

Javascript

채석환

01. 웹 개발 시작하기

01-1 웹 개발 알아보기

01-2 웹 개발, 어디서부터 시작할까



웹 개발이란

웹 개발은 웹 브라우저 화면에 보이는 겉모습을 만드는 것이 전부가 아니다
웹 사이트에서 사용자에게 제공할 기능과 서비스까지 모두 담아야 한다



웹 개발이란

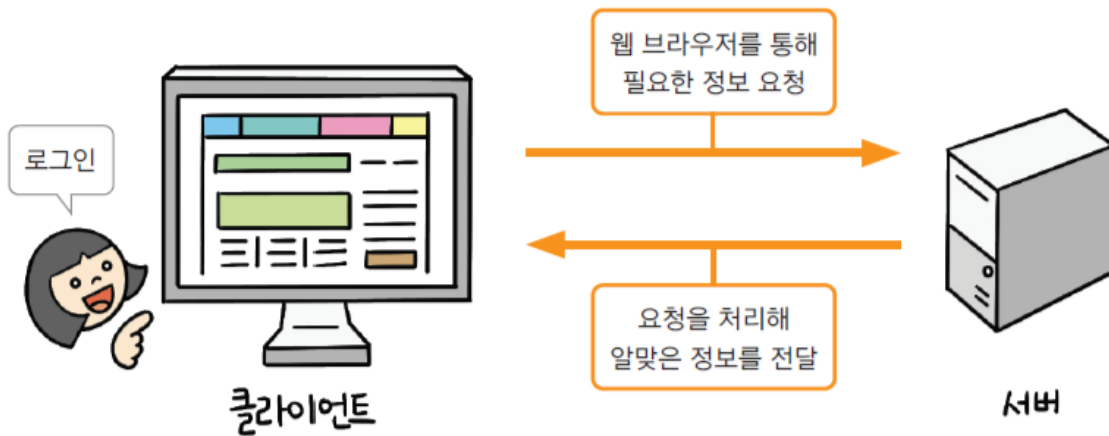
서버 vs 클라이언트

클라이언트

- 사용자가 웹 사이트에 접근할 때 사용하는 기기
- (좁은 의미) 웹 브라우저

서버

- 인터넷에 연결된 컴퓨터.
- 웹 요소와 여러 정보가 저장됨



프론트엔드 개발 vs 백엔드 개발

프론트엔드 개발

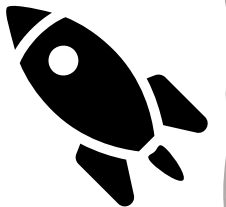
- 웹 브라우저 화면에 보이는 부분을 다룸 → 웹 사이트 제작
- HTML, CSS, 자바스크립트 사용

백엔드 개발

- 사용자 뒤(back)에서 보이지 않는 영역, 즉 서버를 다룸
- 데이터베이스를 설계하거나 데이터 처리
- 자바, PHP, 파이썬 등 프로그래밍 언어 사용

일론 머스크가 쏘아올린 세계 최초 민간 우주선인 Space X의 대시보드 인터페이스는 **크롬 기반의 웹 브라우저에 HTML과 CSS, Javascript**로 만들어졌다

[링크 보기](#)

