

HTML & CSS

소플의 처음 만남 React



Hyper Text Markup Language

웹 페이지의 뼈대를 구성하기 위해 사용하는 마크업 언어

문서나 데이터를 처리하기 위해 추가되는 정보

Markup Language & Metadata

구분	마크업 언어 (Markup Language)	메타데이터 (Metadata)
정의	문서나 데이터를 구조화하고 서식을 지정하는 언어	데이터 자체가 아닌, 데이터를 설명하는 정보
목적	문서의 구조를 정의하고, 내용의 표현 방식, 기능(링크, 이미지 등)을 지정	데이터에 대한 속성, 의미, 관계 등을 설명
형식	태그(tag)나 특정한 문법을 사용하여 구성	일반적으로 키-값(key-value) 형태로 저장
대상	문서(텍스트, 이미지 등)	데이터(파일, 이미지, 웹 페이지)
예시	HTML, XML , Markdown , LaTeX 등	HTML의 <meta> 태그, EXIF(사진 메타데이터), JSON-LD 등
사용 분야	웹 페이지, 문서 작성, 데이터 구조화	검색 엔진 최적화(SEO), 데이터 관리, 디지털 아카이빙

- 마크업 언어는 주로 콘텐츠의 표현에 초점을 맞추고, 메타데이터는 콘텐츠에 대한 추가 정보를 제공하는 데 중점을 둡니다.
- 마크업 언어는 문서의 구조와 표현 방식을 정의하는 데 사용되며, 메타데이터는 데이터에 대한 설명을 제공합니다.
- 마크업 언어는 문서 내에 메타데이터를 포함할 수 있습니다.

HTML Standard

1. W3C (World Wide Web Consortium)

- ✓ URL: <https://www.w3.org/TR/html52/>
- ✓ W3C는 웹 표준을 제정하는 기관으로, HTML의 공식 명세(specification)를 제공합니다.
- ✓ HTML5 및 최신 웹 기술의 표준을 확인할 수 있습니다.

2. MDN Web Docs (Mozilla Developer Network)

- ✓ URL: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
- ✓ HTML 태그 목록과 사용법을 상세히 설명하며, 예제와 브라우저 지원 여부도 제공합니다.
- ✓ 최신 HTML 표준을 가장 쉽게 이해할 수 있는 자료 중 하나입니다.

