

파트 01

웹의 이해

인터넷

전 세계의 모든 컴퓨터를 하나의 통신망 안에서 연결한다는 의미를 가진
“**I**nternational **N**etwork”의 약자

웹(Web)

- 월드 와이드 웹(**W**orld **W**ide **W**eb):
인터넷에 연결된 컴퓨터를 통해 사람들이 정보를 공유할 수 있는 공간
- 간단하게 **WWW**, **W3** 그리고 월드 와이드 웹을 줄여서 **웹 (Web)** 이라고도 불림

웹(Web)의 시작

- 1980년대, 스위스의 한 유럽 입자 물리 연구소, 세른(CERN)의 컴퓨터과학자 **팀 버너스-리에**에 의해 탄생
- 연구원들 간의 신속한 정보교환을 위해 고안되었음

웹(Web)의 특징

- 정보를 **하이퍼 텍스트** 형식으로 표현
- **하이퍼 텍스트 (링크)** 를 따라 이동하며 다양한 정보/문서들을 연결/제공

💡 웹 페이지 vs 웹 사이트

- **웹 페이지(Web Page)** : 웹 브라우저에서 볼 수 있는 하나의 개별적인 문서
- **웹사이트(Website)** : 여러 개의 웹 페이지가 모여서 이루어진 전체적인 사이트

웹 페이지는 개별 문서이지만, 여러 웹 페이지가 모이면 **웹사이트**가 됨

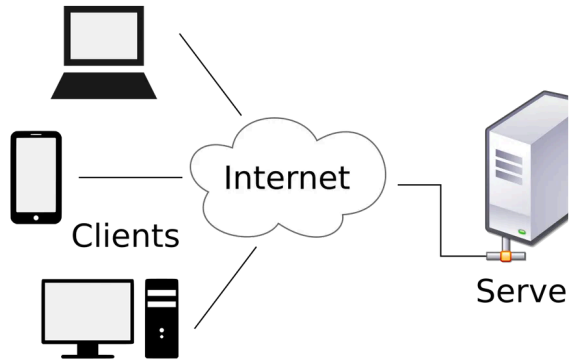
웹의 구조

클라이언트와 서버

- **클라이언트 (Client)** : 서비스를 이용하는(요청하는) 컴퓨터
- **서버 (Server)** : 서비스를 제공하는 컴퓨터

프로토콜

- 서로 정보를 주고 받을 때에 지켜야 하는 **약속**
- 반드시 이 약속을 지켜서 통신해야 하며, 이 약속을 **프로토콜** 이라고 함.



인터넷으로 연결된 **클라이언트**와 **서버**는

웹 프로토콜인

HTTP(HyperText Transfer Protocol)

를 사용하여 데이터를 주고 받음

https://ko.wikipedia.org/wiki/클라이언트_서버_모델

웹 개발 직무 이해



프론트엔드(Frontend)

웹 서비스에서 사용자의 **측면(Client-side)**의 그래픽 사용자 인터페이스로
사용자와 상호작용(글자 입력, 버튼 클릭, 화면 출력 등)을 담당

백엔드(Backend)

웹 서비스에서 사용자의 눈에 보이지 않는 **서버 측(Server-side)**에서

프론트엔드 에서 전달받은 데이터와 요청을 내부 데이터와 연산을 활용하여 처리하고
최종적으로 **프론트엔드** 단에 결과를 전달

HTML, CSS, Javascript 소개

HTML



<https://ko.wikipedia.org/wiki/HTML>

- **HTML (Hyper Text Markup Language)**
- 웹 페이지를 연결하는 기능을 가진 텍스트이자 웹 페이지의 구조를 명시하는 언어
- HTML은 **< > (괄호)** 를 사용하고 이것을 **태그** 라고 부름
- **태그** 는 웹 페이지의 구성 요소 하나하나로 역할을 함

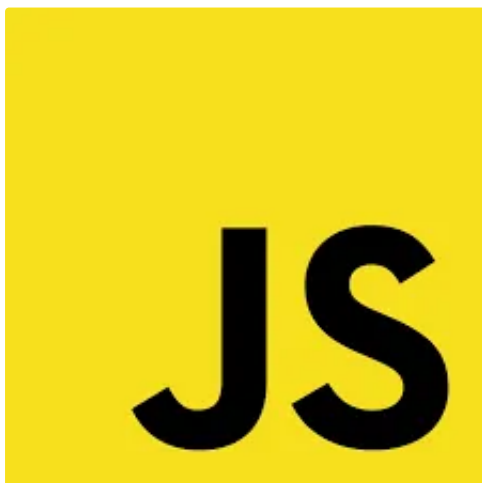
<태그> 감싸진 글자가 그 태그의 역할을 맡음 **</태그>**

- 💡 여는 태그와 닫는 태그가 한 쌍일 수도 있고
닫는 태그 없이 여는 태그가 단독으로 쓰일 수 있음



<https://ko.wikipedia.org/wiki/CSS>

- CSS
- 웹 페이지 구성 요소들을 꾸밈



- JavaScript
- 웹의 내부 로직을 담당

IDE

- IDE (Integrated Development Environment, 통합 개발 환경)
- 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공하는 소프트웨어
- 가장 대중적인 IDE로는 Eclipse, VScode, IntelliJ 등이 있음

