

웹 개발 스터디

1주차: 웹의 기본 개념과 동작 원리 소개: 관련 기술들(HTML, CSS, JavaScript)



HTML: basic

- 계층적인 태그로 구성 되어있다

```
<tagname attribute-name="attribute-value" ...>  
    <!-- 콘텐츠(자식 요소들...) -->  
</tagname>
```

- 각 태그는 하나 이상의 <어트리뷰트-값> 쌍을 가질 수 있다

```

```

- 최상위 태그 `<html>` 바로 아래에 다음 요소가 존재한다

- `<head>`: 파일의 메타데이터에 대해 기술한다
- `<body>`: 표현하고자 하는 콘텐츠를 기술한다

HTML: basic

- ▶ 요소는 형태에 따라 아래 두가지로 분류해 볼 수 있음
- ▶ 1. 요소(Element)



- ▶ 2. 빈 요소(Empty Element)

```
<input type="text" name="name" placeholder="name">  
<meta charset="utf-8">
```

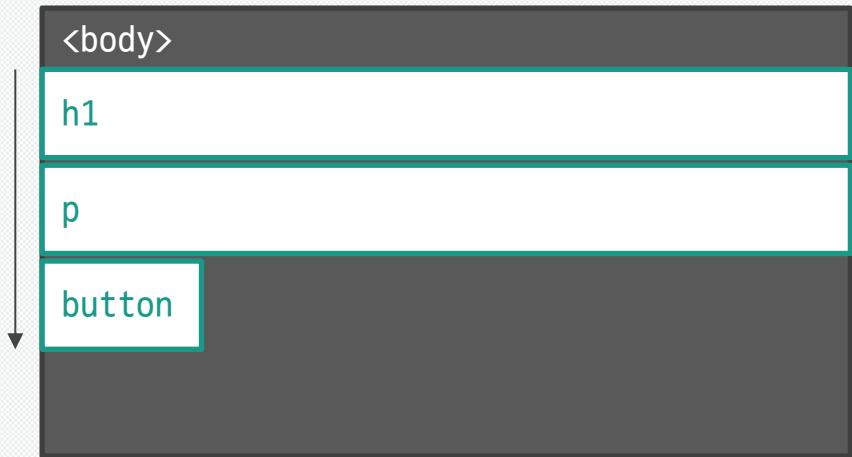
- img, input, link와 같이 종료태그가 없는 요소

HTML: 예제

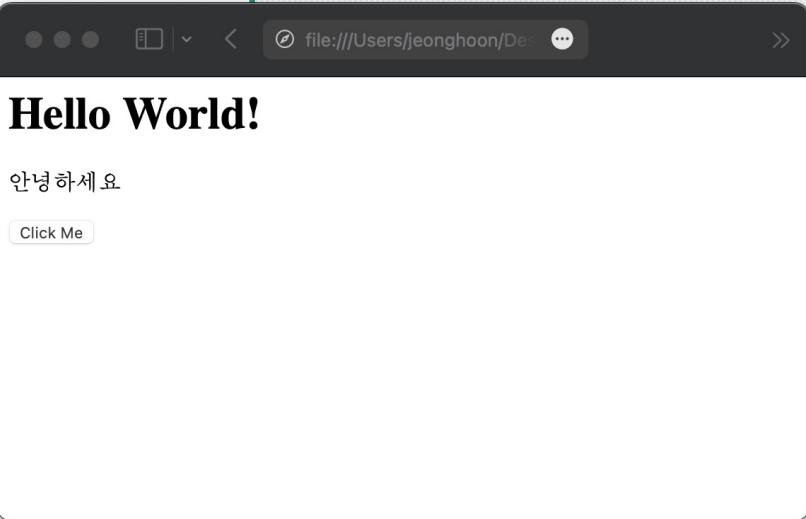
```
web > index.html > ...
1  <html>
2  |  <head>
3  |  |  <title>This is Title</title>
4  |  |  <link rel="stylesheet" href="./style.css">
5  |  |  <script src="./script.js"></script>
6  |  </head>
7  |  <body>
8  |  |  <h1 class="greeting-en">Hello World!</h1>
9  |  |  <p class="greeting-kr">안녕하세요</p>
10 |  |  <button class="click-target">Click Me</button>
11 |  </body>
12 </html>
```

