

# Hello Javascript

## #01. 웹 사이트 관련 직군 / 언어 분류



## #02. 실행환경 구성

### 1) Node.js

#### 설치확인

윈도우의 경우 명령프롬프트 실행

WinKey + R > cmd (엔터)

맥의 경우 터미널 실행

Cmd + Space > 터미널 검색

명령어 수행

```
node --version
```

결과값이 출력되지 않을 경우 <https://nodejs.org> 에서 프로그램 내려받아 설치 필요함.

대부분의 경우 LTS 버전 권장.

Mac M1 버전의 경우 17.1 이상 버전 필요함.

설치 완료 후 열어두었던 명령프롬프트나 터미널을 종료하고 재시작.

앞서 수행한 버전확인 명령어를 통해 설치 완료 확인.

## 2) Visual Studio Code

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

### Javascript 작성에 도움을 주는 Visual Studio Code Extension

이름	설명
Color Highlight	색상 코드 값을 실제 그 색상으로 강조한다.
Prettier - Code formatter	코드의 줄바꿈, 들여쓰기등을 자동으로 정렬한다. 사용방법: <code>Ctrl+Shift+P</code> > <code>Format Document</code> 명령 선택
ESLint	Javascript 구문 검사 기능 터미널에서 <code>npm install -g eslint</code> 수행 필요.
Code Runner	설치후 <code>Alt+Ctrl+N</code> 단축키로 현재 소스코드 실행. 환경설정에서 <code>Code-runner: Clear Previous Output</code> 항목 체크
Tabnine	AI기반 소스코드 자동완성 도구

## #03. Hello Javascript

### 1) 소스코드

프로그램 명령어를 저장해 놓은 파일

사용하는 프로그래밍 언어에 따라 확장자가 서로 다르다. (\*.java, \*.js, \*.py, \*.cpp)

일반적인 메모장용 텍스트파일의 확장자를 강제로 수정해서 사용한다.

### 2) 주석문

프로그램 소스코드 안에 명시하는 필기.

소스코드에 대한 부연 설명을 작성하는 용도.

#### 한 줄 전용

// 를 명시하고 그 뒤에 주석 내용을 작성한다. 여러 줄을 지정해야 할 경우 모든 행 앞에 //가 붙는다.

```
// 여기는 주석입니다.
// 여기는 주석입니다.
// 여기는 주석입니다.
// 여기는 주석입니다.
```

## 여러줄 처리

`/*과 */` 사이에 주석 내용을 작성한다. 이 안에서는 줄바꿈이 자유롭다.

```
/*
여기는 주석입니다.
여기는 주석입니다.
여기는 주석입니다.
여기는 주석입니다.
*/
```

## 3) 첫 번째 Javascript program

### JS가 HTML안에 존재하는 경우.

원래 JS는 HTML에 기생하는 존재.

```
simple.html
```

### JS코드와 HTML을 분리

하나의 파일에 HTML(뼈대) + CSS(옷) + JS(동작)가 혼합되면서 코드가 복잡해지고 길어진다. (스파게티 코드)

CSS와 JS를 별도의 파일로 분리하는 것이 바람직.

확장자가 `*.js`인 파일이 순수 JS코드만 작성

```
HelloWorld.js
```

### JS파일을 참조하는 HTML

```
HelloWorld.html
```

## 4) JS 소스코드 작성시 주의 사항

1. 대소문자를 엄격히 구분한다.
2. 줄바꿈과 띄어쓰기는 개발자가 코드를 읽기 수월하게 하기 위한 용도일 뿐 실행에는 아무런 영향이 없다.

