
HTML에서 웹앱까지

10주차_03

한 동 대 학 교
김군오 교수

학습 목표:

NodeJS 소개 및 로컬 설치하기

1. NodeJS 소개
2. 로컬환경에 NodeJS 설치
3. Visual Studio Code NodeJS 디버깅

NodeJS

- 브라우저 외부에서 실행되는 자바스크립트 엔진
- 구글 크롬의 자바스크립트 엔진인 V8 을 이용해 만들어짐
- 기본 자바스크립트 기능 외에 운영체제 기능을 활용할 수 있는 라이브러리 제공
- 주로 개발자 도구나 웹 서버 상에서 실행되는 프로그램을 작성할 때 많이 사용

Node의 특징

- 비동기 I/O 처리
 - 시간이 많이 걸리는 Input, Output 작업을 비동기로 처리

Node의 특징

- 비동기 I/O 처리
 - 시간이 많이 걸리는 Input, Output 작업을 비동기로 처리
- 빠른 속도
 - 구글 크롬의 자바스크립트 엔진인 V8의 등장으로 자바스크립트의 속도 문제도 어느 정도 해결

Node의 특징

- 비동기 I/O 처리
 - 시간이 많이 걸리는 Input, Output 작업을 비동기로 처리
- 빠른 속도
 - 구글 크롬의 자바스크립트 엔진인 V8의 등장으로 자바스크립트의 속도 문제도 어느 정도 해결
- 패키지 매니저를 통한 의존성 관리
 - NPM 이라는 패키지 매니저를 제공
 - 노드 패키지 매니저는 전세계 수 많은 사람들이 자신들의 모듈을 업로드 할 수 있는 코드 저장소

Node의 특징

- 비동기 I/O 처리
 - 시간이 많이 걸리는 Input, Output 작업을 비동기로 처리
- 빠른 속도
 - 구글 크롬의 자바스크립트 엔진인 V8의 등장으로 자바스크립트의 속도 문제도 어느 정도 해결
- 패키지 매니저를 통한 의존성 관리
 - NPM 이라는 패키지 매니저를 제공
 - 노드 패키지 매니저는 전세계 수 많은 사람들이 자신들의 모듈을 업로드 할 수 있는 코드 저장소
- 단일 쓰레드
 - 쓰레드란 CPU 가 동시에 여러가지 일을 처리할 수 있게 해주는 기능

NPM

- Node Package Manager
- NodeJS 로 만들어진 모듈 공유
- 프로젝트 의존성 관리를 위한 package.json

로컬 환경에 NodeJS 설치

- <https://nodejs.org/>



Node.js® is a JavaScript runtime built on **Chrome's V8 JavaScript engine**.

Download for macOS (x64)

