않게 중요한 역할을 하는데, 2.3절에서 더 자세히 다룬다.

이제까지 살펴본 HTML 요소의 구조를 종합해보면 **Figure 2.2**와 같이 표현될 수 있다.

<tagname attr1="value1" attr="value2">content</tagname>

Figure 2.2 Structure of HTML Element

HTML의 기본적인 구조

HTML 문서의 구조에는 가장 기본적인 틀이 정해져 있다. 먼저 `doctype`을 이용하여 문서가 HTML 문서임을 명시하고, 가장 상위 태그로 `html` 태그, 그 아래에 `head`와 `body` 태그가 위치해야 한다. 코드로 나타낸다면 **Code 2.2**와 같다.³

Code 2.2 Basic Structure of HTML

```
<!doctype html>
<html>
  <head>
    <!-- Head Element Content -->
  </head>

  <body>
    <!-- Body Element Content -->
  </body>
</html>
```

`head` 태그는 문서에 대한 전반적인 정보를 담고 있는 태그로, 이 태그에는 `title`, `meta`, `link` 등 다양한 태그들이 사용된다. 이번 절에서는 `title`과 `meta` 두 태그만 살펴보자. 먼저, `title` 태그 내부의 텍스트는 웹 페이지의 제목으로, 웹 브라우저를 통해 웹 페이지에 접속했을 때 브라우저의 상단에 표시된다. `meta` 태그는 HTML 문서의 인코딩 방법, `viewport`, 키워드 등 중요한 정보들을 지정할 수 있는 태그이다.

`body` 태그는 웹 페이지에서 사용자에게 보여질 부분이 포함되는 태그이다. 개발자가 웹 페이지를 통해 사용자에게 보여주고자 하는 텍스트, 이미지 등의 내용은 `body` 태그 안에 작성되어야 하며, 2.3절에서 본격적으로 웹 페이지의 내용을 작성하는 방법을 다룬다.

³HTML에서는 주석을 `<!--, -->`로 표현한다.

2.3 Commonly Used HTML Tags

이 절에서는 소개되는 태그들은 HTML에서 매우 자주 사용되는 태그들이다. 예제를 따라서 직접 작성해보고 웹 브라우저에서 열어 결과를 확인해보자.

Heading (h1 - h6) Tags

h1 - h6 태그는 heading의 약자로, 제목을 나타낼 때 사용되는 태그이다. h 뒤의 숫자가 작을수록 화면에 표시되는 글자의 크기가 크다.

Code 2.3 h1 - h6 Tags

```
<h1>This is h1 tag</h1>
<h2>This is h2 tag</h2>
<h3>This is h3 tag</h3>
<h4>This is h4 tag</h4>
<h5>This is h5 tag</h5>
<h6>This is h6 tag</h6>
```

p, br Tag

p 태그는 paragraph의 약자로, 문단을 구분해주는 태그이다. 이 태그는 문단과 문단 사이에 공백을 넣어주는 역할을 하며, 문단과 문단의 사이에는 별도의 공간이 생긴다.

HTML에서는 공백(), 줄 바꿈(\n) 등의 whitespace 문자들은 종류와 관계없이 하나의 공백 취급을 받고, 이러한 whitespace 문자들은 여러 개 연속적으로 나열되어도 하나의 공백으로 취급된다. 따라서, 코드 내에서 줄 바꿈을 하여 작성해도 실제로는 줄 바꿈이 되지 않으며, 줄 바꿈을 하기 위해서는 br 태그를 줄 바꿈하고자 하는 위치에 넣어야 한다.

Code 2.4 p, br Tag

```
<p>
  HTML stands for HyperText Markup Language.<br>
  It is a language used to express the structure of webpage.
</p>
<p>
  Languages used to implement client-side are HTML, CSS and Javascript.<br>
  HTML and CSS are not programming languages, but Javascript is. <br>
</p>
```

a Tag

a 태그는 HTML 요소에 다른 웹 페이지로 이동할 수 있는 하이퍼링크를 걸 수 있는 태그이다. 이 태그는 속성 값에 유의하여 작성하여야 하며, 다음 세 속성은 a 태그에서 가장 중요한 속성들이다.

