



KWEB, Korea University

Web Application Essentials

Frontend Section

Written By: Lee, Chanyoung

April 1, 2024

Table of Contents

0	Preface	5
0.1	Introduction to WEB	6
1	HTML: The Basic Structure	7
1.1	Basic Structure of HTML	8
1.2	Commonly Used HTML Tags	12
1.3	Class and ID Attributes	19
1.4	HTML Exercises	21
2	CSS: Desining HTML	23
2.1	Introducing CSS	24
2.2	Basic Structure of CSS	25
2.3	Style Properties	28
2.4	Selectors	29
2.5	Layouts	32
2.6	Responsive Web	39
2.7	CSS Exercises	41
3	Basics of JavaScript	43
3.1	Introducing JavaScript	44
3.2	Declaration of Variables	45
3.3	Data Types	48

3.4	Statements and Functions	52
3.5	Built-in Objects	58
3.6	Basics of JS Exercises	65
4	JavaScript: Dynamic Frontend	67
4.1	JavaScript with Front-end	68
4.2	Document Object Model (DOM)	69
4.3	Browser Object Model (BOM)	75
4.4	Event and Event Listener	78
4.5	JS Exercises	81
A	Source Codes	83
A.1	CSS: Designing HTML Source Codes	84
A.2	JavaScript: Dynamic Frontend Source Codes	87
B	Exercise Answers	93
B.1	HTML Exercise Answers	94
B.2	CSS Exercise Answers	95
B.3	Basics of JS Exercise Answers	99
B.4	JS Exercise Answers	100

Chapter 0

Preface

Contents

0.1	Introduction to WEB	6
-----	-------------------------------	---

0.1 Introduction to 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 웹 페이지는 일반적인 텍스트(plain text)뿐만 아니라 이미지, 동영상, 다른 웹 페이지로 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 프로그램의 형태로 구현되고 작동하며, 웹 애플리케이션을 통해 특수한 제한이 없다면 전 세계 어디서든 웹 사이트에 접속할 수 있다.

본 교재에서 다루는 웹의 영역은 흔히 웹 개발자가 다루는 웹의 영역과 밀접하게 관련이 있는 웹 애플리케이션과 관련된 영역이다. 앞으로 여러분은 이 교재를 통해 웹 애플리케이션의 작동 원리와 구조를 이해하고, 웹을 디자인하고 설계하는 학습을 할 것이다.

Frontend and Backend

우리가 일상에서 사용하는 웹 사이트, 웹 애플리케이션이 어떻게 작동할 지 상상해보자. 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 **frontend**라고 부르며, 사용자(client) 쪽에서 동작하는 영역이라고 하여 **client-side**라고도 한다.

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

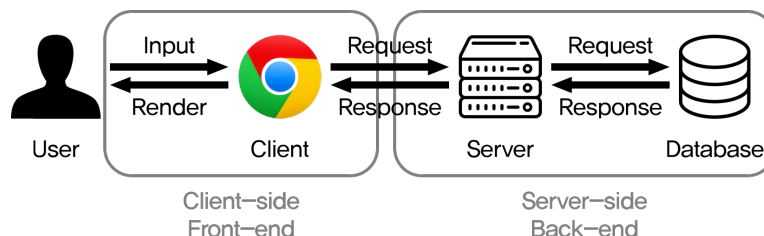


Figure 1 How typical web application works

본 교재에서는 frontend에서 기본적으로 사용되는 HTML, CSS, JavaScript를 학습한다.

Chapter 1

HTML: The Basic Structure

Contents

1.1	Basic Structure of HTML	8
1.2	Commonly Used HTML Tags	12
1.3	Class and ID Attributes	19
1.4	HTML Exercises	21

1.1 Basic Structure of HTML

What is HTML?

일상생활에서 웹 서핑을 하며 방문하는 여러 웹 사이트를 생각해 보자. 어느 웹 사이트를 방문하건 가장 공통적으로 많이 보는 창 중 하나는 로그인 창일 것이다.