3. WHATWG (Web Hypertext Application Technology Working Group)

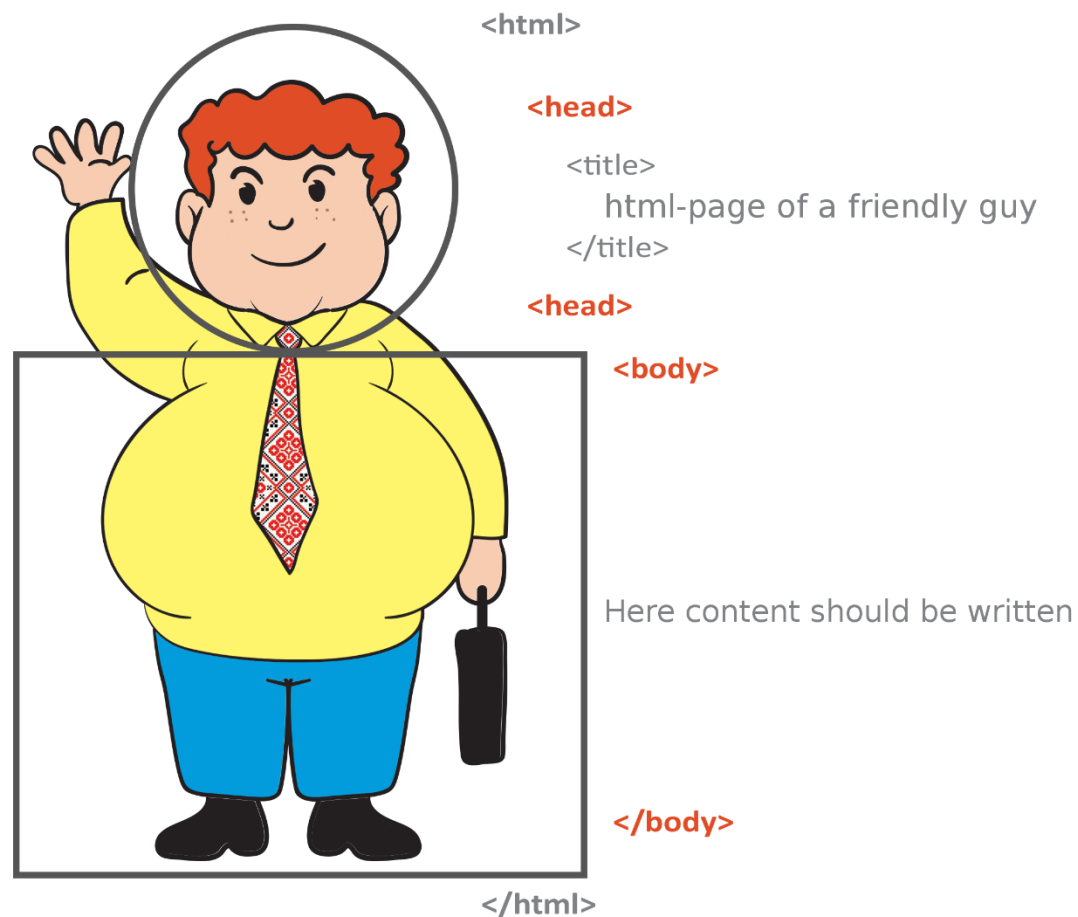
- ✓ URL: <https://html.spec.whatwg.org/multipage/>
- ✓ HTML의 지속적인 발전을 담당하는 단체로, 최신 HTML Living Standard(HTML LS)를 유지보수한다.
- ✓ 가장 최신의 HTML 사양을 확인할 수 있다.

Emmet

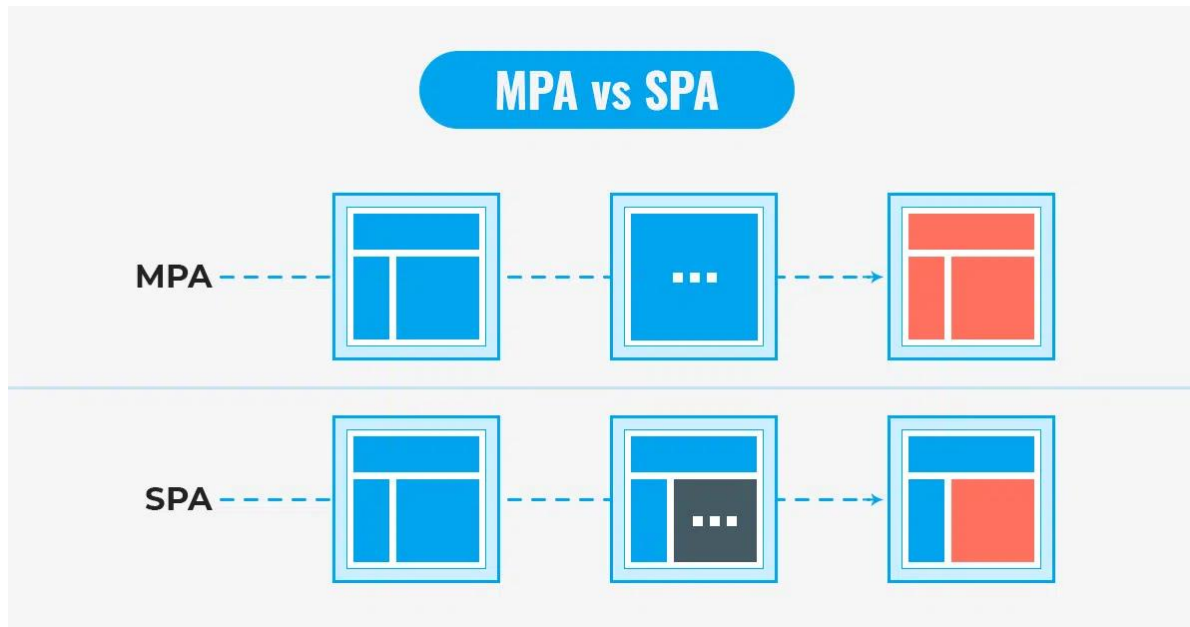
- Emmet은 HTML, CSS 등의 마크업 코드를 빠르게 작성할 수 있도록 도와주는 코딩 단축기(Snippets) 및 자동 완성 도구입니다.
- 주로 VS Code, Sublime Text, Atom, Brackets 같은 코드 편집기에서 사용되며, 짧은 문법을 입력하면 자동으로 확장(expansion)하여 코드 작성을 가속화합니다.
- Emmet 공식 웹사이트 : <https://emmet.io/>
- Emmet GitHub 저장소 : <https://github.com/emmetio/emmet>
- Emmet 문서 : <https://emmet.io/documentation/>
- cheat-sheet : <https://docs.emmet.io/cheat-sheet/>