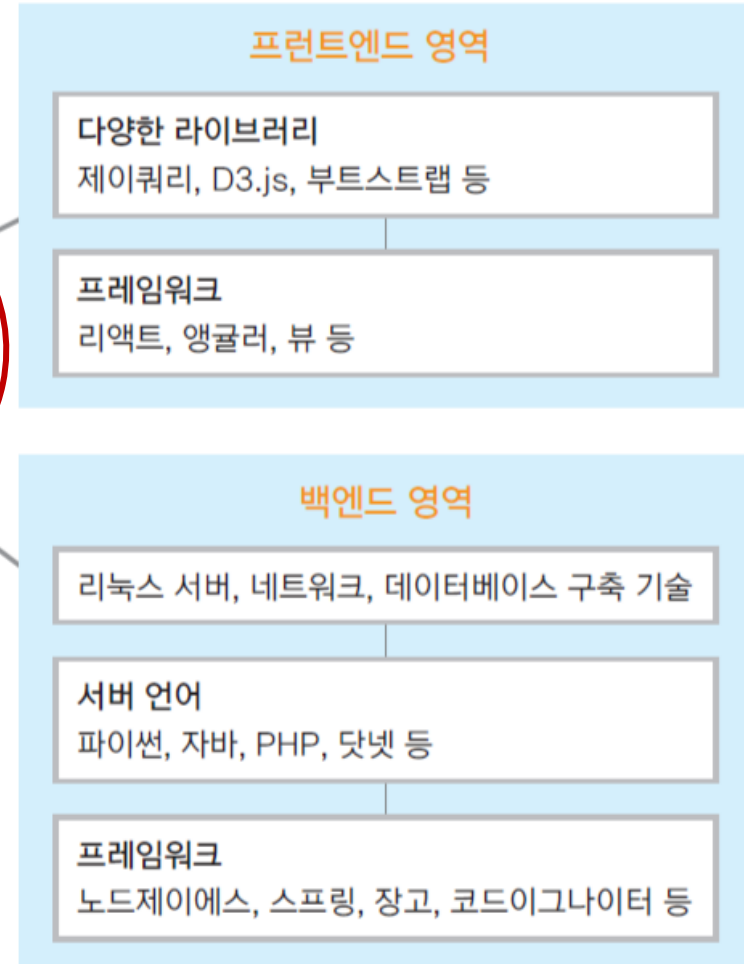


웹 개발, 어디서부터 시작할까

웹 개발을 위해 공부해야 할 기술



기본이면서 가장 중요



웹 개발, 어디서부터 시작할까

웹 개발의 기본



웹 문서의 뼈대를 만드는 HTML

- 웹 브라우저 창에 웹 문서의 내용을 보여주기 위한 약속
- HTML에서 약속한 표기법을 사용해서 문서 작성해야 함



웹 문서를 꾸미는 CSS

- 웹 문서를 꾸미거나 웹 요소를 적절하게 배치하는 방법
- 다양한 디바이스에 맞는 반응형 웹 디자인을 만들기 위해 필수적으로 학습해야 함









사용자 동작에 반응하는 자바스크립트

- 사용자 동작에 반응해서 동적인 효과를 만들기 위한 기술
- 자바스크립트를 알고 있다면 새로운 프레임워크를 배우기 쉬움

웹 개발, 어디서부터 시작할까

프론트엔드 개발을 위한 기술

표 1-1 자바스크립트의 라이브러리와 프레임워크

종류	특징
제이쿼리(jQuery) 	DOM을 활용해 웹 요소를 조작하기 쉽고 사용하기 편리한 라이브러리 도구입니다. 최근에는 웹 애플리케이션 개발에 적합한 다른 라이브러리나 프레임워크에 자리를 내주고 있습니다.
D3.js 	웹 브라우저에서 실시간으로 변하는 정보를 시각적으로 표시하기에 적합한 라이브러리 도구입니다. 데이터를 시각화해 주는 자바스크립트 라이브러리는 D3.js 외에도 라파엘(Raphaël), Three.js 등이 있습니다.
부트스트랩(Bootstrap) 	웹 사이트 디자인을 쉽게 만들어 주는 라이브러리 도구입니다. CSS로 PC, 태블릿, 휴대전화 등에서 동작하는 반응형 디자인을 만들 수 있게 해줍니다.
리액트(React) 	페이스북처럼 한 화면에서 모든 내용을 볼 수 있는 사이트를 SPA(single page application)이라고 합니다. SPA나 복잡한 사이트를 개발할 때 프레임워크를 사용합니다. 이때 가장 많이 사용하는 프레임워크가 리액트이며 페이스북에서 개발했습니다.
앵귤러(Angular) 	앵귤러는 구글에서 개발한 언어로 웹 애플리케이션을 제작하는 프레임워크입니다. 프로젝트 성격에 따라 리액트나 앵귤러 중에서 선택해서 사용합니다.
뷰(Vue) 	웹 애플리케이션에서 사용자 인터페이스를 만드는 프레임워크입니다. 화면에 보이는 부분에만 초점을 맞추기 때문에 다른 라이브러리나 프레임워크와 함께 사용할 수 있습니다.

백엔드 개발을 위한 기술

표 1-2 백엔드 개발의 언어와 프레임워크

종류	특징
노드제이एस(Node.js), 익스프레스(Express) 	이미 프론트엔드 개발을 해본 경험이 있거나 자바스크립트에 익숙하다면 Node.js로 실력을 발휘해 백엔드 개발에 사용할 수 있습니다. Node.js는 서버에서 자바스크립트를 실행할 수 있는 환경, 즉 플랫폼이라고 생각하면 됩니다. 그래서 자바스크립트만 익숙하면 웹 개발의 전체를 다룰 수 있습니다. Node.js에서 주로 사용하는 웹 개발 프레임워크는 익스프레스입니다.
자바(Java), 스프링(Spring) 	자바는 오랫동안 백엔드 개발용으로 사랑받는 언어이고, 그만큼 관련 커뮤니티나 오픈 소스가 많습니다. 자바를 사용하면 안드로이드 앱을 개발할 수도 있고, 셋톱 박스나 하드웨어용 애플리케이션을 만들 수도 있습니다. 자바의 웹 개발 프레임워크로는 스프링을 많이 사용합니다.
파이썬(Python), 장고(Django) 	파이썬은 배우기도, 사용하기기도 쉬워서 개발 입문자가 처음 공부할 때 선택하는 언어입니다. 특히 데이터를 많이 다뤄야 하는 분야에서는 파이썬이 훨씬 유리합니다. 백엔드 개발을 할 때 파이썬의 프레임워크로는 장고를 사용합니다.
PHP, 코드이그나이터(CodeIgniter) 	PHP는 백엔드 개발에서 오랫동안 사용된 개발 언어입니다. 이전 버전은 개인 웹 사이트에서 주로 사용했지만 PHP 7 버전부터는 백엔드 개발을 위한 프레임워크로 탈바꿈했습니다. PHP의 대표적인 웹 개발 프레임워크로 코드이그나이터가 있습니다.

자바스크립트로 무엇을 할까

웹 요소를 제어합니다

- 웹 요소를 가져와서 필요에 따라 스타일을 변경하거나 움직이게 할 수 있음
- 웹 사이트 UI 부분에 많이 활용
- 예) 마우스 포인터를 올렸을 때 펼쳐지는 메뉴
한 화면에서 탭을 눌러 내용만 바뀌도록 하는 콘텐츠

다양한 라이브러리를 사용할 수 있습니다

- 웹을 중심으로 하는 서비스가 늘어나면서 브라우저에서 처리해야 할 일이 늘어남 → 라이브러리와 프레임워크가 계속 등장
- 예) 시각화를 위한 d3.js, 머신러닝을 위한 tensorflow.js
DOM 조작을 위한 jQuery 등
- 예) 웹 애플리케이션 개발을 위한 React, Angular, Vue 등


웹 애플리케이션을 만듭니다

- 최근의 웹 사이트는 사용자와 실시간으로 정보를 주고 받으며 애플리케이션처럼 동작
- 예) 온라인 지도의 길찾기 서비스, 데이터 시각화 서비스
공개된 API를 활용한 다양한 서비스

서버를 구성하고 서버용 프로그램을 만들 수 있습니다

- node.js : 프론트엔드 개발에 사용하던 자바스크립트를 백엔드 개발에서 사용할 수 있게 만든 프레임워크

Javascript의 개발환경 구성

 **Visual Studio Code** Docs Updates Blog API Extensions FAQ Learn

Search Docs [Download](#)