(1) 로그인

(2) 아이디

(3) 아이디

(4) 비밀번호

(5) 비밀번호

(6) 로그인

아직 계정이 없으신가요? [회원가입](#)
아이디가 기억나지 않으시나요? [아이디 찾기](#)
비밀번호가 기억나지 않으시나요? [비밀번호 찾기](#)

Figure 1.1 Sign in page of KWEB homepage

Figure 1.1은 KWEB 홈페이지의 로그인 창으로, 매우 간단한 형태의 로그인 창이다. 이 로그인 창은 어떻게 구성되어 있는가? 먼저 (1) 가장 위에 큰 글씨로 “로그인”이라고 적혀 있고, (3, 5) 그 밑에 아이디와 비밀번호를 입력하기 위한 공간이 마련되어 있다. 그리고 (2, 4) 그 공간의 좌상단에 아이디 또는 비밀번호라는 텍스트가 적혀 있어 그 공간에 어떤 값을 입력해야 하는지 명시한다. 마지막으로 (6) 입력한 아이디와 비밀번호로 로그인을 하기 위해 눌러야 하는 버튼이 있다.

이렇듯 웹 페이지는 텍스트, 입력란, 버튼 등의 여러 요소로 구성되어 있고, 어떤 요소들이 어떻게 웹 페이지를 구성하는지에 대한 정보가 웹 개발자에 의해 잘 명시되고 작성되어야 한다. HTML은 이러한 웹 페이지의 구조를 체계적으로 표현하기 위한 컴퓨터 언어로, 가장 기본적인 뼈대를 표현하며 웹 브라우저가 웹 페이지를 화면에 표시하는 과정인 렌더링(rendering)은 이러한 HTML 문서를 기반으로 이루어진다.

HTML과 관련된 여러 농담에서 보듯, HTML은 프로그래밍 언어가 아니며 구조를 서술하기 위한 언어이다. 그렇기 때문에 C, Java, Python, JavaScript 등의 여러 프로그래밍 언어에 존재하는 조건문, 반복문, 변수 선언 등과 관련된 문법이 전혀 존재하지 않는다. 그러므로 HTML 문서를 작성할 때에는 머릿속에 미리 생각해둔 웹 페이지의 구조를 미리 정해진 규칙에 따라 표현한다고 생각하면 HTML에 비교적 쉽게 다가갈 수 있을 것이다.

Tags and Elements

HTML은 HyperText Markup Language의 약자로, 구조를 서술할 때 정해진 마크로 시작하여 마크로 끝나는 언어인 마크업 언어의 일종이다. **Code 1.1**은 웹 페이지 <https://www.example.com/>의 HTML 문서의 일부를 변형한 것으로, 이를 관찰하며 HTML의 구조를 이해해보자.

Code 1.1 Example of HTML

```
1  <!doctype html>
2  <html>
3  <head>
4      <title>Example Domain</title>
5      <meta charset="utf-8">
6      <meta http-equiv="Content-type" content="text/html; charset=utf-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1">
8  </head>
9  <body>
10     <div>
11         <h1>Example Domain</h1>
12         <p>
13             This domain is for use in illustrative examples in documents.
14             You may use this domain in literature without prior coordination
15             or asking for permission.
16         </p>
17         <p><a href="https://www.iana.org/domains/example">More information...</a></p>
18         <!-- Comment is written like this in HTML -->
19     </div>
20 </body>
21 </html>
```

Line 4를 관찰해보면, “Example Domain”이라는 텍스트를 `<title>`과 `</title>`이 감싸고 있다. 이때 `<title>`과 `</title>`을 **태그(tag)**라고 하며, title을 **태그 이름(tag name)**이라고 한다. 두 태그 중 앞쪽에 위치하는 태그를 시작 태그 또는 여는 태그라고 하고 뒤쪽에 위치하는 태그를 끝 태그 또는 닫는 태그라고 하며, 두 태그의 이름은 반드시 같아야 한다. 한 쌍의 태그에 둘러싸인 부분을 **내용(content)**이라고 하며, 두 태그와 그 사이에 위치한 내용이 하나의 HTML **요소(element)**를 구성한다. 요소는 HTML에서 구조를 표현하는 가장 기본 단위이며, 요소 간의 관계를 통해 구조가 표현된다.