## 5) JS 파일 직접 실행

### 명령프롬프트(win)를 통한 실행

- `WinKey+R` -> `cmd` (엔터) 로 명령프롬프트 실행
- 소스파일이 존재하는 폴더로 이동 `cd` 폴더경로
  - 폴더 위치가 C드라이브가 아닌 경우 `/d` 옵션 적용 -> `cd /d` 폴더경로
- `node` 파일이름 명령으로 코드 실행

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.631]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\EZEN>cd C:\Users\EZEN\leekh\ezzen-javascript-week\01-시작하기

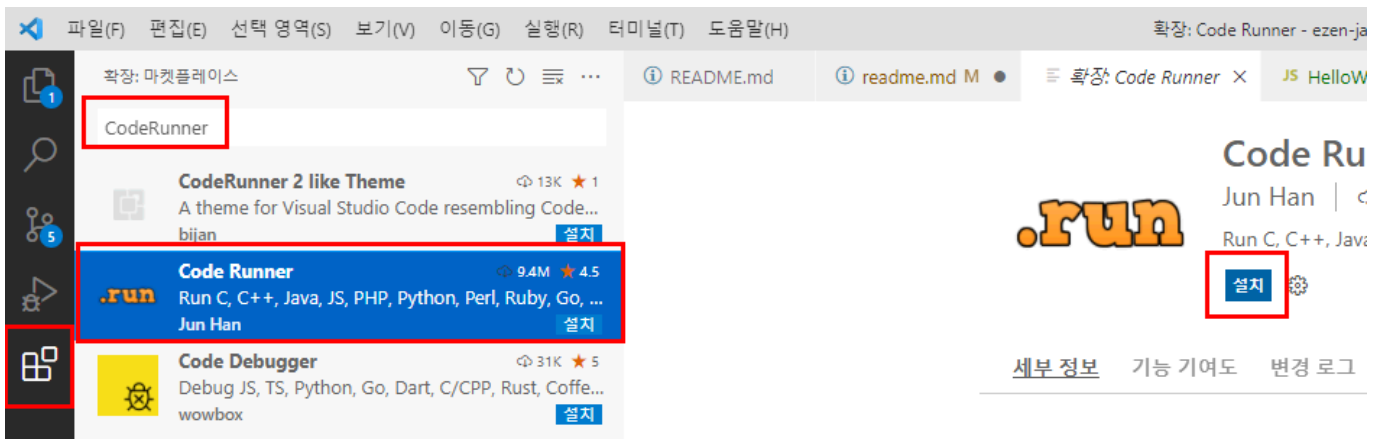
C:\Users\EZEN\leekh\ezzen-javascript-week\01-시작하기>node HelloWorld.js
MyMessage1
안녕하세요. Javascript1
안녕하세요. Javascript2
안녕하세요. Javascript3
MyMessage2
안녕하세요. Javascript4
MyMessage2-1
안녕하세요. Javascript5
안녕하세요. Javascript6