Version 1.67 is now available! [Read about the new features and fixes from April.](#)

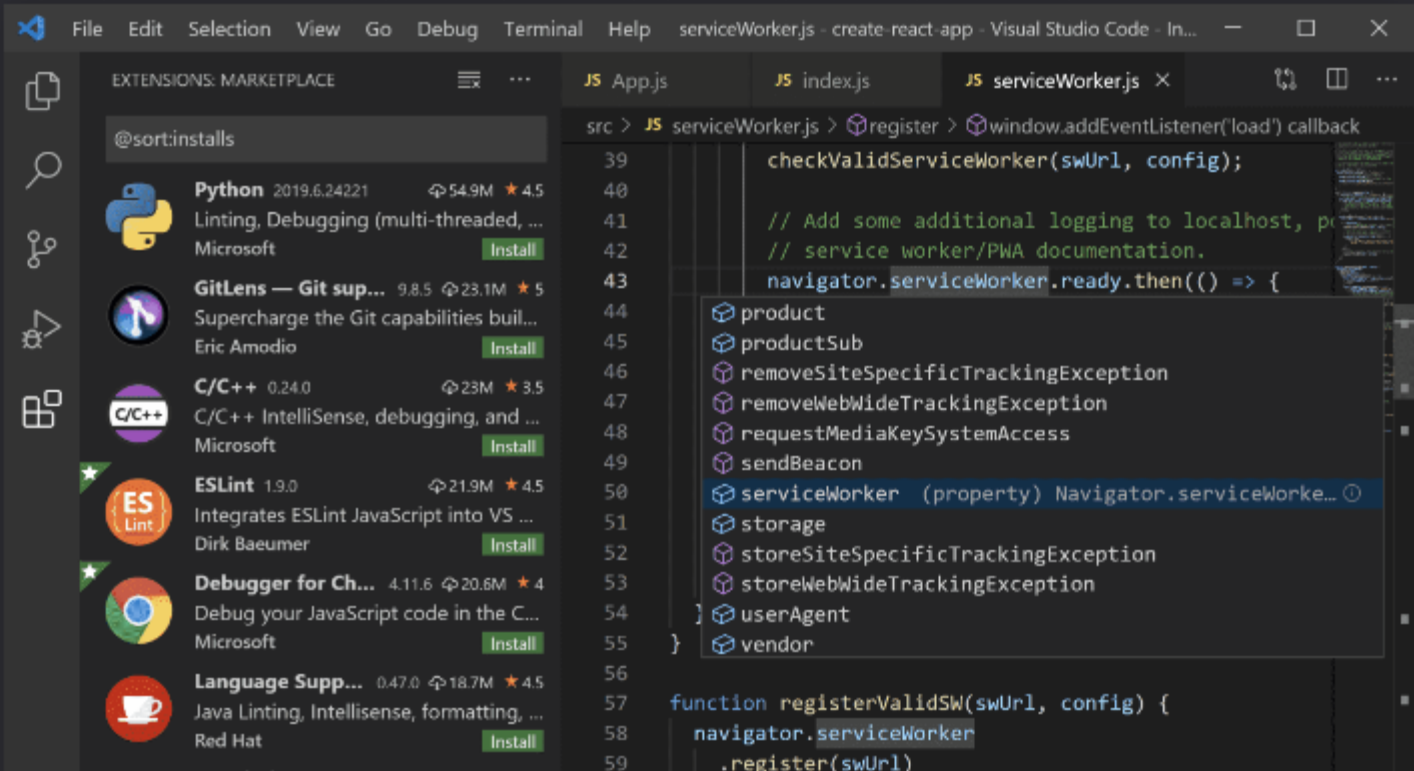
Code editing. Redefined.

Free. Built on open source. Runs everywhere.

[Download for Windows](#)
Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its [license and privacy statement.](#)