정리하자면, Line 4에서 각 용어에 대응하는 부분은 다음과 같다.

- 태그 이름: title
- 시작 태그: `<title>`
- 끝 태그: `</title>`
- 태그: `<title>`, `</title>`
- 내용: Example Domain
- 요소: `<title>Example Domain</title>`

다만, Line 5-7에서 보듯 내용이 없고 시작 태그 그 자체로만 이루어진 요소도 존재한다. 이렇게 내용이 필요하지 않은 태그는 시작 태그로만 요소를 형성할 수 있고, 이를 단일 태그라고 하며, 단일 태그에는 `br`, `hr`, `img`, `input`,

meta 등이 존재한다. 과거에는 단일 태그에도 태그의 끝을 알리기 위해 `<meta charset="utf-8" />`와 같이 별도의 슬래시(/)를 삽입하기도 하였으나, 최근에는 생략하는 것이 권장된다.

Line 3-8에서는 하나의 title 요소와 세 개의 meta 요소가 head 태그의 내용을 구성하는 것을 볼 수 있다. 이렇게 HTML 요소는 내부에 다른 HTML 요소를 포함할 수 있고, 다른 HTML 요소의 내용 부분에 위치함으로써 다른 HTML 요소에 포함될 수 있다. 예를 들어, 요소 A와 요소 B가 요소 C에 포함되어 있을 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. 이러한 포함 관계는 중첩(nested)될 수 있어, 모든 요소는 하나의 tree 구조를 이룬다.

즉, Line 3-8에서 head 요소는 title 요소와 세 meta 요소의 부모 요소이며, title 요소와 세 meta 요소는 모두 head 요소의 자식 요소이다. 또한, title 요소와 세 meta 요소는 서로 형제 요소이다.

Line 5-7의 meta 요소에서는 태그 내에 태그 이름 외의 다른 텍스트가 존재하는데, 이는 각 HTML 요소의 세부적인 성질을 표현한다. 예를 들어, Line 5의 meta 요소는 charset의 값이 utf-8이고, Line 6의 meta 요소는 http-equiv의 값이 Content-type, content의 값이 text/html; charset=utf-8이라는 뜻이다.¹² 이렇듯 HTML의 속성은 기본적으로 key-value pair 형태의 데이터 집합이며, 이때 key에 해당하는 부분을 속성(attribute), value에 해당하는 부분을 속성값이라고 한다. 속성값은 기본적으로 큰따옴표(") 내에 작성되며, 속성과 속성값은 등호(=)로 연결되고, 각 key-value pair는 공백을 사이에 두고 나열된다.

또한, 주석은 Line 15와 같이 `<!--`와 `-->` 사이에 작성하며, 여러 줄의 코드를 주석 처리할 수 있다.

이제까지 살펴본 HTML 요소의 구조를 종합해보면 Figure 1.2와 같이 표현될 수 있다.

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

Figure 1.2 Structure of HTML element

HTML에서는 예약된(reserved) 태그 이름과 속성, 그리고 속성값이 존재하며, 이들 중 자주 사용되는 것들을 1.2절에서 다룰 예정이다. 예약되지 않은 태그 이름이나 속성을 임의로 만들어서 작성할 수 있으나, 이러한 태그 이름이나 속성에 기능을 직접 추가해주지 않는 이상 이러한 태그 이름이나 속성은 어떠한 기능도 하지 않는다.