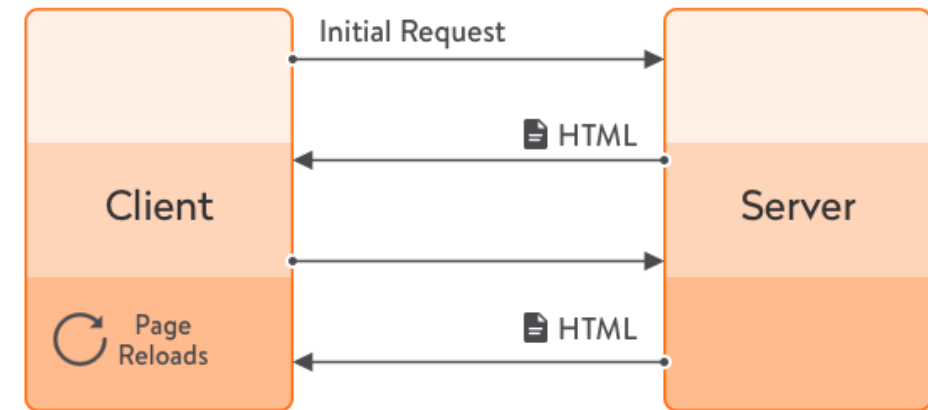
HTML Tutorial



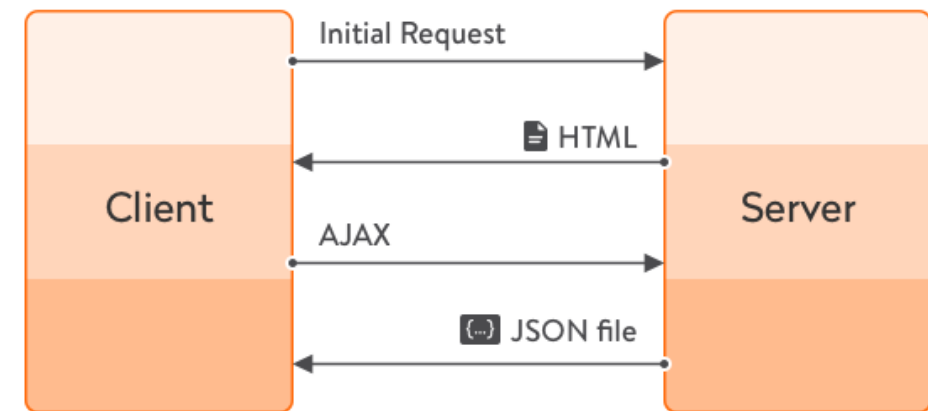
MPA vs SPA



Multi-page app lifecycle

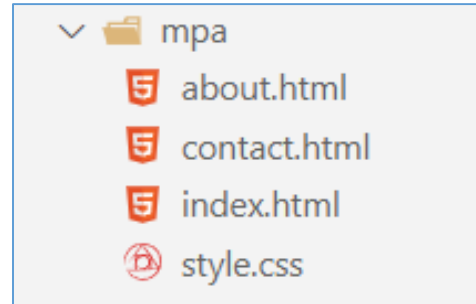


Single-page app lifecycle



[SPA vs MPA]