10.16.0 LTS

Recommended For Most Users

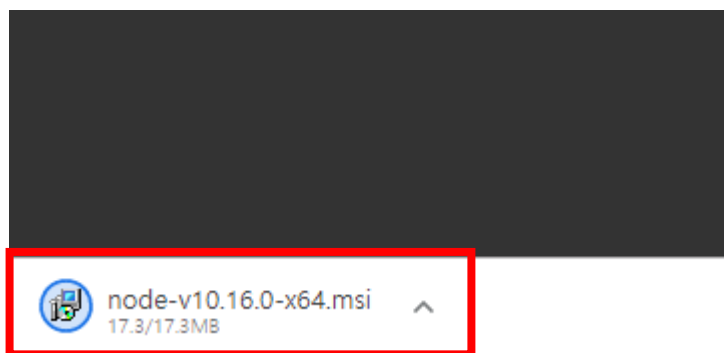
12.5.0 Current

Latest Features

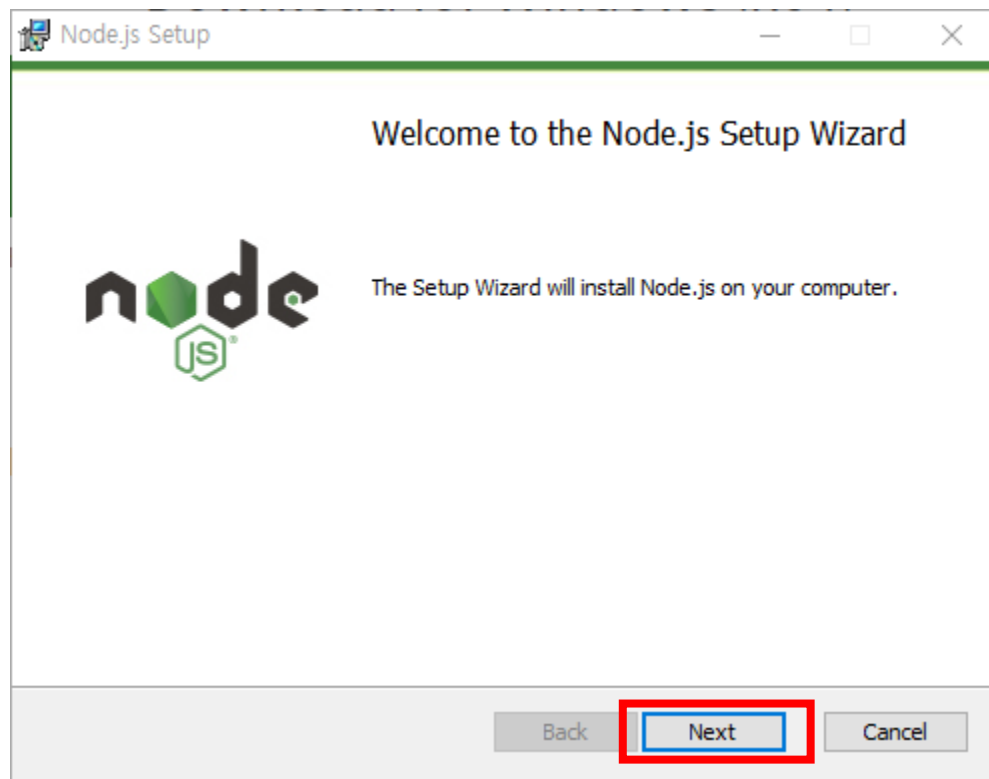
[Other Downloads](#) | [Changelog](#) | [API Docs](#)