Basic Structure of HTML

HTML 문서의 구조에는 가장 기본적인 틀이 정해져 있다. 먼저 doctype을 이용하여 문서가 HTML 문서임을 명시하고, 가장 상위 태그로 html 태그, 그 아래에 head와 body 태그가 위치해야 한다. 코드로 나타낸다면 Code 1.2와 같다.

¹이렇게 데이터의 정의나 이름과 데이터의 값으로 이루어진 데이터의 집합을 키/값 쌍(key-value pair)이라고 하며, 데이터의 정의를 키(key), 데이터의 값을 값(value)라고 한다.

²예를 들어, 어떠한 학생의 이름이 홍길동, 학번이 2024990999, 성별이 남성, 거주지는 서울특별시라고 가정해보자. 이 학생의 인적 정보를 저장한 데이터 집합에서 “이름”, “학번”, “성별”, “거주지” 등은 key가 되고, “홍길동”, “2024990999”, “남성”, “서울특별시 성북구” 등은 각 key에 대한 value가 된다. 대부분의 데이터는 이러한 key-value pair 형태로 표현될 수 있다.

Code 1.2 Basic structure of HTML

```
1 <!doctype html>
2 <html>
3   <head>
4     <!-- Head Element Content -->
5   </head>
6
7   <body>
8     <!-- Body Element Content -->
9   </body>
10 </html>
```

head 태그는 문서에 대한 전반적인 정보를 담고 있는 태그로, 이 태그에는 title, meta, link 등 다양한 태그들이 사용된다. 이번 절에서는 title과 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 보여지는 부분을 담고 있는 태그로, 텍스트, 이미지 등의 내용은 body 태그 안에 작성되어야 한다. 1.2절에서는 본격적으로 웹 페이지의 내용을 작성하는 방법과 필요한 문법을 다룬다.

1.2 Commonly Used HTML Tags

이 절에서는 HTML에서 매우 자주 사용되는 태그들을 소개한다. example.html과 같이 파일을 생성하고, 예제를 따라서 직접 작성한 뒤 해당 파일을 웹 브라우저에서 열어 결과를 확인 및 비교해보자.

Heading (h1 - h6) Tags

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

Code 1.3 h1 - h6 tags

```
1 <h1>This is h1 tag</h1>
2 <h2>This is h2 tag</h2>
3 <h3>This is h3 tag</h3>
4 <h4>This is h4 tag</h4>
5 <h5>This is h5 tag</h5>
6 <h6>This is h6 tag</h6>
```

p, br Tags

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

HTML에서는 스페이스(), 줄 바꿈(\n) 등의 공백(whitespace) 문자들은 종류와 관계없이 하나의 공백 취급을 받고, 이러한 공백 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.³

Code 1.4 p, br tags

```
1 <p>
2     HTML stands for HyperText Markup Language.<br>
3     It is a language used to express the structure of webpage.
4 </p>
5 <p>
6     Languages used to implement client-side are HTML, CSS and JavaScript.<br>
7     HTML and CSS are not programming languages, but JavaScript is.<br>
8 </p>
```

³스페이스 문자를 여러 개 넣기 위해서는 17쪽의 를 참고하여라.

a Tag

a 태그는 HTML 요소에 다른 웹 페이지로 이동할 수 있는 하이퍼링크를 걸 수 있는 태그이다. 이 태그는 속성값에 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 옵션이므로 생략

Code 1.5를 직접 실행해보고, 위의 속성값을 참고하여 a 태그를 활용해보자.