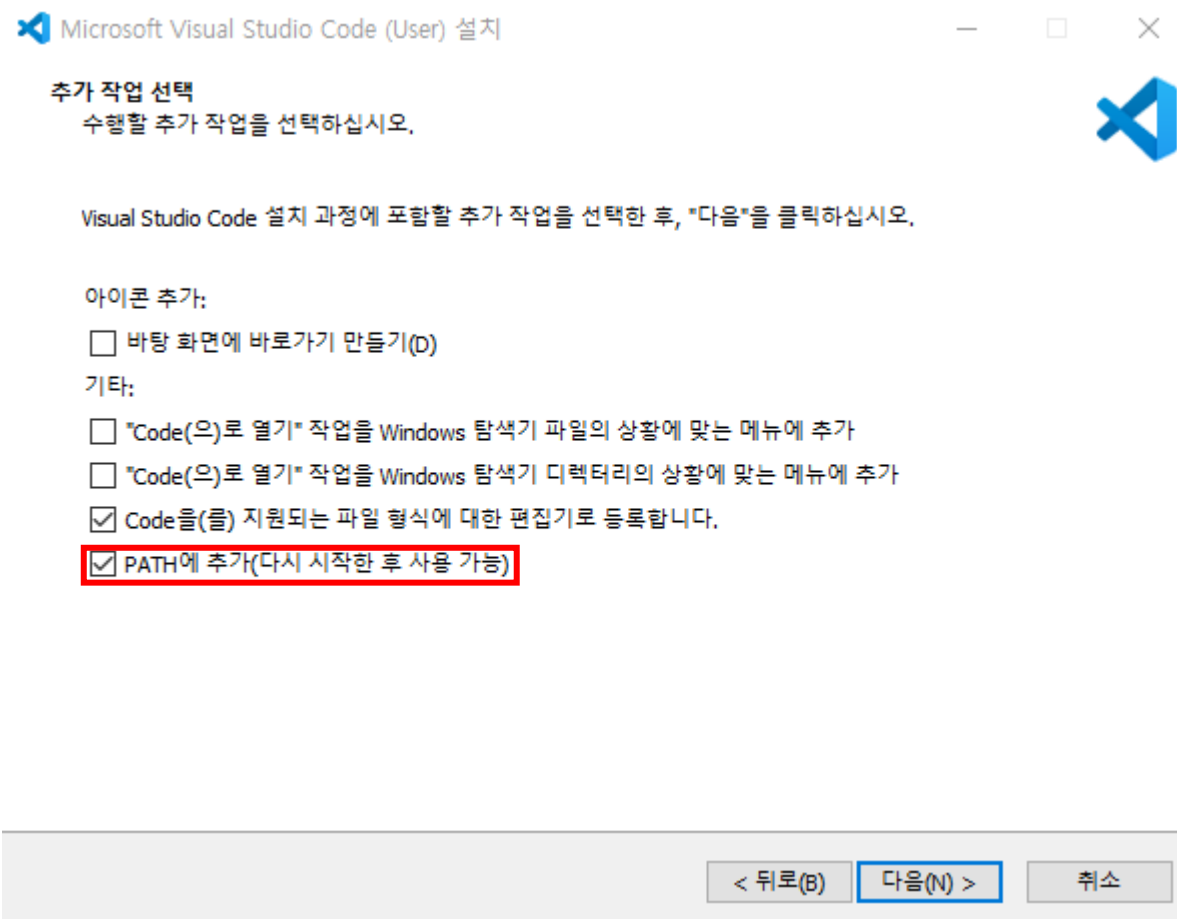


The screenshot shows the Visual Studio Code interface. On the left, the 'EXTENSIONS: MARKETPLACE' sidebar is open, displaying a list of installed and available extensions. The main editor area shows a JavaScript file named 'serviceWorker.js' with a code editor. A dropdown menu is visible over the code, listing various properties and methods of the 'navigator.serviceWorker' object, including 'product', 'productSub', 'removeSiteSpecificTrackingException', 'removeWebWideTrackingException', 'requestMediaKeySystemAccess', 'sendBeacon', 'serviceWorker', 'storage', 'storeSiteSpecificTrackingException', 'storeWebWideTrackingException', 'userAgent', and 'vendor'.

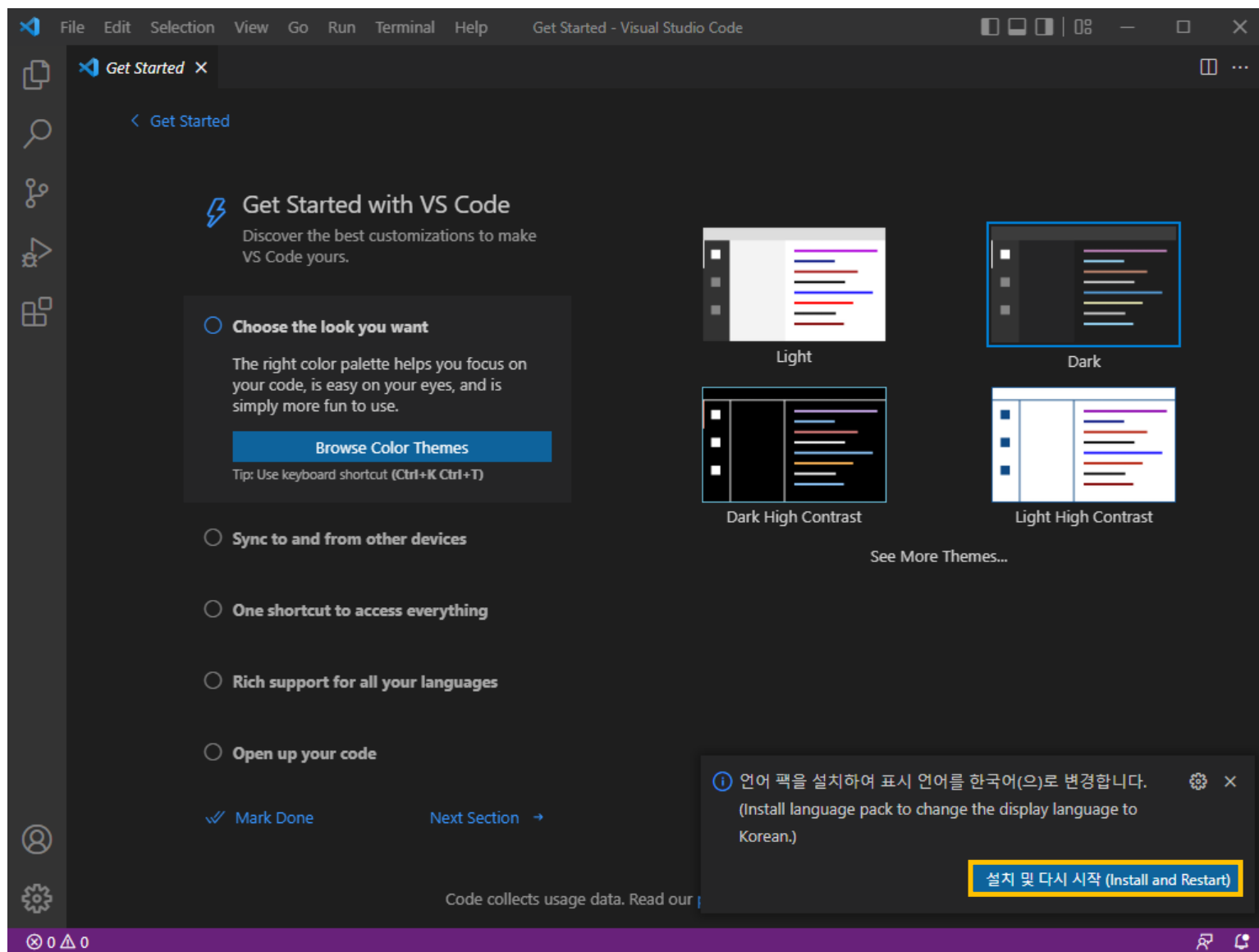
Extension Name	Version	Downloads	Rating	Author
Python	2019.6.24221	54.9M	4.5	Microsoft
GitLens — Git supercharger	9.8.5	23.1M	5	Eric Amodio
C/C++	0.24.0	23M	3.5	Microsoft
ESLint	1.9.0	21.9M	4.5	Dirk Baumer
Debugger for Chrome	4.11.6	20.6M	4	Microsoft
Language Support for Java	0.47.0	18.7M	4.5	Red Hat