- **href**: HTML 요소와 연결될 리소스의 주소
- **title**: HTML 요소와 연결될 리소스에 대한 설명. 호버(마우스를 위에 갖다 대는 것)를 했을 때 표시됨.
- **target**: 문서가 로드될 대상을 지정하는 옵션
 - **_blank**: 새로운 창이나 탭
 - **_self**: 현재 창
 - **_parent, _top**: 잘 쓰이지 않는 옵션

Code 2.5를 직접 실행해보고, 위의 속성값을 참고하여 a 태그를 활용해보자.

Code 2.5 a Tag

```
<a href="https://www.google.com" title="Google Homepage" target="_self">Google</a>
<br>
<a href="https://www.naver.com" title="Naver" target="_blank">Naver</a>
```

button Tag

button 태그는 클릭할 수 있는 버튼을 생성하는 태그이다. 이 태그를 눌렀다가 놓았을 때 특정한 동작이 수행될 수 있으며, input 태그를 이용해서도 유사한 기능을 구현할 수는 있으나 디자인적 관점에서 button 태그가 더 편하므로 더 많이 사용한다.

button 태그에 중요한 속성으로는 disabled, type 등이 있다. disabled 속성을 설정하면 버튼은 비활성화되어 클릭할 수 없는 상태가 되며, type은 버튼을 눌렀을 때 수행되는 동작의 성격에 차이가 생긴다. 버튼을 눌렀을 때 동작이 수행되도록 하는 방법은 5장에서 다룬다.

Code 2.6 button Tag

```
<button type="submit">Learn front-end</button>
<button disabled>Learn back-end</button>
```

List Tags

리스트, 즉 항목들을 나열하여 표현할 수 있는 태그들이 세 가지 있다. 순서가 없는 리스트(unordered list)를 나타내는 ul 태그, 순서가 있는 리스트(ordered list)를 나타내는 ol 태그가 있으며, 두 종류의 리스트 모두 리스트의 각 원소를 li 태그를 이용하여 나열한다.

ul 태그나 ol 태그를 이용하여 리스트를 표현하면 각 원소의 앞에 bullet과 같은 문자나 숫자가 자동으로 생긴다. 이러한 표식은 CSS를 활용하여 제거하거나 변경할 수 있다. Code 2.7을 직접 실행해보고, 다양한 형태의 리스트를 직접 만들어보자.

Code 2.7 List Tags

```
<h3>Unordered List</h3>
<ul>
  <li>Visual Studio Code</li>
  <li>Atom</li>
```

```

    <li>Sublime Text</li>
    <li>Vim</li>
</ul>

<h3>Ordered List</h3>
<ol>
    <li>Visual Studio Code</li>
    <li>Atom</li>
    <li>Sublime Text</li>
    <li>Vim</li>
</ol>

```

img Tag

img 태그는 HTML 문서에 이미지를 삽입할 수 있는 단일 태그이다. 다음은 img 태그에 사용되는 몇 가지 속성이다.

- **src:** 삽입하고자 하는 이미지의 주소
- **alt:** 이미지를 가져오는 데 실패하였을 때 대신 표시되는 이미지에 대한 설명
- **width:** 이미지를 띄우고자 하는 너비 (px)
- **height:** 이미지를 띄우고자 하는 높이 (px)

Code 2.8 img Tag^a

```




```

^a이 코드에서는 HTML의 속성-속성값 쌍을 분리하기 위해 1칸의 공백 대신 개행 문자와 5칸의 공백을 사용하였다. HTML에서는 연속된 whitespace 문자들을 하나의 공백으로 취급하기 때문에 이렇게 표시할 수 있으며, 속성이 많거나, 속성값이 길거나, 내용이 길거나 등의 이유로 HTML 요소가 지나치게 길어지면 이와 같이 개행하여 작성하는 것이 좋다.