Code 1.5 a tag

```
1 <a href="https://www.google.com/" title="Google Homepage" target="_self">Google</a>
2 <br>
3 <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 수행⁴될 수 있으며, input 태그를 이용해서도 유사한 기능을 구현할 수는 있으나 요소를 디자인할 때 button 태그가 더 편하므로 더 많이 사용된다.

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.

Code 1.6 button tag

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

List Tags

여러 항목을 나열하여 표현할 수 있는 리스트와 관련된 태그는 세 가지가 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. 리스트를 표현하기 위해서는 **Code 1.7**과 같이 먼저 ul 태그나 ol 태그를 이용하여 리스트를 생성한 후, 그 안에 li 태그를 이용하여 나열하고자 하는 원소들을 나열한다.

⁴4장에서 다룸

ul 태그를 사용하면 각 원소 앞에 bullet이 생기고 ol 태그를 사용하면 순서대로 숫자가 표시되는데, 이러한 표식은 CSS를 제거하거나 변경할 수 있다. **Code 1.7**을 직접 실행해보고, 다양한 형태의 리스트를 직접 만들어보자.

Code 1.7 List tags

```
1 <h3>Unordered List</h3>
2 <ul>
3     <li>Visual Studio Code</li>
4     <li>Atom</li>
5     <li>Sublime Text</li>
6     <li>Vim</li>
7 </ul>
8
9 <h3>Ordered List</h3>
10 <ol>
11     <li>Visual Studio Code</li>
12     <li>Atom</li>
13     <li>Sublime Text</li>
14     <li>Vim</li>
15 </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 1.8은 인터넷상에 존재하는 네이버와 다음의 로고 이미지를 HTML 문서에 넣은 것이다. 네이버 로고는 원래 너비가 206px 인 이미지이나, width 속성의 값이 640이므로 640px로 변경되었다. 다음 로고 이미지 주소는 실제 존재하지 않으므로 이미지를 가져오는 데에 실패하여 alt 속성의 값인 “Daum Logo”가 표시된다.

Code 1.8 img tag for online image⁵

```
1 
4 
```

Code 1.9는 인터넷상의 이미지가 아닌 로컬 컴퓨터에 위치한 이미지를 HTML 문서에 넣은 것이다. 다음 링크의 이미지를 HTML 파일이 위치한 디렉토리에 kweb-logo.png로 이름을 바꾸어 저장하여야. 그러면 이 이미지는 HTML 파일과 같은 디렉토리에 위치하므로 이미지의 주소를 ./kweb-logo.png와 같이 작성할 수 있다.

⁵이 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.

- <https://i.imgur.com/hSSFVSw.png>

Code 1.9 img tag for local image

```
1 
```

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 1.10 Font tags

```
1 <b>Bold Text</b><br>
2 <strong>Important Text</strong><br>
3 <i>Italic Text</i><br>
4 <em>Emphasized Text</em><br>
5 <mark>Marked Text</mark><br>
6 <small>Small Text</small><br>
7 <del>Deleted Text</del><br>
8 <ins>Inserted Text</ins><br>
9 <sub>Subscript</sub> Text<br>
10 <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 - text와 같이 텍스트를 입력받으나 입력한 내용이 표시되지 않음
- 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 1.11 Input tags

```
1 <h3>Input username: </h3>
2 <input type="text" name="username">
3
4 <h3>Input password: </h3>
```

```

5 <input type="password" name="password">
6
7 <h3>Gender: </h3>
8 <label><input type="radio" name="gender" value="male">Male</label>
9 <label><input type="radio" name="gender" value="female">Female</label>
10 <label><input type="radio" name="gender" value="other">Other</label>
11
12 <h3>Your Major: </h3>
13 <select name="major">
14     <option value="cs">Computer Science</option>
15     <option value="phy">Physics</option>
16     <option value="chm">Chemistry</option>
17     <option value="math">Mathematics</option>
18 </select>
19
20 <h3>Introduce yourself: </h3>
21 <textarea cols="40" rows="5"
22     placeholder="Introduce yourself"
23     name="introduction">
24 </textarea>

```

Code 1.11을 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, name의 값과 value의 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. name과 value의 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 1.12**와 같이 작성된 HTML 문서는 어떤 문제가 발생할지 예상해보자.

Code 1.12 Improper way of displaying < and >