Javascript의 개발환경 구성

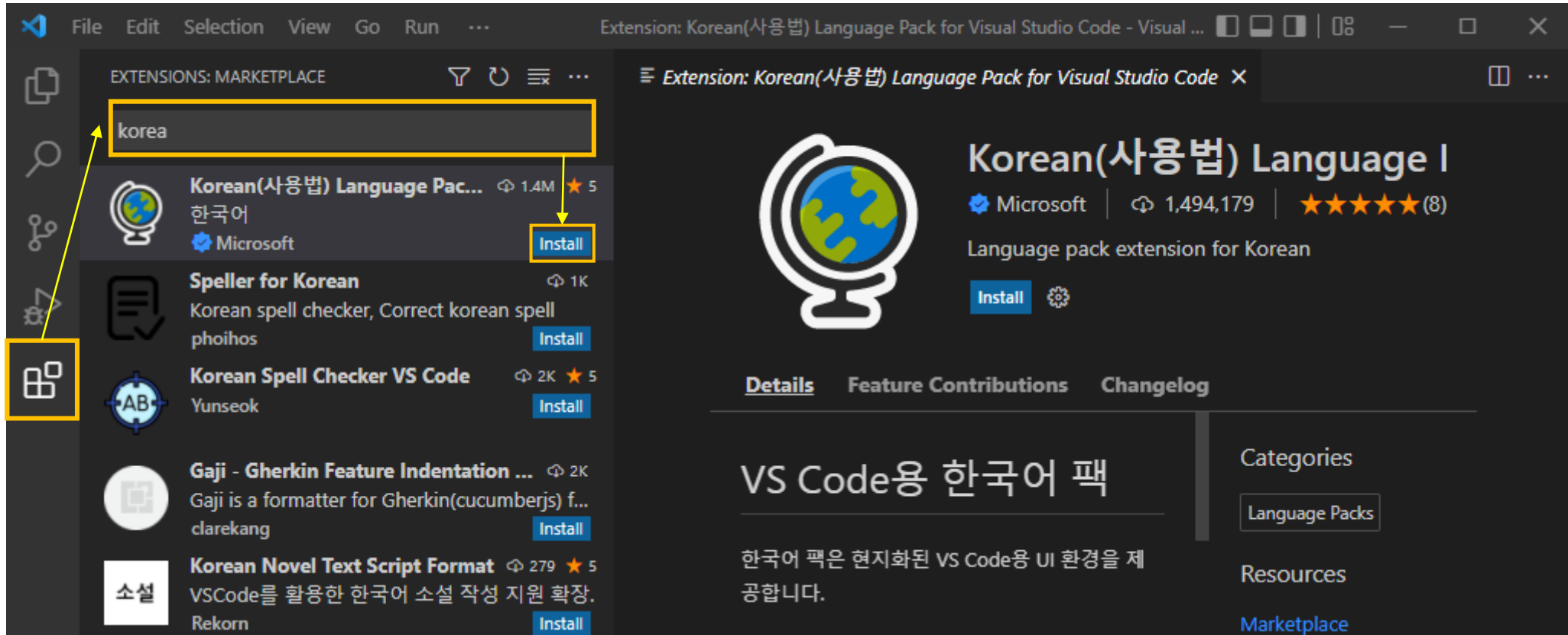
1. 다운 받은 파일을 실행
2. 설치디렉토리에 한글이 없는 것을 확인
3. PATH에 추가 체크된 것 확인