MPA

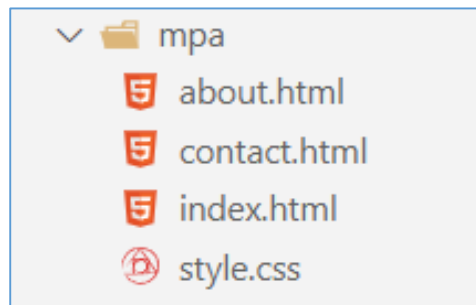


about.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>MPA - About</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <nav>
    <a href="index.html">Home</a>
    <a href="about.html">About</a>
    <a href="contact.html">Contact</a>
  </nav>
  <h2>About</h2>
  <p>회사 소개 페이지입니다.</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>MPA - Home</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <nav>
    <a href="index.html">Home</a>
    <a href="about.html">About</a>
    <a href="contact.html">Contact</a>
  </nav>
  <h2>Home</h2>
  <p>홈 페이지입니다.</p>
</body>
</html>
```


MPA



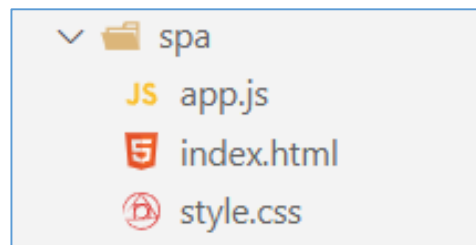
```
body {  
    font-family: Arial, sans-serif;  
    text-align: center;  
}  
nav {  
    margin-bottom: 20px;  
}  
nav a {  
    margin: 0 10px;  
    text-decoration: none;  
    font-weight: bold;  
    color: blue;  
}
```

style.css

```
<!DOCTYPE html>  
<html lang="ko">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>MPA - About</title>  
    <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
  
    <nav>  
        <a href="index.html">Home</a>  
        <a href="about.html">About</a>  
        <a href="contact.html">Contact</a>  
    </nav>  
  
    <h2>About</h2>  
    <p>회사 소개 페이지입니다.</p>  
  
</body>  
</html>
```

contact.html

SPA



index.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SPA 예제</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

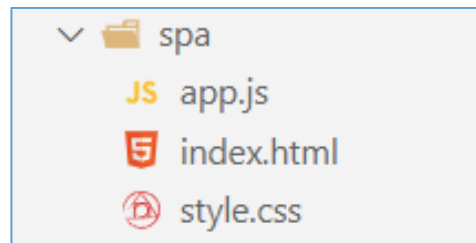
  <nav>
    <a href="#" onclick="navigate('home')">Home</a>
    <a href="#" onclick="navigate('about')">About</a>
    <a href="#" onclick="navigate('contact')">Contact</a>
  </nav>

  <div id="app">
    <h2>Welcome to SPA</h2>
    <p>여기에 페이지 내용이 표시됩니다.</p>
  </div>

  <script src="app.js"></script>

</body>
</html>
```

SPA



```
const routes = {
  home: "<h2>Home</h2><p>홈 페이지입니다.</p>",
  about: "<h2>About</h2><p>회사 소개 페이지입니다.</p>",
  contact: "<h2>Contact</h2><p>연락처 페이지입니다.</p>"
};

function navigate(page) {
  document.getElementById("app").innerHTML = routes[page] || "<h2>404</h2><p>페이지를 찾을 수 없습니다.</p>";

  // URL을 변경하여 브라우저 기록에 반영
  window.history.pushState({ page }, page, `#${page}`);
}

// 뒤로가기/앞으로가기 이벤트 처리
window.onpopstate = function(event) {
  if (event.state) {
    navigate(event.state.page);
  }
};
```

app.js

```
body {
  font-family: Arial, sans-serif;
  text-align: center;
}
nav {
  margin-bottom: 20px;
}
nav a {
  margin: 0 10px;
  text-decoration: none;
  font-weight: bold;
  color: blue;
  cursor: pointer;
}
```