```

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

```

Code 1.12에서는 “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 <signs>가 HTML 요소로 취급되어 발생하는 현상이다. 그렇다면 이러한 문자들은 표시하기 위해서는 어떻게 작성해야 하는가? 또한, ※와 →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 1.12**는 **Code 1.13**과 같이 고쳐질 수 있다.

Code 1.13 Proper way of displaying < and >

```
1 <h3>This is how inequality &lt;signs&gt; look like.</h3>
```

이외에도 "("), &(&) 등의 개체도 존재한다. 특히 는 non-breaking space의 약자로, 공백을 뜻한다. 앞에서 br 태그의 필요성을 설명하면서 공백 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 1.14**를 확인해보자.

Code 1.14 Usage of

```
1 <h3>This is how you use      &nbsp;</h3>
2 <h3>This is how you use &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</h3>
```

이 외에도 별도의 개체 이름이 없는 특수문자의 경우 유니코드 번호⁶를 이용하여 삽입할 수 있다. 예를 들어, reference mark, 혹은 흔히 당구장 문자(※)로 불리는 문자의 유니코드 번호는 8251이므로, **Code 1.15**와 같이 작성될 수 있다.

Code 1.15 Using unicode number to display reference mark

```
1 <h3>&#8251; This is how inequality &lt;signs&gt; look like.</h3>
```

div and span Tags

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 태그들인데, 코드를 구조화할 때 사용되는 태그는 특정한 기능이 있어서는 안 된다. 이러한 이유로 존재하는, 아무 기능이 없고, HTML 요소들을 묶어서 레이아웃(layout)을 구성하기 위해 존재하는 태그를 non-semantic 태그라고 하며, div와 span 태그가 있다.

div 태그는 인접한 요소와 같은 줄에 있으려고 하지 않고, span 태그는 인접한 요소와 같은 줄에 있으려고 한다는 차이가 있다. **Code 1.16**을 통해 이러한 특징을 확인해보자.

Code 1.16 div and span tags

```
1 <div>
2   <h3>&lt;div&gt; tag</h3>
3   <div>
4     <textarea></textarea>
```

⁶유니코드 번호는 <https://www.fileformat.info/info/unicode/char/search.htm> 페이지에서 검색하여 찾을 수 있다.

```
5     </div>
6     <div>
7         <textarea></textarea>
8     </div>
9 </div>
10
11 <div>
12     <h3>&lt;span&gt; tag</h3>
13     <span>
14         <textarea></textarea>
15     </span>
16     <span>
17         <textarea></textarea>
18     </span>
19 </div>
```

Code 1.16을 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 좋아지는 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.

1.3 Class and ID Attributes

1.2절에서 HTML 문서를 작성하는 기본적인 방법에 대하여 학습하였다. 2장부터는 CSS와 JS를 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.

1.2절에서 HTML 문서를 구조화하기 위해 div나 span 태그의 쓰임새에 대해 다루었다. 그러나 태그만으로는 기능이나 역할 등 개발자가 원하는 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 1.17은 class 속성을 활용한 예제이다.

Code 1.17 Example of class attribute

```
1 <div class="page-thumbnail new">
2   
3   <span class="page-title">
4     <a href="/study/frontend/3">Week 2 Handout</a>
5   </span>
6 </div>
7 <div class="page-thumbnail">
8   
9   <span class="page-title">
10    <a href="/study/frontend/2">Week 1 Assignment</a>
11  </span>
12 </div>
13 <div class="page-thumbnail">
14   
15   <span class="page-title">
16    <a href="/study/frontend/1">Week 1 Assignment</a>
17  </span>
18 </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.⁷ 다만, 각 HTML 요소는 class와 ID를 동시에 가질 수 있다. Code 1.18은 ID 속성을 활용한 예제이다.