HTML: 예제

```
<body>
  <h1 class="greeting-en">Hello World!</h1>
  <p class="greeting-kr">안녕하세요</p>
  <button class="click-target">Click Me</button>
</body>
```



HTML: 예제



The screenshot shows a web browser window displaying the output of the provided HTML code. The browser interface includes a title bar with file navigation icons and a status bar indicating the file path: file:///Users/jeonghoon/Desktop/untitled.html. The main content area shows the rendered HTML.

```
<body>
  h1
  p
    button
```

Hello World!
안녕하세요
Click Me

HTML: 예제

The screenshot shows a web developer tool interface with the following details:

- Title Bar:** 웹 속성 – index.html
- Toolbar:** Includes icons for zoom, refresh, and various developer tools like 콘솔 (Console), 소스 (Source), 네트워크 (Network), 타임라인 (Timeline), 저장 공간 (Storage), and search.
- Left Panel (DOM Tree):** Shows the structure of the HTML document:

```
<html>
  <head>...</head>
  <body> = $0
    <h1 class="greeting-en">Hello World!</h1>
    <p class="greeting-kr">안녕하세요</p>
    <button class="click-target">Click Me</button>
  </body>
</html>
```
- Right Panel (Element Inspector):**
 - 상자 모델 (Box Model):** Displays the visual hierarchy of the element. The main element has a width of 558px and a height of 298px, with 8px margins on all sides. Inside is a green padding box (8px) and an orange border box (8px).
 - 속성 (Properties):** Lists the properties of the selected element:

```
color: #rgb(0, 0, 0);
display: block;
height: 298px;
```
 - Buttons:** Includes '필터' (Filter), '클래스' (Classes), and 'Pseudo'.
- Preview Area:** Shows the rendered HTML page with the text "Hello World!" and "안녕하세요" and a "Click Me" button.

HTML: 예제

The screenshot shows a web browser window displaying an HTML page. The page content includes a large blue

Hello World!

, a smaller text "안녕하세요" (Hello), and a button labeled "Click Me". A yellow callout box highlights the **body** element with the dimensions **558px x 298px**. To the right, the browser's developer tools sidebar is open, showing the DOM tree. The **<body>** node is selected, with its value set to **\$0**. The expanded body node shows the following code:

```
<html>
  <head>...</head>
  <body> = $0
    <h1 class="greeting-en">Hello</h1>
    <p class="greeting-kr">안녕하세요</p>
    <button class="click-target">Click Me</button>
  </body>
</html>
```

HTML: 예제

The screenshot shows a web browser window displaying an HTML page. The page content includes a large blue header with the text "Hello World!" and a main body section with Korean text "안녕하세요" and an English button labeled "Click Me". A yellow callout box highlights the "Click Me" button, providing its CSS class ("h1.greeting-en"), dimensions ("558px x 37px"), and role ("heading"). To the right of the browser, the developer tools sidebar is open, showing the DOM tree. The tree starts with the root element <html>, followed by <head> and <body>. The <body> node has several children: a highlighted <h1> element with the class "greeting-en" containing the text "Hello", a <p> element with the class "greeting-kr" containing the Korean greeting, and a <button> element with the class "click-target" containing the text "Click Me". The <body> element ends with a closing tag </body>, which in turn ends the <html> element.

```
<html>
  <head>...</head>
  <body>
    <h1 class="greeting-en">Hello</h1>
    <p class="greeting-kr">안녕하세요</p>
    <button class="click-target">Click Me</button>
  </body>
</html>
```

HTML: 예제

The screenshot shows a web browser window displaying an HTML page. The page content includes:

- A large heading **Hello World!** in black font.
- A paragraph of Korean text: **안녕하세요**.
- A button labeled **Click Me**.