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

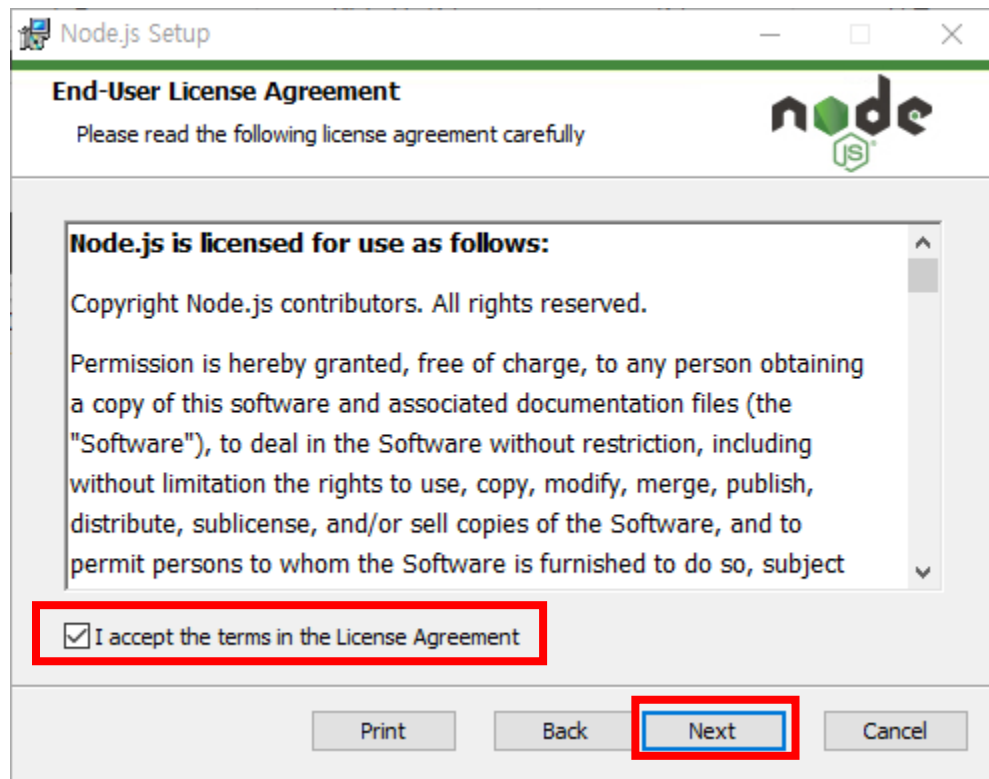
로컬 환경에 NodeJS 설치