C:\Users\EZEN\leekh\ezzen-javascript-week\01-시작하기>

```

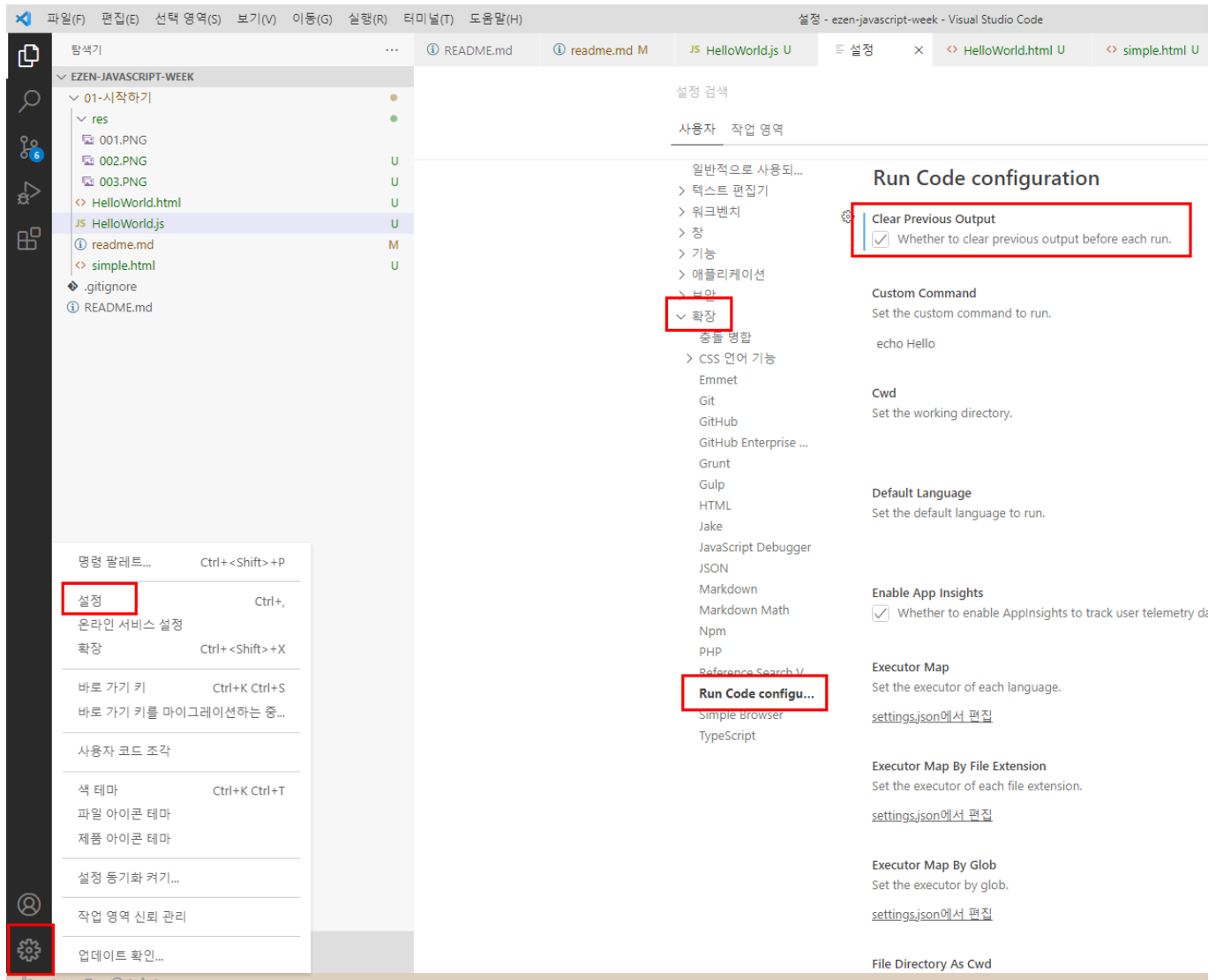
## VSCode를 통한 실행

- **Code Runner** 확장 익스텐션 설치



- 코드 창에서 **Ctrl + Alt + N** 으로 실행
- 원하는 부분만 드래그 후 부분 실행 가능함.

실행시 이전 출력 내용 삭제 설정



## #04. 프로그램 소스코드가 실행되는 과정

### 1) 컴파일 언어

반드시 기계어로 컴파일되어야만 실행시킬 수 있는 프로그래밍 언어

1. 개발자가 프로그램 소스코드를 작성.
2. 작성한 소스코드를 2진수(바이너리, 기계어) 형태로 변환 -> **컴파일(compile)**
  - 소스코드를 컴파일 해 주는 소프트웨어 -> **컴파일러(compiler)**
3. **컴파일된** 바이너리를 실행한다.
  - 한번 컴파일된 파일은 2진수 형태로 저장되어 있기때문에 재실행시 컴퓨터가 해석할 필요가 없다.
  - 컴파일된 결과물은 독립 실행이 가능한 형태이기 때문에 별다른 도구가 필요 없다.

C, C++, Java 등

### 2) 인터프리터 언어

컴파일러를 거쳐서 기계어로 변환되지 않고 바로 실행되는 프로그래밍 언어

1. 컴파일을 거치지 않고 매 실행시마다 소스코드를 해석하는 과정이 반복적으로 수행되는 형태.
2. 컴파일을 하지 않는다는 간편함 때문에 상대적으로 배우기 쉽다.
3. 컴파일 언어보다는 실행속도가 느리다.

- 대용량 서비스에 적용하는 것은 적합하지 않다.
- 4. 컴파일을 하지 않는 대신 실행을 하기위해 해석기가 매번 필요하다.
  - Javascript코드는 실행하기 위해서 반드시 Node나 웹브라우저가 필요하다.

## Javascript, Python

### 스크립트 언어

인터프리터 방식에서 사용하기 위해 고안된 프로그래밍 언어.