Code 1.18 Example of ID attribute

```
1 <div id="article-form">
2     <input id="article-title" name="title">
3     <textarea id="article-content" name="content"></textarea>
4     <button>Submit</button>
5 </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.⁸

- 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)

⁷이 규칙을 위반하더라도 HTML 문서는 정상적으로 렌더링된다.

⁸아래 소개되는 작명 규칙보다 더 자세한 규칙은 다음 링크를 참조하길 바란다: <https://bogmong.tistory.com/14>

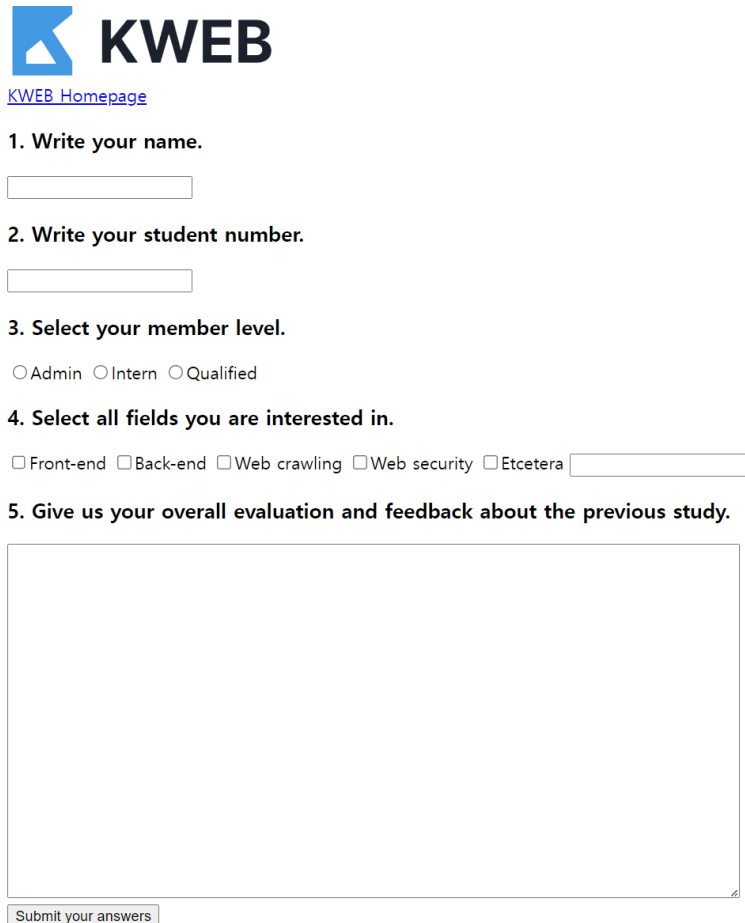
1.4 HTML Exercises

Exercise 1.1: Simple Survey Page

KWEB 동아리 설문 조사에 쓰일 HTML 문서를 **Figure 1.3**과 같이 작성하여라. KWEB 로고 이미지와 홈페이지 주소는 아래와 같으며, HTML의 표준 구조를 준수하고, 각종 태그를 사용해서 구현한다. 제시되지 않았더라도 각 요소에 적당한 속성과 속성값을 적절히 부여하고, div, span 태그와 class, id 등을 적절히 활용하여 HTML 문서를 구조화하고, 가독성을 높여라.

- KWEB 로고: <https://i.imgur.com/hSSFVSw.png>
- KWEB 홈페이지: <https://kwebofficial.com/>

KWEB Survey Page



KWEB
[KWEB Homepage](#)

1. Write your name.

2. Write your student number.

3. Select your member level.
☐ Admin ☐ Intern ☐ Qualified

4. Select all fields you are interested in.
☐ Front-end ☐ Back-end ☐ Web crawling ☐ Web security ☐ Etcetera

5. Give us your overall evaluation and feedback about the previous study.

Figure 1.3 KWEB survey page example

Exercise 1.2: Structurizing HTML Code

1.1절에서 학습한 HTML의 기본 구조, 1.2절에서 학습한 div, span 태그와 1.3절에서 학습한 class, id를 이용하여, 앞의 **Code 1.11**을 HTML 표준에 맞게 수정하고, 자유롭게 구조화해보자. (정해진 정답은 없다)

