개발환경 설치

2020-01-19

이승진

학습목표

node.js 개발환경 설치

hello world 출력

목차

[1. node.js 개발환경 2](#_Toc32748976)

[1) node.js 설치 2](#_Toc32748977)

[2) visual studio code 설치 2](#_Toc32748978)

[3) Git 클라이언트 설치 3](#_Toc32748979)

[2. 관련 개념 5](#_Toc32748980)

[1) 컴파일러(compiler), 인터프리터(interpreter) 5](#_Toc32748981)

[2) node.js 5](#_Toc32748982)

[3) 터미널(terminal) 6](#_Toc32748983)

[3. hello world 7](#_Toc32748984)

[1) 소스코드 폴더 생성 7](#_Toc32748985)

[2) 소스코드 파일 생성 8](#_Toc32748986)

[3) 터미널 열기 9](#_Toc32748987)

[4) javascript 파일 실행 12](#_Toc32748988)

[5) let 키워드로 변수 선언 13](#_Toc32748989)

[6) var 키워드로 변수 선언 14](#_Toc32748990)

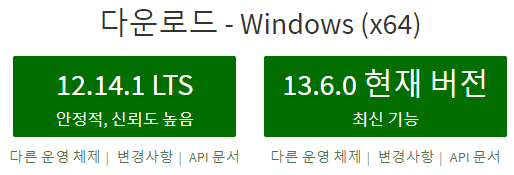
[7) 세미콜론 생략 가능 14](#_Toc32748991)

[4. 내용 정리 15](#_Toc32748992)

# node.js 개발환경

## node.js 설치

<https://nodejs.org/ko/>



LTS 버전을 내려받자.

|  |
| --- |
| **참고:**  **LTS (Long Term Support)**  버그 패치 및 보안 패치를 오랜 기간 지원하는 버전.  너무 최신 버전은 호환성 문제가 있을 수 있으므로, LTS 버전을 다운받는 것이 좋다.  Java의 경우 LTS 버전은 Java 8, Java 11 이고, Java 9, Java 10 은 아니다.  그래서 Java 9, Java 10 보다 Java 8을 사용하는 프로젝트가 많다. |

내려받은 설치파일을 실행해서 설치하자.

기본 설치 옵션으로 설치하면 된다.

## visual studio code 설치

### 설치

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



내려받은 설치파일을 실행해서 설치하자.

기본 설치 옵션으로 설치하면 된다.

### 사용법

<https://demun.github.io/vscode-tutorial/>

## Git 클라이언트 설치

<https://git-scm.com/downloads>

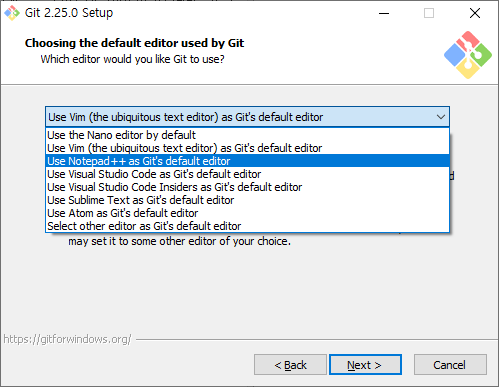


Download 버튼 클릭

설치 파일을 실행하여 설치하자.

아래 두 항목 이외에는 디폴트 값을 선택하면 된다.

### 텍스트 에디터 선택



Git에서 사용할 텍스트 에디터를 선택하는 화면이다.

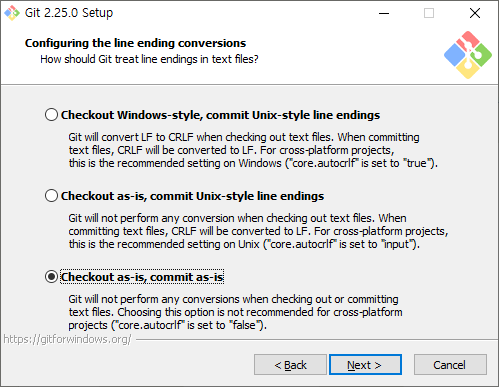
Vim 에디터가 디폴트로 선택되어 있을 텐데, 이 에디터는 리눅스의 기본 에디터인데,

사용법이 복잡하기 때문에 따로 공부해야 한다.