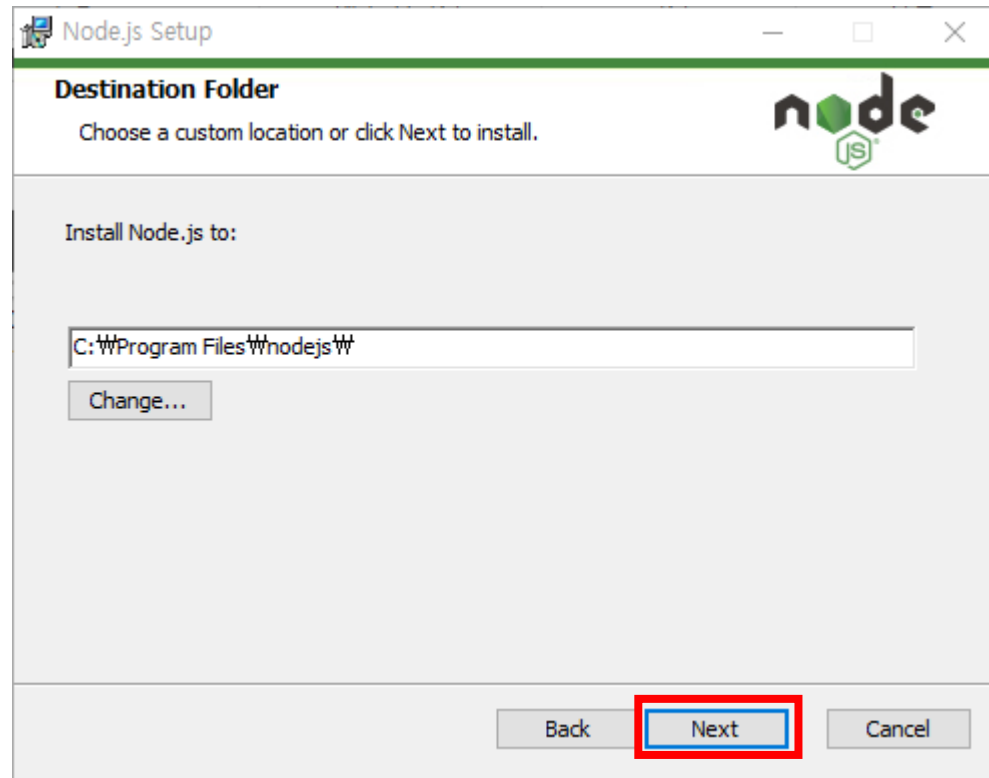
로컬 환경에 NodeJS 설치



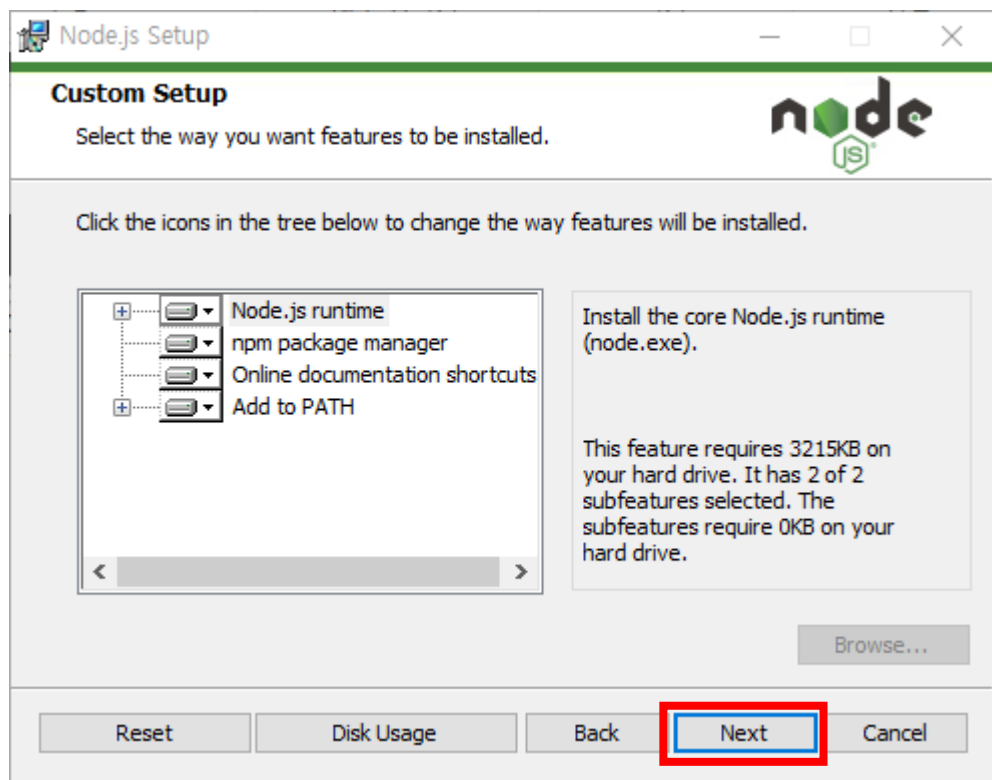
로컬 환경에 NodeJS 설치