Font Tags

글자의 폰트를 변경할 수 있는 태그들이다. 간단한 HTML 문서를 작성할 때는 쓰일 수 있으나, CSS를 활용하여 작성하는 것이 권장되기 때문에 자주 쓰이지는 않는다.

Code 2.9 Font Tags

```

<b>Bold Text</b><br>
<strong>Important Text</strong><br>
<i>Italic Text</i><br>
<em>Emphasized Text</em><br>
<mark>Marked Text</mark><br>
<small>Small Text</small><br>
<del>Deleted Text</del><br>
<ins>Inserted Text</ins><br>
<sub>Subscript</sub> Text<br>
<sup>Superscript</sup> Text<br>

```


Input Tags

입력 태그들은 사용자로부터 정보를 입력받을 수 있는 태그들로, `input`, `textarea`, `select` 등이 있다. `input` 태그는 `type` 속성의 값에 따라 다양한 형태의 입력을 받을 수 있다. 다음은 `input` 태그의 `type` 속성의 값에 따라 입력받을 수 있는 정보의 형태 중 일부를 나타낸 것이다.

- `text`: 일반 텍스트 (plain text)
- `password`: 비밀번호
- `radio`: 선택 목록 중 하나만 선택할 수 있음
- `checkbox`: 선택 목록 중 여러 개 선택할 수 있음

`textarea` 태그는 `input` 태그와는 달리 여러 줄의 텍스트를 입력받을 수 있으며, 입력받는 부분의 크기나 설명 등은 속성을 이용하여 설정할 수 있다. `select`, `option` 태그는 여러 선택지 중 하나를 선택할 수 있는 drop-down 리스트를 만드는 태그이다.

Code 2.10 Input Tags

```
<h3>Input username: </h3>
<input type="text" name="username">

<h3>Input password: </h3>
<input type="password" name="password">

<h3>Gender: </h3>
<label><input type="radio" name="gender" value="male">Male</label>
<label><input type="radio" name="gender" value="female">Female</label>

<h3>Your Major: </h3>
<select name="major">
  <option value="cs">Computer Science</option>
  <option value="phy">Physics</option>
  <option value="chm">Chemistry</option>
  <option value="math">Mathematics</option>
</select>

<h3>Introduce yourself: </h3>
<textarea cols="40" rows="5"
  placeholder="Introduce yourself"
  name="introduction">
</textarea>
```

Code 2.10을 살펴보면, 모든 입력 태그에는 `name` 값이 지정되어 있고, 선택지가 있는 입력 태그에는 `value` 값이 지정되어 있다. `name` 속성의 값과 `value` 속성의 값은 key-value pair를 이루어, `name`의 값과 `value`의 값은 각각 입력받은 항목의 이름과 값을 뜻한다. 다만 텍스트를 입력받는 태그들의 경우 입력란에 입력한 값이 `value` 값이므로 `value` 값이 필요하지 않는데, 이 값을 직접 지정해줄 경우 기본값이 된다. `name`과 `value`의 값은 입력받은 데이터를 key-value pair의 형태로 전송할 때 사용되기 때문에 매우 중요한 속성이다.

HTML Entities

HTML에서는 특정 문자들이 구조를 표현할 목적으로 사용되기 때문에, 이러한 문자들을 실제로 웹 페이지 상에 표시하고자 할 때 문제가 발생할 수 있다. **Code 2.11**과 같이 작성된 HTML 문서는 어떤 문제가 발생할지 예상해보자.

Code 2.11 Improper Way of Displaying < and >

This is how inequality <signs> look like.

Code 2.11에서는 “This is how inequality <signs> look like.”이라는 텍스트를 표시하려고 했던 의도와는 달리 <signs>라는 텍스트가 태그로 취급되어 “This is how inequality look like.”이라는 텍스트가 표시된다. 이는 <, > 두 문자열이 태그를 나타내기 위한 문자이기 때문에 발생하는 현상이다. 그렇다면 이러한 문자들은 표시하기 위해서는 어떻게 작성해야 하는가? 또한, ※과 →와 같은 키보드 상에는 존재하지 않는 문자는 어떻게 작성하는가?