자신에게 익숙한 텍스트 에디터를 선택하자.

Notepad++ 에디터를 추천한다. (<https://notepad-plus-plus.org/>)

### 줄바꿈 문자 처리 선택



checkout as-is, commit as-is 항목을 선태하자.

|  |
| --- |
| **참고**  Linux의 줄바꿈 문자는 1 문자이다. (\n)  Windows의 줄바꿈 문자는 2 문자이다. (\r\n)  좋은 텍스트 편집기는 Linux 줄바꿈 문자, Windows 줄바꿈 문자를 둘 다 잘 인식해서 처리하지만,  그렇지 않은 텍스트 편집기도 있다.  checkout Windows-style, commit Unix-style line endings  Git 저장소에서 꺼낼 때는, Windows 줄바꿈 문자로 변환하고  Git 저장소에 저장할 때는, Unix 줄바꿈 문자로 변환한다.  checkout as-is, commit Unix-style line endings  Git 저장소에서 꺼낼 때는, 변환하지 않는다  Git 저장소에 저장할 때는, Unix 줄바꿈 문자로 변환한다.  checkout as-is, commit as-is  Git 저장소에서 꺼낼 때 변환하지 않고  Git 저장소에 저장할 때도 변환하지 않는다. |

# 관련 개념

## 컴파일러(compiler), 인터프리터(interpreter)

C는 컴파일 방식의 프로그래밍 언어이다.

컴파일러는 소스 코드를 실행 파일로 변환한다.

실행 파일만 있으면 실행이 되기 때문에, 소스 코드를 배포할 필요 없다.

javascript 언어는 인터프리터 방식의 프로그래밍 언어이다.

인터프리터는 소스 코드를 실행 파일로 변환하지 않는다.

인터프리터는 소스 코드를 한 줄 씩 읽으며 실행하기 때문에,

실행할 때 마다 소스 코드 파일이 필요하다.

## node.js

javascript 언어는 인터프리터(interpreter) 방식으로 실행된다.

javascript 인터프리터가 javascript 소스코드를 읽어서 실행한다.

javascript 인터프리터는 웹브라우저에 내장되어 있다.

그렇기 때문에 웹브라우저에서 javascript 소스 코드가 실행될 수 있다.

javascript 소스 코드를 서버에서 실행하기 위한 엔진이 node.js 이다.

당연히 node.js 에도 javascript 인터프리터가 내장되어 있다.

가장 유명한 javascript 인터프리터는

구글이 만든 chrome v8 javascript 인터프리터이다.

이 javascript 인터프리터가 chrome 웹브라우저에 내장되어 있고,

node.js 에도 내장되어 있다.

node.js는, javascript로 구현된 코드를 서버에서 실행하기 위한 플랫폼이다.

플랫폼 = 실행 엔진(인터프리터) + 라이브러리 + 개발도구

node.js 플랫폼에서 실행되는 웹 서버, REST API 서버, 채팅 서버를

javascript 언어로 어렵지 않게 구현할 수 있다.

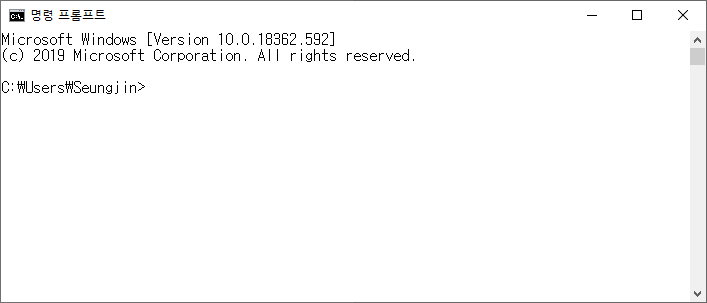
## 터미널(terminal)