style.css

Cascading Style Sheets

웹 페이지의 레이아웃과 글꼴 색상 등의 디자인을 입히는 역할

CSS Zen Garden



프론트엔드 개발자가 되기 위한 로드맵

Node.js

Run JavaScript Everywhere

서버 측에서 자바스크립트를 실행할 수 있게 해주는 런타임 환경

Platform, Environment, Framework, Runtime

구분	설명	예시
Platform	소프트웨어가 실행될 수 있는 기반 환경	운영체제 (Windows, macOS, Linux), 클라우드 플랫폼 (AWS, Azure), 웹 플랫폼
Environment	특정 소프트웨어가 실행될 수 있도록 구성된 조건	개발 환경, 운영 환경, 테스트 환경
Framework	소프트웨어 개발을 위한 뼈대 또는 기반 구조	웹 프레임워크 (React, Angular, Vue.js), 모바일 프레임워크 (React Native, Flutter)
Runtime	프로그램이 실행되는 동안 필요한 환경	JRE (.NET Framework), Node.js

멋진 프론트엔드를 위한 세 가지 파이썬 프레임워크

용어	Node.js에서의 의미
Platform	Node.js가 실행되는 OS(Windows, Linux) 또는 배포되는 클라우드(AWS, Azure)
Environment	로컬 개발 환경(VS Code), 실행 환경(Node.js Runtime), 운영 환경(Production)
Framework	Express.js (Node.js에서 웹 서버 개발을 쉽게 해줌)
Runtime	Node.js Runtime (JavaScript를 서버에서 실행할 수 있도록 제공)

JavaScript runtime environment

Node.js는 OS 수준의 플랫폼이 아니다

- 플랫폼(Platform)은 일반적으로 하드웨어나 운영체제 수준에서 실행 환경을 제공한다.
- 반면 Node.js는 특정 운영체제에 종속되지 않고, 다양한 OS에서 실행 가능한 JavaScript 실행 환경을 제공한다.
- 따라서 Node.js 자체는 OS를 대체하지 않고, OS 위에서 동작하는 "환경(Environment)"에 가깝다.

Node.js는 특정 기능을 강제하지 않는다

- 플랫폼(Platform)은 일반적으로 개발자에게 특정한 개발 방식이나 구조를 강제한다.
- 예를 들어, .NET, Spring, Django 같은 플랫폼은 개발 방식과 구조를 어느 정도 강제한다.
- 하지만 Node.js는 특정한 개발 방식을 강요하지 않고, 실행 환경만 제공한다.
 - ✓ Node.js는 "서버로 사용할 수도 있고, CLI 도구로도 활용 가능"
 - ✓ 즉, 웹 프레임워크가 아니며, 특정 아키텍처를 강요하지 않는다.

기존의 "Runtime Environment" 개념과의 유사성

- Node.js는 **JavaScript 코드 실행을 위한 "런타임 환경(Run-time Environment)"**으로 설계되었다.
- 이는 Java의 JRE (Java Runtime Environment) 와 유사한 개념이다.
- JRE (Java Runtime Environment)
 - ✓ Java 애플리케이션이 실행될 수 있도록 환경을 제공
 - ✓ OS 위에서 실행되며, 개발 방식 자체는 강제하지 않음
- Node.js (JavaScript Runtime Environment)
 - ✓ JavaScript가 브라우저가 아닌 환경에서도 실행될 수 있도록 함
 - ✓ V8 엔진 기반에서 실행되며, OS 위에서 작동
- 즉, "JavaScript 실행 환경"을 제공하는 것이 핵심 역할이므로 "환경(Environment)"이라는 용어가 적절하다.

-
- Node.js는 운영체제(OS)나 프레임워크가 아니며, 특정 개발 방식을 강요하지 않는다.
 - JavaScript 코드 실행을 위한 "런타임 환경(Run-time Environment)"이므로 Environment라는 용어를 사용한다.
 - Java의 JRE처럼, Node.js는 특정 OS 위에서 동작하는 실행 환경이기 때문에 플랫폼(Platform)보다는 Environment라는 표현이 더 적절하다.