The developer tools sidebar is open, showing the DOM structure:

```
<html>
  <head>...</head>
  <body>
    <h1 class="greeting-en">Hello</h1>
    <p class="greeting-kr">안녕하세요</p>
    <button class="click-target">Click Me</button>
  </body>
</html>
```

A tooltip is displayed over the Korean paragraph, indicating its style: **p.greeting-kr 558px x 19px** and its role: **Role: paragraph**.

HTML: 어떤 태그들이 있을까

➤ **div**: 여러 요소를 담는 컨테이너의 역할

➤ 텍스트 관련 태그

- **<h1>** ~ **<h6>**: 제목을 나타낸다
- **<p>**: 단락(paragraph)을 지정한다
- ****: 볼드체(bold)를 설정한다

➤ 목록, 표를 그리는 태그

- **<table>**: 테이블 최상위 태그
- **<tr>**: 테이블의 한 행(row)
- **<td>**: 테이블의 한 칸(data)
- **<th>**: 제목 셀 한 칸(heading)

HTML: 어떤 태그들이 있을까

➤ 목록을 나타내는 태그

- ****: 순서를 갖는 list(ordered list)
- ****: 순서를 갖지 않는 list(unordered list)
- ****: list 내 요소

➤ 멀티미디어 태그

- ****: src 속성에 명시한 이미지를 보여준다
- **<audio>**: src 속성에 명시한 오디오 재생도구를 보여준다
- **<video>**: src 속성에 명시한 비디오 재생도구를 보여준다

HTML: 어떤 태그들이 있을까

➤ 사용자의 입력을 받는 폼(form)태그

➤ 최상위태그 <form>

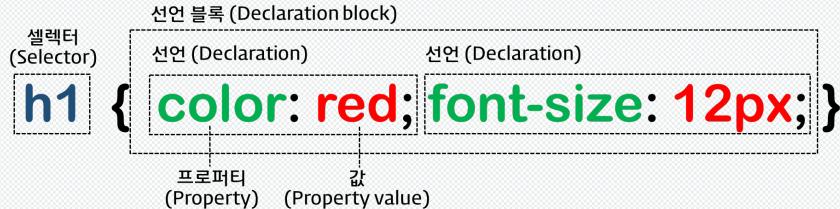
```
<form action="path-to-api-server" method="POST">
```

- **action**: 사용자 입력을 받아서 처리할 API경로
- **method**: 어떤 유형으로 요청을 보낼 것인지

➤ 어떤 값을 입력 받을까

- <**input**>: type속성에 text, date, radio, file 등의 선택지가 있다
- <**textarea**>: 줄글 형태의 입력
- <**select**>: 드롭다운 메뉴로, 여러 선택지 중 하나를 선택
- <**button**>: input button과 유사하지만, 내부에 자식 요소를 넣을 수 있음

CSS: basic



- **선택자(Selector):** 어떤 HTML 요소에 스타일을 적용할까
- **프로퍼티(Property):** 어떤 스타일을 설정할까
- **속성 값(Value):** 프로퍼티에 대한 키워드, 크기, 색상 등

CSS: 예제

web > index.html > ...

```
1  <html>
2    <head>
3      <title>This is
4      <link rel="sty 5
5        <script src="./script.js"></script>
6      </head>
7      <body>
8        <h1 class="greeting-en">Hello World!</h1>
9        <p class="greeting-kr">안녕하세요</p>
10       <button class="click-target">Click Me</button>
11     </body>
12   </html>
```

web > style.css

```
1  p {
2    color: #FF0000;
3    font-size: 2em;
4    font-weight: bold;
5    text-align: center;
6    margin: 0.5em;
7  }
```

CSS: 예제

The screenshot shows a browser window displaying "Hello World!" and a red "안녕하세요" (Hello) text, along with a "Click Me" button. To the right is a developer tools panel for "index.html". The "Elements" tab shows the DOM structure:

```
<html>
  <head>...</head>
  <body>
    <h1 class="greeting-en">Hello World!</h1>
    <p class="greeting-kr">안녕하세요</p>
    <button class="click-target">Click Me</button>
  </body>
</html>
```

The "Elements" tab also displays the current selection: `<p class="greeting-kr">안녕하세요</p>`. The "Elements" tab has a green arrow pointing to the "Style" tab.