Windows에서 명령 프롬프트(command prompt) 창에 해당하는 것을

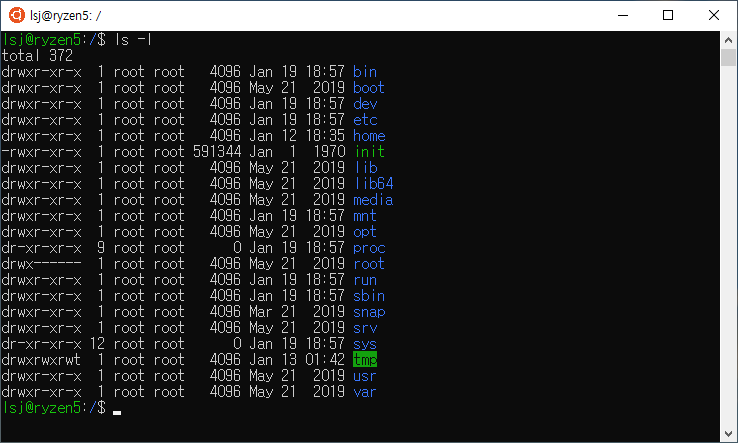
Linux 에서는 터미널(terminal) 이라고 부른다.

visual studio code 에서도 이 창을 터미널이라고 부른다.

### Windows의 명령 프롬프트 창



### Linux의 터미널 창



### 쉘(shell)

**명령 프롬프트 창**에서 입력된 명령을 실행하는 프로그램을 쉘(shell)이라고 한다.

Windows의 기본 쉘은 cmd.exe 이다.

Windows의 고급 쉘은 powershell.exe 이다.

Linux의 기본 쉘은 bash 이다.

Windows 에서도 bash 쉘을 설치하여 사용할 수 있다.

Git 클라이언트를 다운로드하여 설치하면,

Windows 용 bash 쉘이 설치된다.

# hello world

## 소스코드 폴더 생성

소스코드들을 저장할 적당한 폴더를 만들자

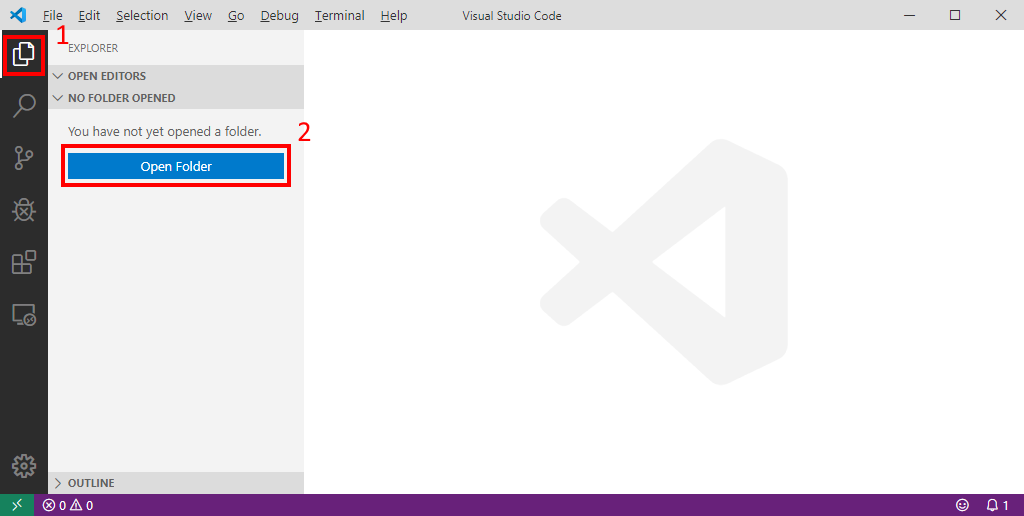
**예: c:/PJ/node1**

visual studio code에서 소스코드 폴더를 열자

### 소스코드 폴더 열기 방법 #1

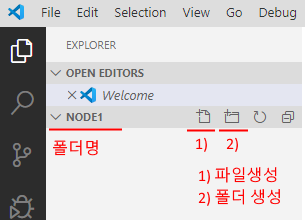
메뉴: File - Open Folder

### 소스코드 폴더 열기 방법 #2



## 소스코드 파일 생성

### 소스코드 파일 생성



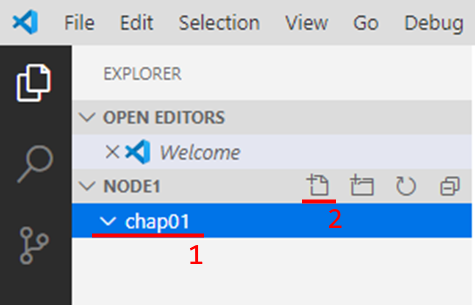
소스코드를 저장할 폴더(예: node1)을 선택하고,

파일 생성(New File) 버튼을 클릭하자.