Javascript의 개발환경 구성

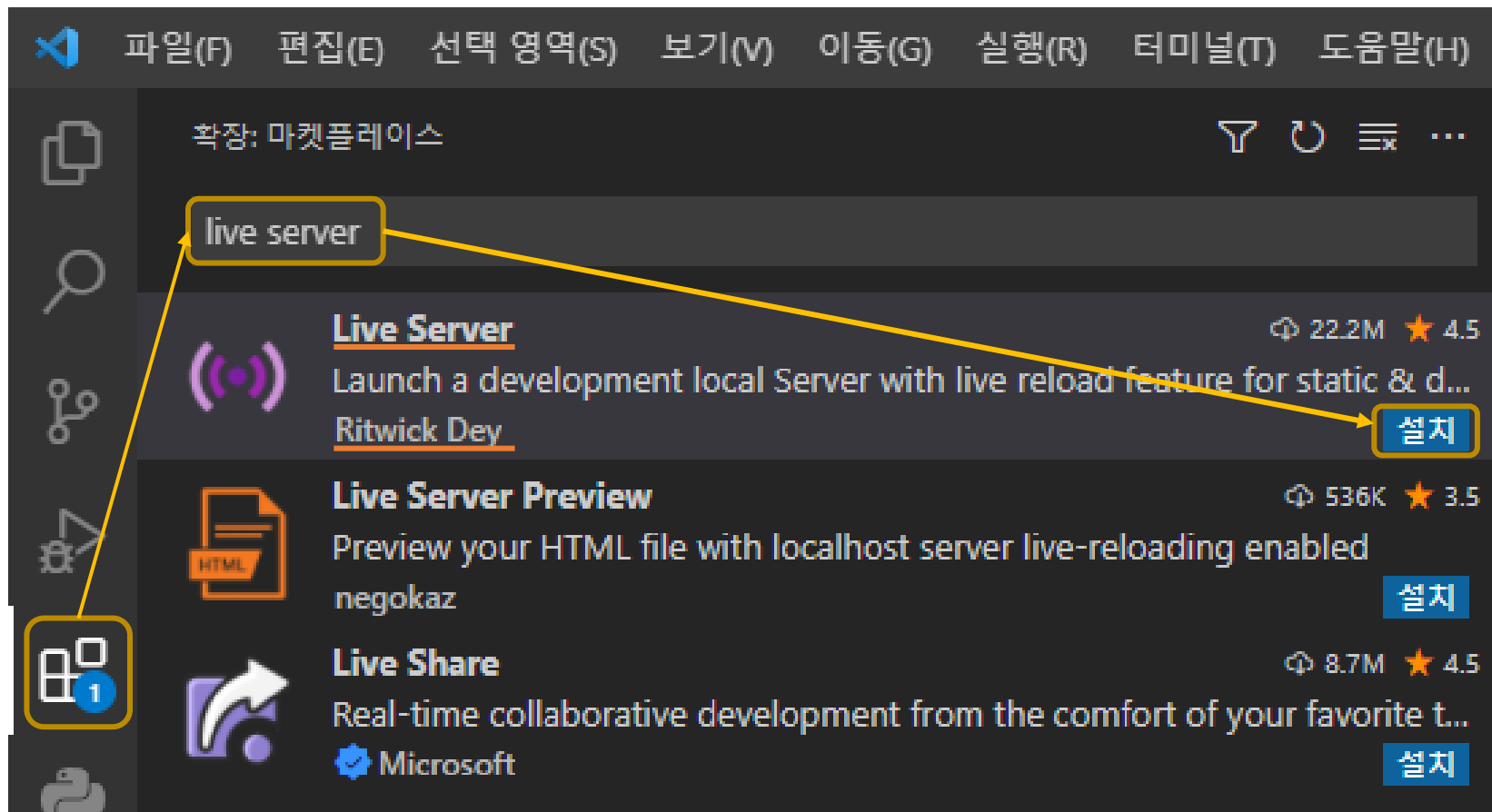


Javascript의 개발환경 구성



- 설치후 프로그램 재시작 해도 메뉴가 한글로 안바뀌는 경우
 - ① 메뉴에서 View > Command Palette 선택
 - ② Configure Display Language 검색후 목록에서 한글(KO) 선택

Javascript의 개발환경 구성



크롬브라우저로 오류확인하기

- 채팅창에 공유된 js_time.html를 크롬브라우저로 열어보세요
- 화면에 아무것도 출력되지 않습니다.
- ctrl+shift+j 또는 i를 눌러보세요
- 콘솔창에 빨간색 글자로 오류내용이 표시 되고 오류개수도 표시될 것입니다.
- 오류내용을 visual studio code를 사용하여 수정하고 저장한후 새로고침을 해보시면
- 정상작동하는 것을 확인할 수 있습니다.


[좀 더 알아보기①] 오류를 확인하는 방법

- 내가 무엇을 잘못 입력했는지 알아내는 방법과 찾는 방법

```
<script>
  alrt('Hello World')
</script>
```



alert를 alrt로 잘못 입력했다고 가정

- 01: 현재 상태에서 코드를 실행해보면 아무 것도 출력되지 않음
- 02: 크롬에서 코드를 실행한 후 마우스 오른쪽 버튼을 클릭해 [검사]를 선택
- 03: 개발자 도구 오른쪽 위에 × 표시가 되어 있는 붉은색 원  (자바스크립트 코드 등에 오류가 발생했을 때 출력되는 아이콘) 아이콘을 클릭하거나 개발자 도구의 [Console] 탭을 클릭
- 04: 'Uncaught ReferenceError: alrt is not defined'라는 오류 출력, 어떤 오류인지 확인. test.html : 6은 오류가 발생한 위치. [test.html : 6]을 클릭하면 오류가 발생한 위치로 이동
- 05: 붉은색 밑줄이 표시되어 있어 쉽게 오류를 찾을 수 있음

- 처음 자바스크립트를 공부할 때 자주 접하는 오류
 - ReferenceError: 예외 처리
 - SyntaxError: 구문 오류

SECTION 1-3 알아두어야 할 기본 용어(2)

- 표현식과 문장
 - 표현식: 자바스크립트에서 값을 만들어내는 간단한 코드
 - 문장: 하나 이상의 표현식이 모여 문장(statement)을 구성. 문장 끝에는 마침표를 찍듯이 세미콜론(;) 또는 줄바꿈을 넣어서 문장의 종결을 나타냄
 - 프로그램: 줄바꿈으로 문장을 구분해 코드를 작성
- 키워드: 자바스크립트가 처음 만들어질 때 정해놓은 특별한 의미가 있는 단어

await	break	case	catch
class	const	continue	debugger
default	delete	do	else
export	extends	finally	for
function	if	import	in
instanceof	new	return	super
switch	this	throw	try
typeof	var	void	while
with	yield	let	static

SECTION 1-3 알아두어야 할 기본 용어(3)

- 식별자: 프로그래밍 언어에서 이름을 붙일 때 사용하는 단어. 주로 변수명이나 함수명 등으로 사용
 - 키워드를 사용 안됨
 - 숫자로 시작 불가
 - 특수 문자는 _와 \$만 허용
 - 공백 문자를 포함할 수 없음
- 식별자를 만드는 일반적인 관례
 - 클래스(Chapter 9-1 참조)의 이름은 항상 대문자로 시작
 - 변수(Chapter 2-2 참조)와 인스턴스(Chapter 09-1 참조), 함수(Chapter 05-1 참조), 메소드(Chapter 06-1 참조)의 이름은 항상 소문자로 시작
 - 여러 단어로 이루어진 식별자는 각 단어의 첫 글자를 대문자
- 식별자의 종류

구분	단독으로 사용	다른 식별자와 사용
식별자 뒤에 괄호 없음	변수	속성
식별자 뒤에 괄호 있음	함수	메소드

SECTION 1-3 알아두어야 할 기본 용어(4)