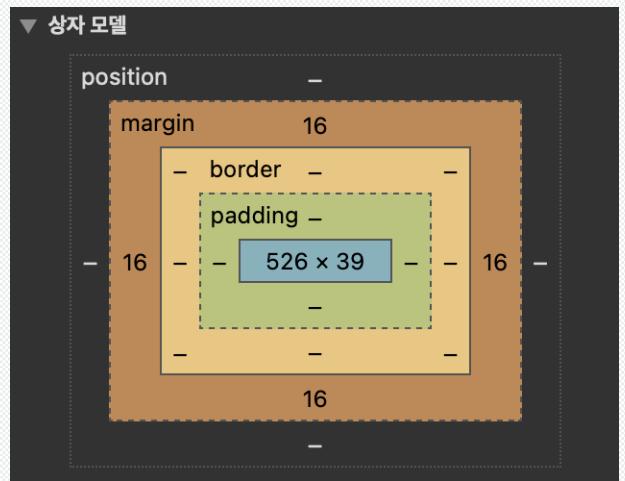
The "Style" tab shows the CSS rules for the selected element:

```
1  p {
2   color: #FF0000;
3   font-size: 2em;
4   font-weight: bold;
5   text-align: center;
6   margin: 0.5em;
7 }
```

The "Elements" tab has a green arrow pointing to the "Style" tab.

CSS: box model

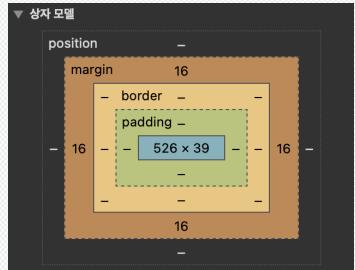
- ▶ 모든 HTML요소는 상자 형태의 영역을 갖는다
- ▶ 이 상자는 Content / Padding / Border / Margin 으로 구성됨



CSS: box model

➤ width / height 프로퍼티

- 요소 내부 content의 크기를 지정한다



➤ padding / margin 프로퍼티

- margin은 border 바깥의 여백, padding은 border 내부의 여백
- 각 방향(top, bottom, left, right)에 대해 지정이 가능하다

➤ border (border-width, border-...)

- 경계선의 스타일, 두께, 색상, 모서리 곡률 등을 지정한다

➤ box-sizing 프로퍼티

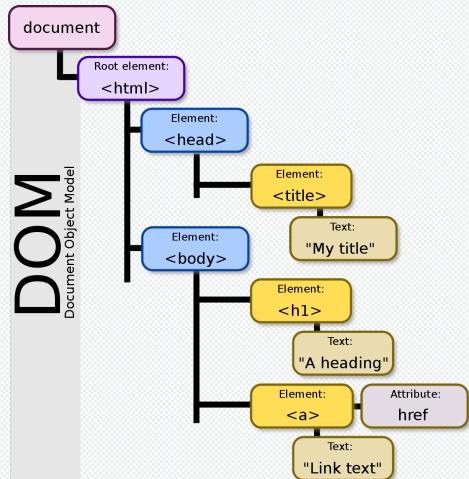
- "content-box"인 경우, width/height가 content를 기준으로
- "border-box"인 경우, width/height가 border를 기준으로

CSS: 정말 너무 많다

- <https://bit.ly/too-many-css-prop>
- 브라우저마다 다른 속성을 갖기도 하고, 너무 많다
- 필요한 속성을 사용해보면서 익히도록 하자