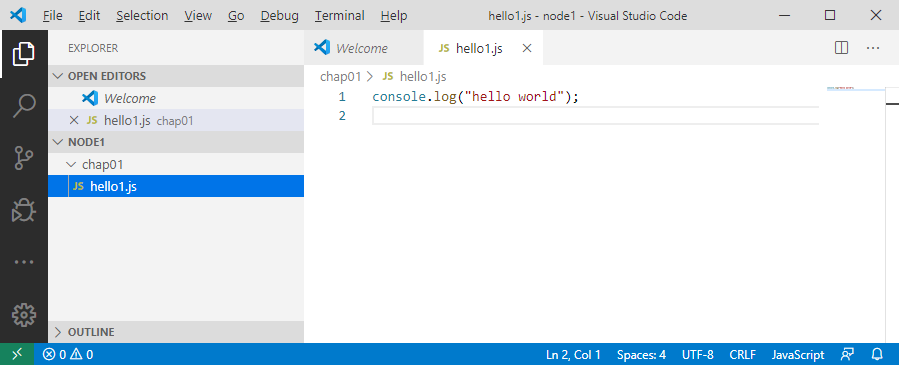
### 폴더 생성

위 화면에서 폴더 생성 버튼을 클릭하여, node1 폴더 아래에 chap01 폴더를 만들자.

### 파일 생성



chap01 폴더를 선택하고, 파일 생성 버튼을 클릭하여, chap01 폴더 아래에 hello1.js 파일을 만들자.



### hello1.js

|  |
| --- |
| console.log("hello world"); |

console 객체의 log 메소드는, 화면(console)에 값을 출력한다.

Java의 System.out.println 메소드와 기능이 같다.

javascript 소스 코드에는 main 함수가 필요없다.

javascript 소스코드 파일을 실행하면, 파일의 소스 코드들이 한 줄씩 순서대로 실행된다.

### 파일 저장

단축키 Ctrl+S

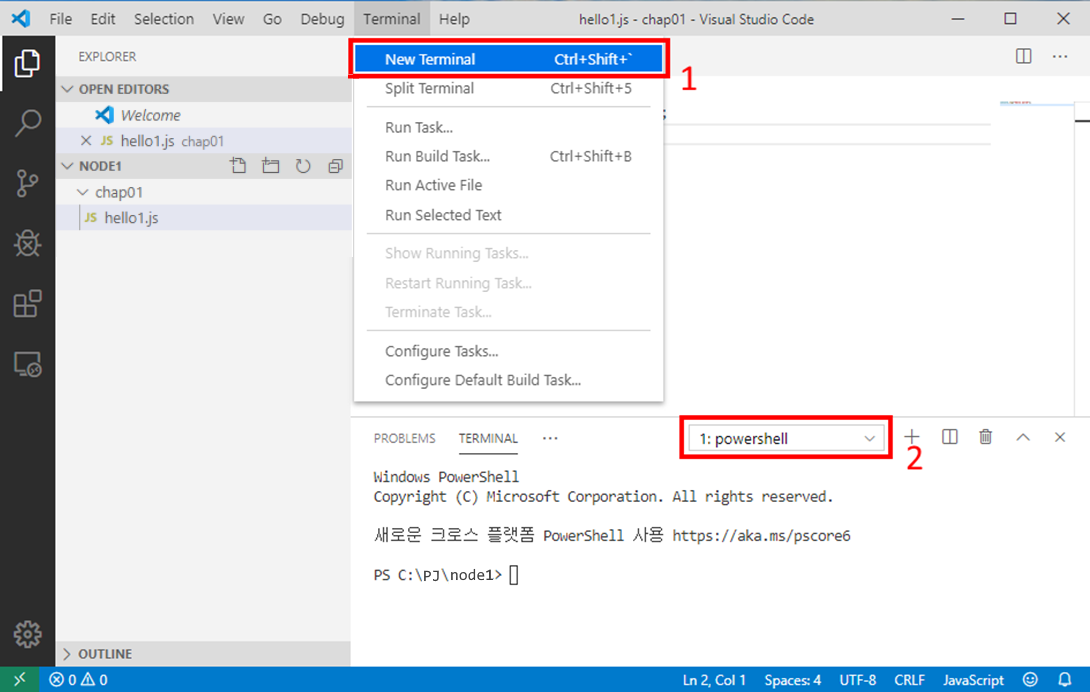
## 터미널 열기

node.js는 기본적으로 명령 프롬프트 창의 쉘(shell)에서 실행된다.