- 주석: 프로그램 코드를 설명할 때 사용하며 프로그램 진행에는 전혀 영향을 주지 않음
 - HTML 태그 주석: <!-- -->로 문자열을 감싸 생성
 - 자바스크립트 주석
 - [방법1] //를 입력하는 것으로 한 줄 주석을 표현(// 뒤의 문장은 실행되지 않음)
 - [방법2] /*와 */를 입력하여 여러 줄 주석을 표현(/*와 */ 사이에 있는 모든 문장은 실행되지 않음)

```
<script>  
// 주석은 코드 실행에 아무 영향을 미치지 않습니다.  
/*  
alert('Hello JavaScript')  
alert('Hello JavaScript')  
alert('Hello JavaScript')  
*/  
</script>
```

SECTION 1-3 알아두어야 할 기본 용어(5)

- 출력: 자바스크립트는 다른 프로그래밍 언어와 비교해서 출력 방법이 많고 복잡한 편
 - 간단한 표현식 결과 확인하기
 - 01: 구글 크롬의 주소창에 about:blank를 입력해 빈 페이지로 들어가 단축키 Ctrl + Shift+ I (알파벳 '아이')를 눌러서 개발자 환경을 띄우기
 - 02: about:blank에서 [Console] 탭을 클릭해 구글 크롬 개발자 도구에 진입. 이곳에 어떤 값을 입력하면 곧바로 그 결과가 출력
 - 경고창에 출력하기
 - 개발 전용 에디터를 사용할 때의 출력하는 방법
 - alert() 함수를 사용하여 웹 브라우저에 경고창을 띄우기
 - 콘솔에 출력하기
 - console.log() 메소드 사용

```
<script>  
alert('Hello JavaScript...!')  
</script>
```

```
<script>  
console.log('Hello JavaScript...!')  
</script>
```

자바스크립트 스타일 가이드

스타일 가이드란

코딩 규칙을 '스타일 가이드', '코딩 컨벤션', '코딩 스타일', '표준 스타일' 등으로 부름

코딩 규칙이 왜 필요할까?

- 자바스크립트는 다른 프로그래밍 언어에 비해 데이터 유형이 유연해서 오류 발생이 잦다
- 오픈 소스에 기여하거나 누군가와 공유할 소스라면 더욱 깔끔한 소스가 중요하다
- 팀 프로젝트를 진행한다면 통일된 코딩 규칙이 필요하다
- 코딩 규칙에 따라 작성된 웹 사이트는 유지 보수도 수월하고 그만큼 비용도 줄어든다

자바스크립트 스타일 가이드

구글 자바스크립트 스타일 가이드 (google.github.io/styleguide/jsguide.html) 또는

에어비앤비 자바스크립트 스타일 가이드(github.com/airbnb/javascript) 참고

회사 프로젝트의 경우 팀 내에서 상의해서 결정

Google JavaScript Style Guide	
Table of Contents	
1 Introduction	5.8 Control structures
1.1 Terminology notes	5.9 this
1.2 Guide notes	5.10 Equality Checks
	5.11 Disallowed features
2 Source file basics	6 Naming
2.1 File name	6.1 Rules common to all identifiers
2.2 File encoding: UTF-8	6.2 Rules by identifier type
2.3 Special characters	6.3 Camel case: defined
3 Source file structure	7 JSDoc
3.1 License or copyright information, if present	7.1 General form
3.2 @fileoverview, JSDoc, if present	7.2 Markdown
3.3 goog.module, statement	7.3 JSDoc tags
3.3.3 goog.module Exports	7.4 Line wrapping
3.4 ES modules	7.5 Top-level comments
3.5 goog.setTestOnly	7.6 Class comments
3.6 goog.require and goog.requireType statements	7.7 Enum and typedef comments
3.7 The file's implementation	7.8 Method and function comments
4 Formatting	7.9 Property comments
4.1 Braces	7.10 Type annotations
4.2 Block indentation: +2 spaces	7.11 Visibility annotations
4.3 Statements	
4.4 Column limit: 80	8 Policies
4.5 Line wrapping	8.1 Issues unspecified by Google Style: Be Consistent!

Airbnb JavaScript Style Guide() {
<i>A mostly reasonable approach to JavaScript</i>
Note: this guide assumes you are using Babel , and requires that you use babel-preset-airbnb or the equivalent. It also assumes you are installing shims/polyfills in your app, with airbnb-browser-shims or the equivalent.
downloads: 6.2M/month downloads: 12M/month gitter join chat
This guide is available in other languages too. See Translation
Other Style Guides
<ul style="list-style-type: none">• ES5 (Deprecated)• React• CSS-in-JavaScript• CSS & Sass• Ruby
Table of Contents
1. Types
2. References
3. Objects
4. Arrays
5. Destructuring
6. Strings
7. Functions

자바스크립트 스타일 가이드

자바스크립트 소스를 작성할 때 지켜야 할 기본 규칙

1. 코드를 보기 좋게 들여쓴다

- 'Tab' 키나 '스페이스바'를 눌러 2칸이나 4칸 들여쓰
- 최근에는 공백 2칸 들여쓰기를 많이 사용함

2. 세미콜론으로 문장을 구분한다

- 세미콜론을 붙일 것을 권장함
- 소스는 한 줄에 한 문장만 작성하는 것이 좋다

세미콜론 규칙

```
var n = 10 // 권장하지 않음
var n = 10; // 권장함
var n = 10; var sum = 0; // 권장하지 않음
```

3. 공백을 넣어 읽기 쉽게 작성한다

식별자나 연산자, 값 사이에 공백을 넣어 읽기 쉽게 작성한다

공백 규칙

```
var sum=num+10; // 권장하지 않음
var sum = num + 10; // 권장함
```

4. 코드를 설명하는 주석을 작성한다

- 한 줄 주석 : 슬래시 2개(//) 바로 뒤에 작성
- 여러 줄 주석 : 여는 기호(/*)를 맨 앞에, 닫는 기호(*/)를 맨 뒤에 넣고 그 사이에 주석 내용을 작성
- 주석 사이에 또다른 주석을 넣을 수 없음