이러한 특수문자는 HTML 개체(entity)를 사용하여 나타낸다. HTML 개체는 개체의 이름을 사용하여 `<entity-name>;`의 형태로 나타내거나, 개체의 번호를 사용하여 `&#<entity-number>;`의 형태로 나타낸다. 예를 들어, `<`, `>`의 개체 이름은 각각 `lt`, `gt`이므로 **Code 2.11**은 **Code 2.12**와 같이 나타내어질 수 있다.

Code 2.12 Proper Way of Displaying < and >

This is how inequality <signs> look like.

이외에도 “(“), &(&) 등의 개체도 존재한다. 특히 는 non-breaking space의 약자로, 공백을 뜻한다. 앞에서 br 태그의 필요성을 설명하면서 whitespace 문자는 여러 번 연속되어 나열되어도 하나의 공백으로 취급된다고 하였다. 그러므로 여러 칸을 띄워 표시할 때 단순히 space를 여러 번 입력하여 나타내면 안되고, 라는 HTML 개체를 이용하여 연속된 공백을 나타내야 한다. **Code 2.13**을 확인해보자.

Code 2.13 Usage of

`<h3>`This is how you use `</h3>`

`<h3>`This is how you use ` </h3>`

이 외에도 별도의 개체 이름이 없는 특수문자의 경우 유니코드 번호⁴를 이용하여 삽입할 수 있다. 예를 들어, reference mark, 혹은 흔히 당구장 문자(※)로 불리는 문자의 유니코드 번호는 8251 이므로, **Code 2.14**와 같이 작성될 수 있다.

Code 2.14 Proper Way of Displaying < and >

※ This is how inequality <signs> look like.</h3>

⁴유니코드 번호는 <https://www.fileformat.info/info/unicode/char/search.htm> 페이지에서 검색하여 찾을 수 있다.

div and span Tag

HTML 코드의 가독성을 좋게 하고, CSS와 JS를 이용하여 웹 페이지를 디자인하고 추가 기능을 부여하기 위해서는 HTML을 구조화할 필요가 있다. 지금까지 살펴본 태그들은 모두 특정한 기능을 가진 태그들이는데, 구조화할 때 사용되는 태그가 앞의 태그들처럼 특정한 기능을 갖는다면, HTML 코드의 구조가 엉망진창이 될 수 밖에 없다. 이러한 이유로 존재하는, 특정한 기능이 없고, HTML 요소들을 묶어서 레이아웃(layout)을 구성하기 위해 존재하는 태그를 non-semantic 태그라고 하며, `div`와 `span` 태그가 있다.

`div` 태그는 인접한 요소와 같은 줄에 있으려고 하지 않고, `span` 태그는 인접한 요소와 같은 줄에 있으려고 한다는 차이가 있다. **Code 2.15**를 통해 이러한 특징을 확인해보자.

Code 2.15 div and span Tags

```
<div>
  <h3>&lt;div&gt; tag</h3>
  <div>
    <textarea></textarea>
  </div>
  <div>
    <textarea></textarea>
  </div>
</div>

<div>
  <h3>&lt;span&gt; tag</h3>
  <span>
    <textarea></textarea>
  </span>
  <span>
    <textarea></textarea>
  </span>
</div>
```

Code 2.15를 조금만 관찰하면, 비슷한 기능이나 역할을 하는 요소들이 묶여있어 전체적인 구조를 파악하기 용이하다. 이렇게 `div`와 `span` 태그를 사용하는 것만으로도 그렇지 않았을 때보다 코드가 훨씬 구조화되고, 가독성이 대폭 좋아진 것을 확인할 수 있다.

`div`와 `span`은 대표적인 non-semantic 태그이지만, 개발자의 편의에 따라 예약되지 않은 태그 이름 사용하여 HTML 문서를 작성할 수 있다. 유사하게, 태그 속성 역시 예약되지 않은 속성의 이름과 속성값을 사용하여 HTML 문서를 작성하고, 이를 CSS와 JS에서 사용할 수 있다.