visual studio code 에서 명령 프롬프트 창을 열자. (터미널을 열자)

### 터미널 열기

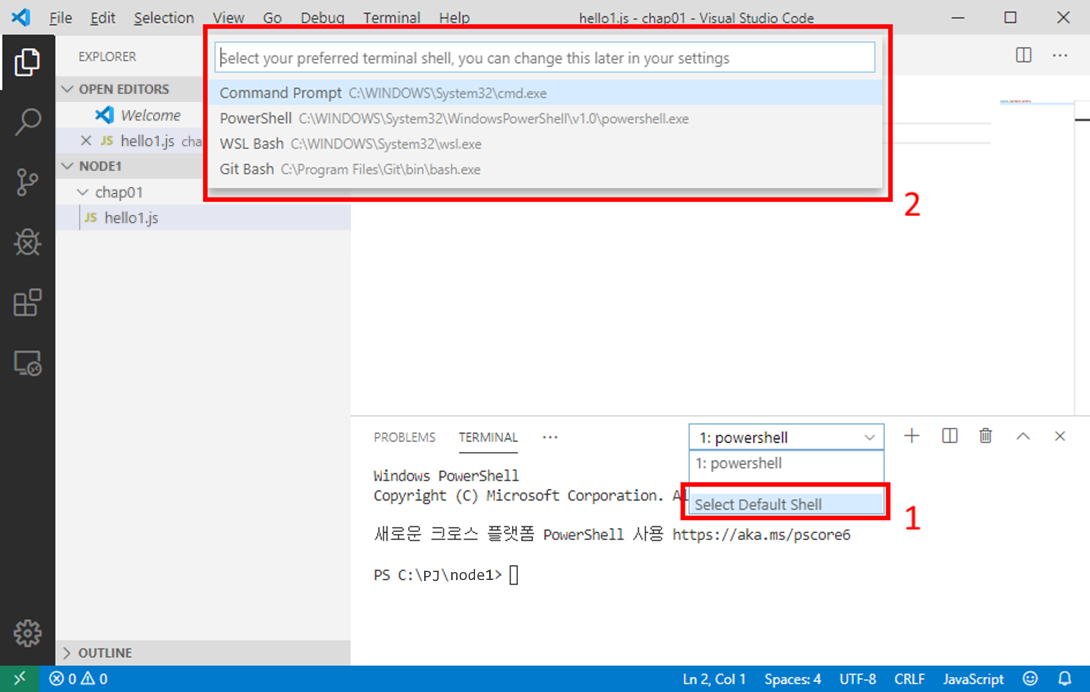
메뉴: Terminal - New Terminal



위 화면의 (2)를 클릭하여, 터미널에서 사용할 쉘(shell)을 선택하자.

### 디폴트 쉘 선택

터미널에서 사용할 쉘을 선택하자.