첫 HTML 만들기

- `<!DOCTYPE html>` : document type이 html임을 의미.
HTML5 문서임을 선언하는 태그로
브라우저가 올바르게 HTML문서를 렌더링하도록 도움.
- `<html></html>` : html 문서의 시작과 끝을 나타내는 태그.
모든 HTML 콘텐츠는 이 태그 안에 포함이 되어야 함.
- `<head></head>` : html파일에 대한 정보(메타 정보)를 작성
 - `<meta charset = "">` : 문서의 문자 인코딩을 설정하는 태그.
보통 한국어를 지원하는 "UTF-8"을 사용
- `<body></body>` : 실제 웹 페이지에서 사용자에게 출력되는 콘텐츠를 포함하는 태그

```
<!DOCTYPE html> <html> <head>
<meta charset="UTF-8"> <title>
첫 번째 HTML </title> </head>
<body> Hello, html! </body>
</html>
```

← → ↻ ⓘ 127.0.0.1:5500/2주차/WebStart/first.html

Hello, html!

결과물



! + Tab 이면 손쉽게 기본적인 html 구조 코드를 작성해준다.

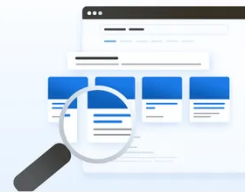
자주 쓰는 HTML 태그

• 자주 쓰는 HTML 태그 순위

HTML Study - AWR SEO Guide

This study looks at 11 mm index pages gathered from Top 20 Google results to uncover how people write HTML markup as

 <https://www.advancedwebranking.com/seo/html-study>



h 태그

- **Heading**
- `<h1> ~ <h6>` 요소는 6단계의 **구획 제목**을 나타냄
- 단계는 `<h1>` 이 가장 높고, `<h6>` 은 가장 낮음

```
<h1>Heading Level 1</h1>
<h2>Heading Level 2</h2>
<h3>Heading Level 3</h3>
<h4>Heading Level 4</h4>
<h5>Heading Level 5</h5>
<h6>Heading Level 6</h6>
```

Heading Level 1

Heading Level 2

Heading Level 3

Heading Level 4

Heading Level 5

Heading Level 6

<a> 태그

- **Anchor**, 앵커 요소라고도 함
- href (**hyper** **reference**) 특성을 통해 다른 URL 또는 페이지 등으로 연결할 수 있는 하이퍼링크를 생성

```
<!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title> GO </title>
<body> <h2>We Can Go</h2> <a href="https://programmers.co.kr"
<a href="first.html">My First Page</a>
</html>
```

We Can Go

[Programmers](#) [My First Page](#)

 태그

- 텍스트 안에 **줄바꿈**을 생성


```
<!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title> GO </title> <body> <h2>We Can Go</h2> <a href="https://programmers.co.kr"> <br> <a href="first.html">My First Page</a> </body> </html>
```

We Can Go

[Programmers](#)
[My First Page](#)

로그인 화면 만들기

<form> 태그

- 데이터를 전송하기 위한 대화형 컨트롤이 포함된 문서 구획을 나타냄

<input> 태그

- 사용자로부터 데이터를 받기 위한 대화형 컨트롤을 생성

```
<!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title> LOGIN </title> </head> <body> <h1>Login</h1> <form> <input type="text"> </form> </body> </html>
```


Login

- 여기에 속성을 추가하여 그럴듯한 Login 폼을 만든다
- `<input>` type

`<input>`: The HTML Input element - HTML: HyperTe...

The `<input>` HTML element is used to create interactive controls for web-based forms in order to accept data from

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

 mdn web docs

```
<!DOCTYPE html> <html> <head>
<meta charset="UTF-8"> <title>
LOGIN </title> </head> <body>
<h1>Login</h1> <form> ID :
<input type="text"> <br> PW :
<input type="password"> <br>
<input type="button"
value="Login"> </form> </body>
</html>
```

Login

ID :

PW :



`<form>` 과 `<input>` 은 왜 같이 사용할까?

1. 사용자 입력을 그룹화하고 전송하기 위해

`<input>` : 사용자로부터 데이터를 입력받음

`<form>` : 입력 요소들을 감싸고, 입력된 데이터를 서버로 전송

2. 엔터 키 제출 및 버튼 동작을 위해

`form` 내에서 `input` 을 사용하면, 사용자가 입력 후 `Enter` 를 눌렀을 때 자동으로 폼이 제출