2.4 Class and Id Attributes

2.3절에서 HTML 문서를 작성하는 기본적인 방법에 대하여 학습하였다. 3장부터는 CSS와 JS를 배우면서 HTML에 적용하는 과정을 다루는데, CSS에서는 각 요소에 원하는 디자인을 적용할 수 있고, JS에서는 각 요소를 추가 및 삭제하거나, 그 속성을 수정하는 등의 작업을 할 수 있다. 이러한 CSS와 JS를 HTML 문서에 적용할 때, 특정 요소 혹은 특정 분류의 모든 요소에 CSS나 JS를 적용하게 된다.

2.3절에서 HTML 문서를 구조화하기 위해 `div`나 `span` 태그의 쓰임새에 대해 학습하였다. 그러나 태그만으로는 기능이나 역할 등 개발자가 원하는 분류 기준에 따라 HTML 요소들을 분류하는 것은 어려우며, HTML 문서가 매우 길어진다면 단순히 태그 이름만으로 요소들을 구분하는 것은 불가능하다.

이렇게 특정 기준에 따라 요소들을 분류하거나, 특정 HTML 요소를 지정할 때 필요한 속성이 **class**와 **id**이다. Class와 id는 모든 HTML 요소에 적용할 수 있으며, CSS, JS를 HTML과 연동할 때 매우 중요한 역할을 한다.

Class Attribute

먼저, class 속성은 HTML 요소들을 특정한 기준에 따라 분류(classify)할 때 사용되는 속성이며, class 속성의 값을 class name이라고 한다. 특정한 기준으로 분류하였을 때 하나의 묶음으로 묶이는 요소들에는 각각 같은 이름의 class를 사용한다. 하나의 HTML 요소는 여러 class를 가질 수 있고, 각 class name은 공백을 이용하여 구분한다. **Code 2.16**은 class 속성을 활용한 예제이다.

Code 2.16 Class Attributes Example

```
<div class="page-thumbnail new">
  
  <span class="page-title">
    <a href="/study/201R/3">Week 2 Handout</a>
  </span>
</div>
<div class="page-thumbnail">
  
  <span class="page-title">
    <a href="/study/201R/2">Week 1 Assignment</a>
  </span>
</div>
<div class="page-thumbnail">
  
  <span class="page-title">
    <a href="/study/201R/1">Week 1 Assignment</a>
  </span>
</div>
```

Id Attribute

Id 속성은 특정한 HTML 요소 하나를 식별(identification)하기 위해 사용되는 속성이다. 하나의 요소는 여러 id를 가질 수 없고, 특정 id의 값을 갖는 HTML 요소가 여러 개가 될 수 없다.⁵ 다만, 각 HTML 요소는 class와 id를 동시에 가질 수 있다. **Code 2.17**은 id 속성을 활용한 예제이다.

Code 2.17 Id Attributes Example

```
<div id="article-form">
  <input id="article-title" name="title">
  <textarea id="article-content" name="content"></textarea>
  <button>Submit</button>
</div>
```

Naming Convention

Class name이나 id를 작성할 때 반드시 준수해야 하는 작명 규칙(naming convention)은 없다. 그러나 협업이나 유지보수 등 생산성의 향상을 위해 널리 통용되고 권장되는 규칙을 소개한다.⁶

- 대문자의 사용은 지양하고, 소문자로만 구성한다. 숫자로 시작하지 않는다.
- 이름은 class나 id의 의미에 잘 부합하여 어떠한 기준으로 지어진 이름인지 알기 쉽게 작명한다.
- 여러 단어의 조합은 하이픈(-)으로 연결하여 작명한다. (예: multiple-words)

⁵이 규칙을 위반하더라도 HTML 문서는 정상적으로 렌더링된다.

⁶아래 소개되는 작명 규칙보다 더 자세한 규칙은 다음 링크를 참조하길 바란다: <https://bogmong.tistory.com/14>