(1) Select Default Shell 클릭

(2) 선택 가능한 쉘 목록

Command Prompt : Windows의 기본 쉘

PowerShell : Windows의 고급 쉘

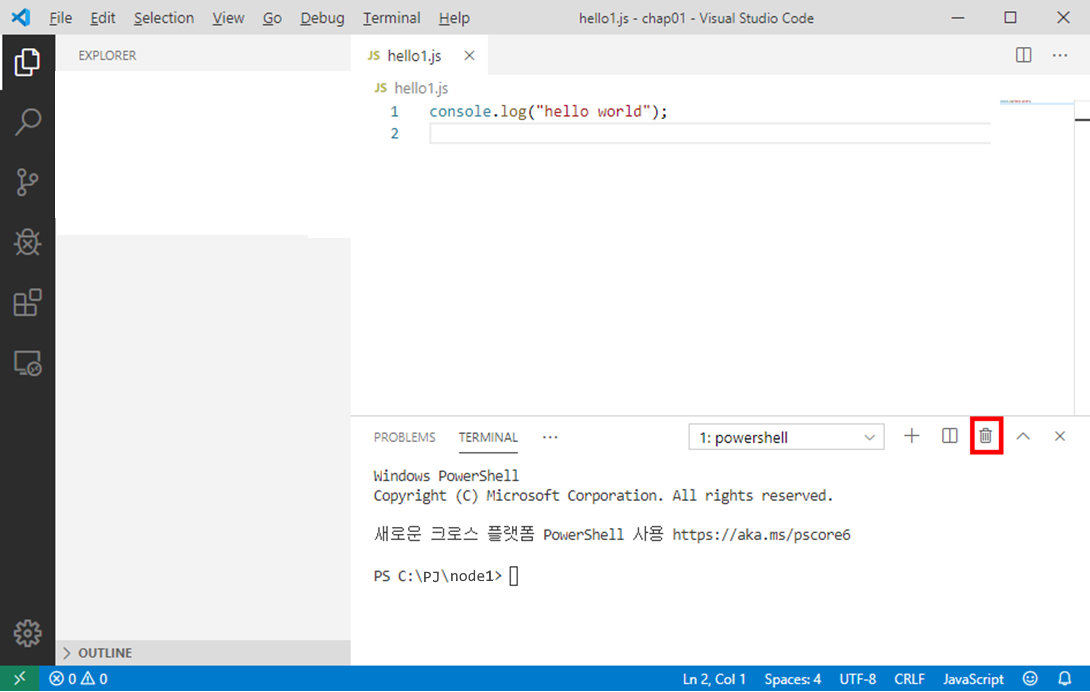
Git Bash : Git 클라이언트에 포함된 bash 쉘

오픈 소스 개발자들이 사용하는 운영체제는 리눅스이다.

실력있는 개발자라면 리눅스의 bash 쉘에 익숙해져야 한다.

Git Bash 항목을 선택하자.

### 터미널 다시 열기



터미널 창 위의 휴지통 버튼을 클릭하여 터미널 창을 닫자.

그리고 다시 터시널 창을 새로 열자 (단축키: Ctrl+`)

새로 열린 터미널 창에서 bash 쉘이 실행될 것이다.

현재 디렉토리의 파일 목록을 출력하는 리눅스 명령은 다음과 같다. (디렉토리 == 폴더)

|  |
| --- |
| ls -al |

|  |
| --- |
|  |

현재 디렉토리 아래의 chap01 디렉토리로 들어가는 명령은 다음과 같다.

|  |
| --- |
| cd chap01 |

|  |
| --- |
|  |

## javascript 파일 실행

### 쉘에서 실행

자바스크립트 파일을 쉘에서 실행하기 위한 명령은 다음과 같다.

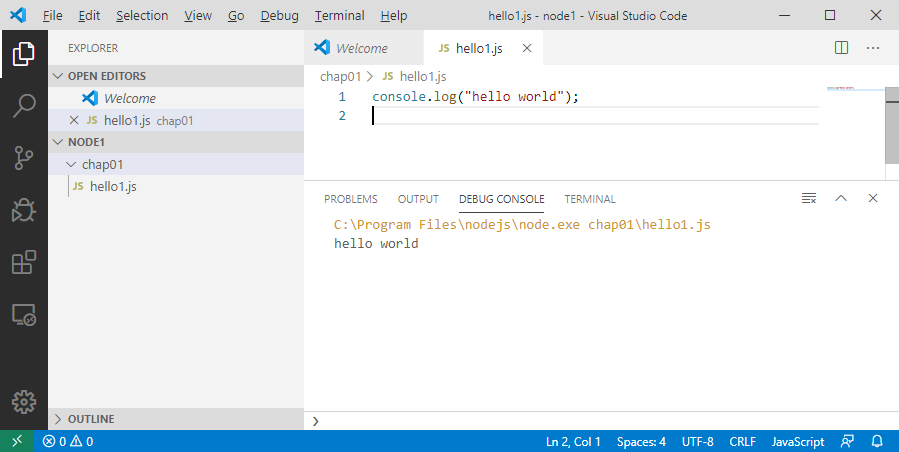
|  |
| --- |
| node 파일명 |

|  |
| --- |
|  |

화면에 hello world 문자열이 출력되었다.