한 줄 주석 작성 방법

```
var today = new Date(); // 현재 날짜 가져오기
var h = today.getHours(); // 시간 추출하기
```



여러 줄 주석 작성 방법

```
/* 현재 날짜를 가져와
시와 분, 초로 추출하고
화면에 표시하는 스크립트
*/
function startTime() { ..... }
```

5. 식별자는 정해진 규칙을 지켜 작성한다

- 첫 글자는 반드시 영문자나 언더스코어(_), 달러 기호(\$)로 시작해야 한다
- 두 단어 이상이 하나의 식별자를 만들 때 단어 사이에 공백을 둘 수 없다
- 예약어는 식별자로 사용할 수 없다

자바스크립트 스타일 가이드

자바스크립트 소스를 작성할 때 지켜야 할 기본 규칙

6. 중괄호의 위치

```
for ( let l = 0; l < length; i++) {  
  const element = array[i]  
}
```

```
for ( let l = 0; l < length; i++)  
{  
  const element = array[i]  
}
```

자료형 이해하기

자료형이란

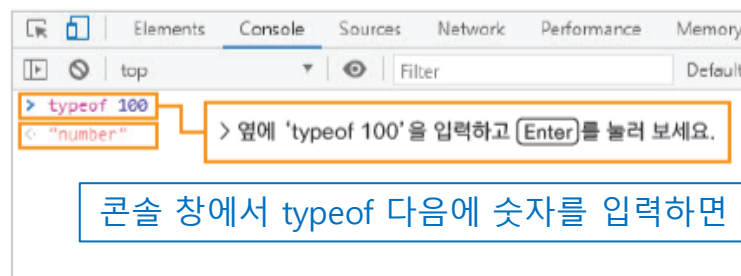
컴퓨터가 처리할 수 있는 자료의 형태

종류		설명	예시
기본 유형	숫자형	따옴표 없이 숫자로만 표기합니다.	var birthYear = 2000;
	문자열	작은따옴표(' ')나 큰따옴표(" ")로 묶어서 나타냅니다. 숫자를 따옴표로 묶으면 문자로 인식합니다.	var greeting = "Hello!"; var birthYear = "2000";
	논리형	참(true)과 거짓(false)이라는 2가지 값만 있는 유형입니다. 이때 true와 false는 소문자로만 표시합니다.	var isEmpty = true;
복합 유형	배열	하나의 변수에 여러 개의 값을 저장합니다.	var seasons = ['봄', '여름', '가을', '겨울'];
	객체	함수와 속성을 함께 포함합니다.	var date = new Date();
특수 유형	undefined	자료형이 지정되지 않았을 때의 상태입니다. 예를 들어 변수 선언만 하고 값을 할당하지 않은 변수는 undefined 상태입니다.	
	null	값이 유효하지 않을 때의 상태입니다.	

숫자형(number)

숫자

- 정수 : 소수점 없는 숫자
 - 실수 : 소수점이 있는 숫자
- ※ 자바스크립트는 실수를 정밀하게 계산하지 못함



콘솔 창에서 typeof 다음에 숫자를 입력하면 number라고 표시됨

자료형 이해하기

문자열(string)

작은따옴표(' ')나 큰따옴표(" ")로 묶은 데이터

```
top Filter Default levels
> typeof "안녕하세요?"
< "string"
> typeof "12345"
< "string"
> |
```

콘솔 창에서 typeof 다음에 따옴표로 묶은 내용을 입력하면 string이라고 표시됨

논리형(boolean)

- 참true이나 거짓false의 값을 표현하는 자료형. 불린 유형이라고도 함.
- 조건을 확인해서 조건이 맞으면 true, 맞지 않으면 false라는 결과값 출력

```
top Filter Default levels
> 100 < 10
< false
> 100 > 10
< true
> |
```

조건에 따라 true나 false 값을 표시하는 논리형

undefined 유형

- 자료형이 정의되지 않았을 때의 데이터 상태
- 변수 선언만 하고 값이 할당되지 않은 자료형

null 유형

- 데이터 값이 유효하지 않은 상태
- 변수에 할당된 값이 유효하지 않다는 의미

자료형 이해하기

배열(array)

하나의 변수에 여러 값을 저장할 수 있는 복합 유형

기본형 배열명["값1 ", "값2",]

배열명[]

빈 배열 선언

예) 계절 이름을 프로그램에 사용할 경우

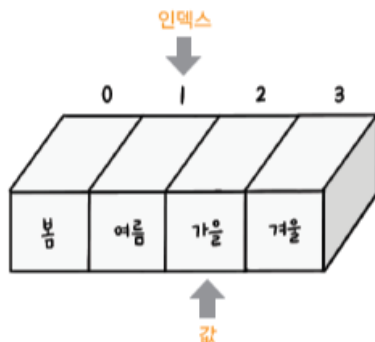
배열을 사용하지 않는다면

```
var spring = "봄";  
var summer = "여름";  
var fall = "가을";  
var winter = "겨울";
```

배열을 사용한다면

```
var season = ["봄", "여름", "가을", "겨울"];
```

변수 이름 → Season



자바스크립트의 데이터 유형 자동 변환

자바스크립트의 편리한 점이면서도 약점인 부분이 데이터 유형이 유연하다는 것입니다. 다시 말해 변수의 데이터 유형이 중간에 바뀔 수 있다는 것이죠.

책에 있는 '나이 계산 프로그램'에서는 프롬프트 창을 통해 사용자의 태어난 해를 입력받는데, 이때 프롬프트 창에서 입력받은 값은 문자열이지만 사칙연산에 사용된 문자열은 자동으로 숫자형으로 변환되어 계산됩니다