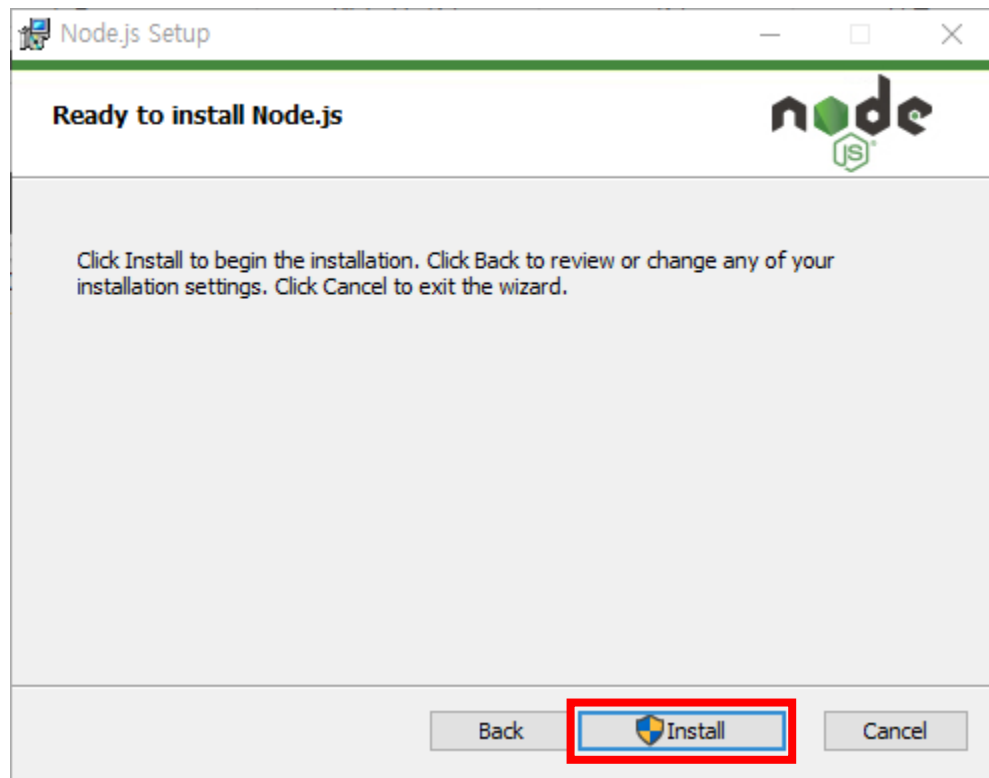
로컬 환경에 NodeJS 설치



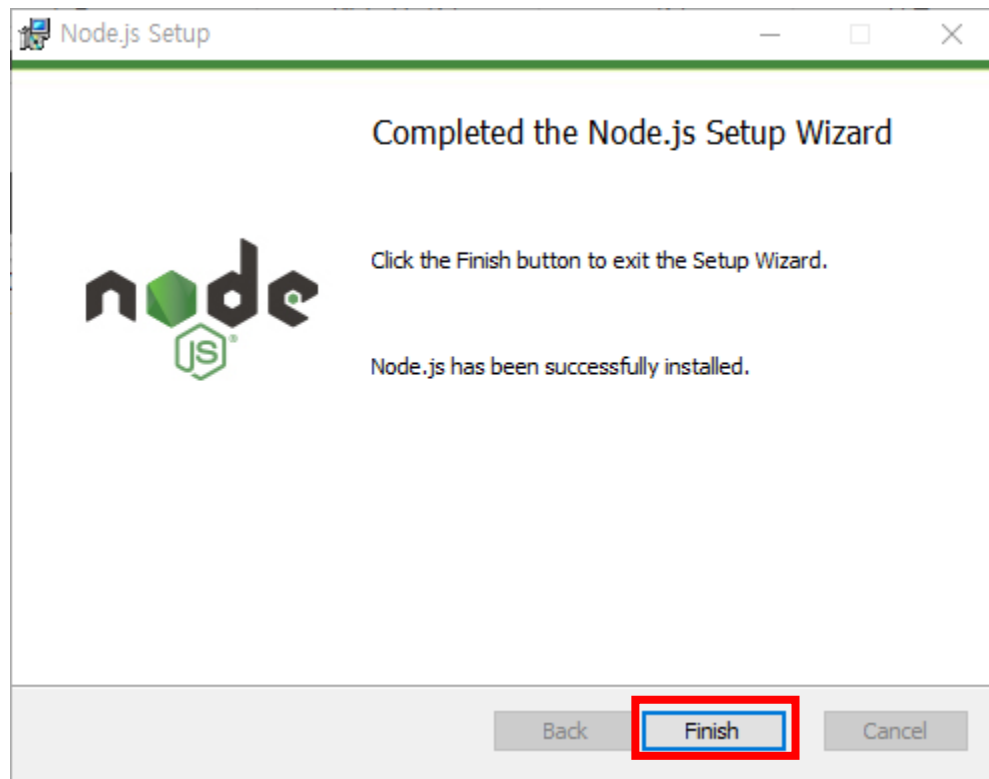
로컬 환경에 NodeJS 설치