### visual studio code에서 현재 파일 실행 단축키

Ctrl+F5



### 터미널 창 감추기

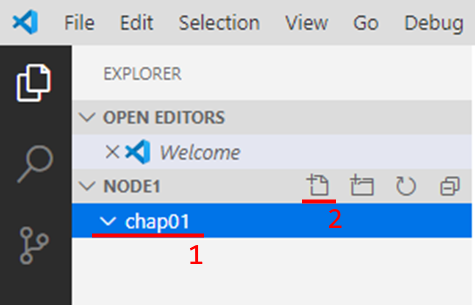
Ctrl+` 단축키를 계속해서 누르면, 터미널 창이 보였다 감춰졌다 한다.

쉘이 종료되는 것은 아니다.

터미널 창 위의 휴지통 버튼을 클릭하면, 터미널 창이 닫히고 쉘이 종료된다.

## let 키워드로 변수 선언

### 파일 생성



chap01 폴더를 선택하고, 파일 생성 버튼을 클릭하여, chap01 폴더 아래에 hello2.js 파일을 만들자.

### hello2.js

|  |
| --- |
| let message = "hello world";  console.log(message); |

let 키워드를 사용하여, 지역 변수를 선언한다.

javascript 언어에서는 변수를 선언할 때 변수의 타입을 지정하지 않는다.

let message = "hello world";

messsage 지역 변수에 "hello world" 문자열이 대입된다.

### hello2.js 실행

상위 폴더로 이동 명령

|  |
| --- |
| cd .. |

|  |
| --- |
|  |

/c/PJ/node1/chap01 폴더에서 /c/PJ/node1 폴더로 이동함.

chap01/hello2.js 실행 명령

|  |
| --- |
| node chap01/hello2.js |

|  |
| --- |
|  |

## var 키워드로 변수 선언

### hello3.js

|  |
| --- |
| var message = "hello world";  console.log(message); |

실행

|  |
| --- |
|  |

var 키워드를 사용하여 변수를 선언할 수도 있다.

var 키워드로 선언한 변수는, let 키워드로 선언한 변수와 미묘한 차이점이 있다.

이 차이를 이해하는 것은 너무 고급 문법이라서 여기서 설명하지 않겠다.

기억해야할 요점은,

지역 변수를 선언할 때 var 이 아니고 let 을 사용해야 한다는 것.

## 세미콜론 생략 가능

### hello4.js

|  |
| --- |
| let message = "hello world"  console.log(message) |

실행

|  |
| --- |
|  |

javascript 문법에서는, 문장(statement) 끝의 세미콜론(;) 문자를 생략할 수 있다.

# 내용 정리

javascript 언어는 인터프리터 방식으로 실행된다.

웹브라우저에 javascript 인터프리터가 내장되어 있다.

node.js에도 javascript 인터프리터가 내장되어 있다.

node.js는, javascript로 구현된 코드를 서버에서 실행하기 위한 플랫폼이다.

플랫폼 = 실행 엔진(인터프리터) + 라이브러리 + 개발도구

Windows의 기본 쉘은 cmd.exe 이다.

Linux의 기본 쉘은 bash 이다.

쉘을 실행하기 위한 창을 터미널(terminal)이라고 한다.

LTS (Long Term Support)는 버그 패치 및 보안 패치를 오랜 기간 지원하는 버전이다.

console.log(...) 메소드는 화면에 값을 출력한다.

let 키워드를 사용하여 지역 변수를 선언한다. 예: let i = 3;

변수를 선언할 때 var 키워드를 사용할 수도 있지만, let 이 더 좋다.

javascript 언어에서는 변수를 선언할 때 변수의 타입을 지정하지 않는다.

javascript 소스 코드 파일의 실행은 main 함수에서 시작하지 않는다.

javascript 소스코드 파일을 실행하면, 파일의 소스 코드들이 순서대로 실행된다.

javascript 문법에서는, 문장(statement) 끝의 세미콜론(;) 문자를 생략할 수 있다.