Javascript: basic

- 웹 브라우저, Node.js 환경에서 동작하는 프로그래밍언어
- 둘간의 대표적인 차이점은 클라이언트 사이드 Web API 유무
 - 그중 대표적인 요소는, DOM(Document Object Model) API
 - 이를 통해 JS는 HTML에 접근한다



Javascript: type

- javascript는 동적 타입 언어
 - number, string, boolean, object, undefined 등 타입이 있지만
 - 선언시에는 var, let 등으로 선언하고
 - 실행시에 타입이 결정된다
- 대부분의 문법은 C style에서 크게 벗어나지 않는다

Javascript: grammar

- C와 차이점 위주로 몇가지 꼽아보면
- ASI(Auto Semicolon Insertion)
- 함수 선언, 문자열 표현

```
JS main.js > ...
1   function doSomething(param1, param2) {
2     console.log(param1 + param2);
3   }
4
5   doSomething('hello, ', 'world!');
```

Javascript: role

➤ DOM조작

- HTML요소를 가변적으로 추가 / 삭제 / 조작한다
- Dynamic Web을 구성하는데 필수적(css의 media query와 더불어)

➤ 이벤트처리

- 발생한 HTML이벤트(click, change, ...)에 대한 핸들러(handler)를 등록하고 처리한다

➤ HTTP통신

- 페이지에서 동적으로 필요로 하는 정보를 요청하고 받아온다
- 페이지에서 얻어진 정보를 서버로 전송한다

Javascript: 이벤트처리

➤ 이벤트처리

- 우리가 만든 시스템내에서 발생한 일련의 사건(이벤트)에 대해 어떤 동작, 처리를 해야 하는지 기술하여 이벤트에 대한 동작을 수행하는 것

➤ 이벤트 핸들러

- 특정 HTML요소에 할당되어, 해당 요소에서 발생한 특정 이벤트를 처리하는 함수

➤ 이벤트 종류[1]

- click, focus, change 등등...

[1] https://developer.mozilla.org/en-US/docs/Web/Events#event_listing

Javascript: 이벤트처리 예제

The screenshot shows a code editor interface with two files open:

- index.html**: Contains an HTML document structure with a greeting and a button.
- main.js**: Contains JavaScript code that adds an event listener to the button to log "hit!" to the console and show an alert when the document is ready.

```
index.html
<html>
  <head>
    <script src="./main.js"></script>
  <title>Title</title>
  </head>
  <body>
    <h1>Greeting</h1>
    <div>
      <ul id="target_unordered_li">
        <li>list entity</li>
      </ul>
    </div>
    <button id="hit">hit me</button>
  </body>
</html>
```

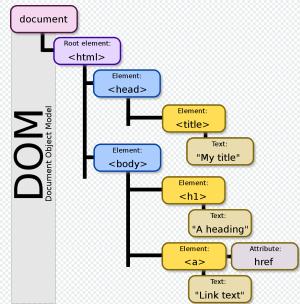
```
main.js
document.addEventListener('DOMContentLoaded', () => {
  let hit_button = document.getElementById('hit');
  hit_button.addEventListener('click', () => {
    console.log('hit!');
  });
  alert('document ready');
});
```

The code editor has a dark theme and includes various icons in the sidebar and footer. The status bar at the bottom shows file statistics, live share, Quokka, and other development tools.

Javascript: DOM조작

▶ DOM(tree)

- HTML요소를 트리 형태의 객체로 변환하여 Javascript 가 접근할 수 있는 형태로 가공한 것



▶ DOM조작

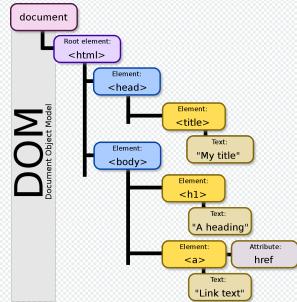
- HTML요소를 가변적으로 추가 / 삭제 / 조작한다
- Dynamic Web을 구성하는데 필수적(css의 media query와 더불어)
- document라는 인터페이스(전역변수)를 통해 접근한다