❖ 즉, Node.js는 "JavaScript를 서버 및 다양한 환경에서 실행할 수 있도록 만들어진 런타임 환경"이다!

1. Node.js 설치

<https://nodejs.org/en>

2. Node.js 버전 확인

> node -v

> npm -v

3. 간단한 Node.js 실행 1

> node

> console.log("Hello, Node.js!")


4. 간단한 Node.js 실행 2

> copy con hello.js

console.log("Hello, Node.js!")

^Z

> node hello.js

Node.js 다운로드 (LTS) 

```
C:\Users\wisadora>node -v
v18.12.0

C:\Users\wisadora>npm -v
8.19.2

C:\Users\wisadora>node
Welcome to Node.js v18.12.0.
Type ".help" for more information.
> console.log("Hello, Node.js!")
Hello, Node.js!
undefined
> .exit
```

```
C:\Users\wisadora>copy con hello.js
console.log("Hello, Node.js!")
^Z
1개 파일이 복사되었습니다.

C:\Users\wisadora>node hello.js
Hello, Node.js!
```

Integrated **D**evelopment **E**nvironment

소프트웨어 개발을 쉽게 할 수 있도록 코드 편집, 디버깅, 빌드, 실행 등의 기능을 하나의 프로그램에 통합한 개발 도구

IDE의 주요 기능

- 코드 편집기(Code Editor) : 구문 강조, 자동 완성, 코드 포매팅 등을 지원
- 디버거(Debugger) : 코드 실행 중 오류를 탐지하고 수정할 수 있도록 지원
- 빌드 자동화 : 소스 코드를 컴파일하고 실행 파일을 생성하는 과정을 자동화하는 도구를 제공
- 버전 관리 연동 : Git, SVN 같은 형상 관리 시스템과 통합
- 프로젝트 관리 : 여러 파일을 체계적으로 관리 가능
- 패키지 관리자 연동 : npm, pip, Maven 등과의 호환성 제공
- 플러그인 및 확장성 : 다양한 플러그인이나 확장 기능을 통해 IDE의 기능을 추가하거나 사용자 맞춤형 환경을 구성

대표적인 IDE 와 코드 편집기

IDE	지원 언어	특징
Visual Studio	C#, C++, Python, JavaScript 등	강력한 Windows 개발 지원, .NET 개발에 최적화
IntelliJ IDEA	Java, Kotlin, Python 등	Java 개발에 최적화, 스마트 코드 보완 기능
PyCharm	Python	Python 개발에 특화된 IDE
Eclipse	Java, C++, Python 등	오픈소스, 플러그인 확장성이 뛰어남
Android Studio	Java, Kotlin	Android 앱 개발 전용 IDE
Xcode	Swift, Objective-C	macOS 및 iOS 앱 개발 지원
NetBeans	Java, PHP, C++	오라클에서 제공하는 오픈소스 IDE

코드 편집기	특징
VS Code	빠르고 가벼우며, 다양한 플러그인 지원
Sublime Text	빠른 성능, 최소한의 기능 제공
Atom	Git Hub에서 개발한 코드 편집기 (현재 지원 종료)
Notepad++	가볍고 빠른 Windows용 텍스트 편집기

VS Code

가벼운 코드 편집기이자 IDE 기능을 갖춘 개발 도구

다운로드 및 설치

<https://code.visualstudio.com/>

VS 코드의 장점

- **무료 및 오픈 소스** : 누구나 무료로 사용할 수 있으며, 커뮤니티를 통해 활발하게 개발되고 있습니다.
- **크로스 플랫폼** : Windows, macOS, Linux 등 다양한 운영체제에서 동일하게 사용할 수 있습니다.
- **활발한 커뮤니티** : 많은 사용자와 개발자들이 활발하게 활동하며, 다양한 정보를 공유하고 문제를 해결하는 데 도움을 받을 수 있습니다.
- **풍부한 확장 기능** : 다양한 확장 기능을 통해 기능을 확장하고 개인화 할 수 있어 활용도가 높습니다.