▶ document 인터페이스

- getElementById(), querySelector(), createElement()
- 위와같은 함수들은 Element라는 객체(HTML요소)를 반환해준다

Javascript: DOM조작

- Element 객체
 - 각 HTML요소에 해당하는 객체
- Element 객체가 가진 함수(메서드)
 - **addEventListener()**: 이벤트를 등록한다
 - **hasAttribute()**: 특정 HTML attribute를 갖는지 확인한다
 - **remove()**: DOM에서 이 요소를 삭제한다



Javascript: DOM조작 예제

The screenshot shows a code editor interface with two files open:

- index.html**:
A simple HTML document with a script tag pointing to main.js, a title, a greeting, a list, and a button.
- main.js**:
A JavaScript file containing event listener logic for the button click. It adds a new list item and logs "hit!" to the console upon click, followed by an alert confirming the document is ready.

```
index.html
<html>
<head>
<script src="./main.js"></script>
<title>Title</title>
</head>
<body>
<h1>Greeting</h1>
<div>
<ul id="target_unordered_list">
<li>list entity</li>
</ul>
</div>
<button id="hit">hit me</button>
</body>
</html>
```

```
main.js
document.addEventListener('DOMContentLoaded', () => {
  let hit_button = document.getElementById('hit');
  let target_list = document.getElementById('target_unordered_list');
  hit_button.addEventListener('click', () => {
    target_list.append(document.createElement('li'));
    console.log('hit!');
  });
  alert('document ready');
});
```

The code editor has a dark theme and includes standard icons for file operations, search, and navigation. The status bar at the bottom shows file details like "Live Share", "22 mins", and "Quokka".

Javascript: HTTP통신

➤ HTTP(Hyper Text Transfer Protocol)

- 웹페이지에서 필요한 자원(html, css, js)를 요청하고 받아오는 프로토콜

HTTP/1.1 200 OK

Date: Sat, 09 Oct 2010 14:28:02 GMT

Server: Apache

Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT

ETag: "51142bc1-7449-479b075b2891b"

Accept-Ranges: bytes

Content-Length: 29769

Content-Type: text/html

<!DOCTYPE html... (here comes the 29769 bytes of the requested web page)

Javascript: HTTP통신

- Javascript는 어떻게 HTTP통신을 하는가
 - Axios, Ajax(전통적)라는 HTTP클라이언트를 통해 통신한다
 - 아주 간단하게,,, 라이브러리라고 생각하면 될듯

- JS는 DOM접근이 가능하니까... 사용자로부터 전달받은 값을 변수에 담아서 Axios를 통해 요청을 만들어 보낼 수 있겠다.
 - Axios, Ajax(전통적)라는 HTTP클라이언트를 통해 통신한다

Javascript: HTTP통신 예제

- node.js 설치
 - <https://nodejs.org/ko/download>
- npm init
- npm i webpack webpack-cli axios -D

Javascript: HTTP통신 예제

The terminal window shows the following directory structure:

```
> node_modules
index.html
main.js
package-lock.json
package.json
webpack.config.js
```

The code editor shows the `package.json` file with the following content:

```
package.json > {} scripts
▶ 디버그
6   "scripts": [
7     "build": "webpack"
8   ],
```

Javascript: HTTP통신 예제

```
JS main.js > ...
1 import axios from 'axios';  48.8k (gzipped: 17.9k)
2
3 ˜ document.addEventListener('DOMContentLoaded', () => {
4      let hit_button = document.getElementById('hit');
5      let target_list = document.getElementById('target_unordered_list');
6
7 ˜ hit_button.addEventListener('click', () => {
8      axios.get('http://www.tcpsschool.com/webbasic/works').then((response) => {
9          console.log(response);
10     });
11 });
12 alert('document ready');
13 });
14
```

Javascript: HTTP통신 예제