로컬 환경에 NodeJS 설치

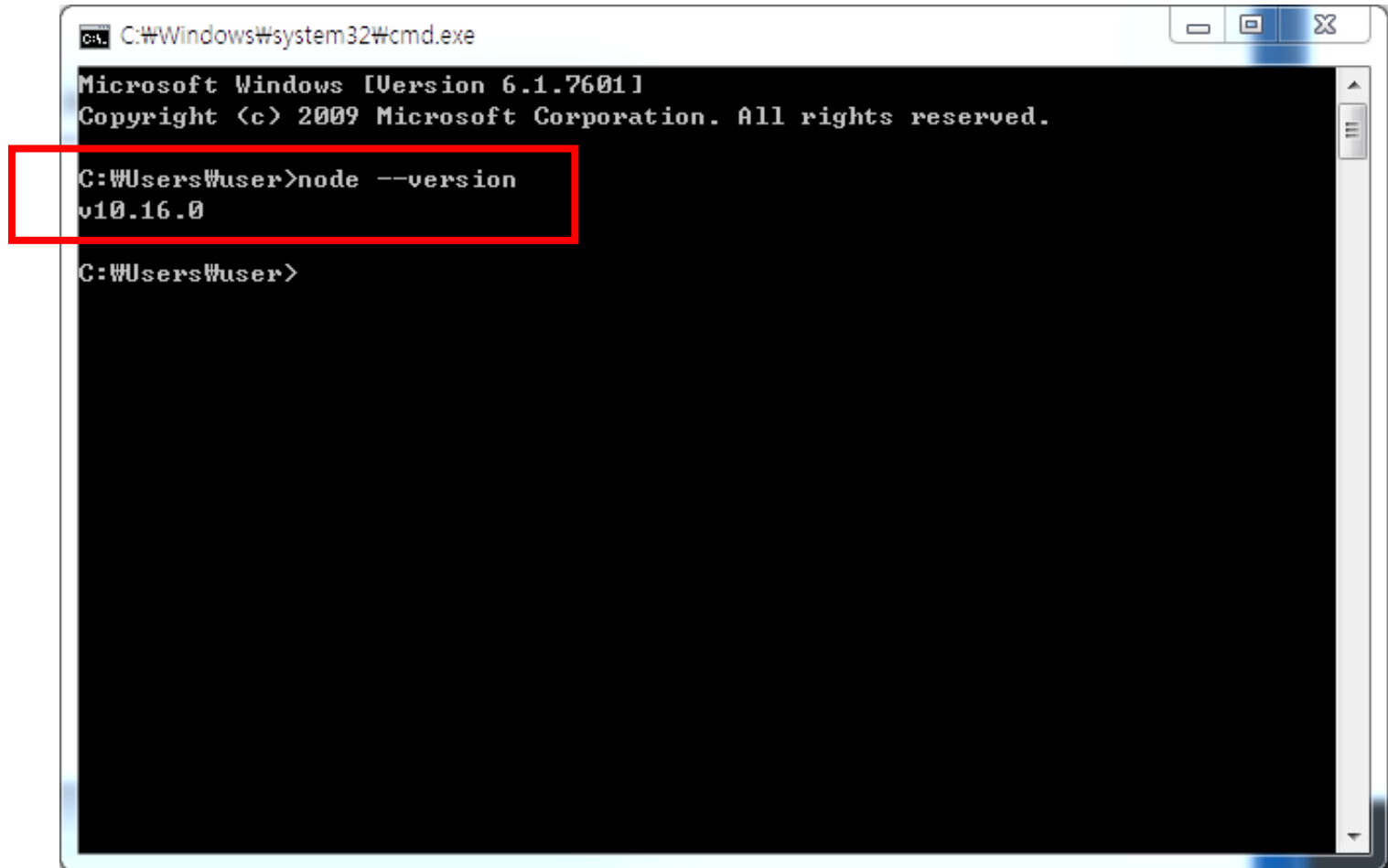


로컬 환경에 NodeJS 설치



NodeJS 설치 확인

- node --version



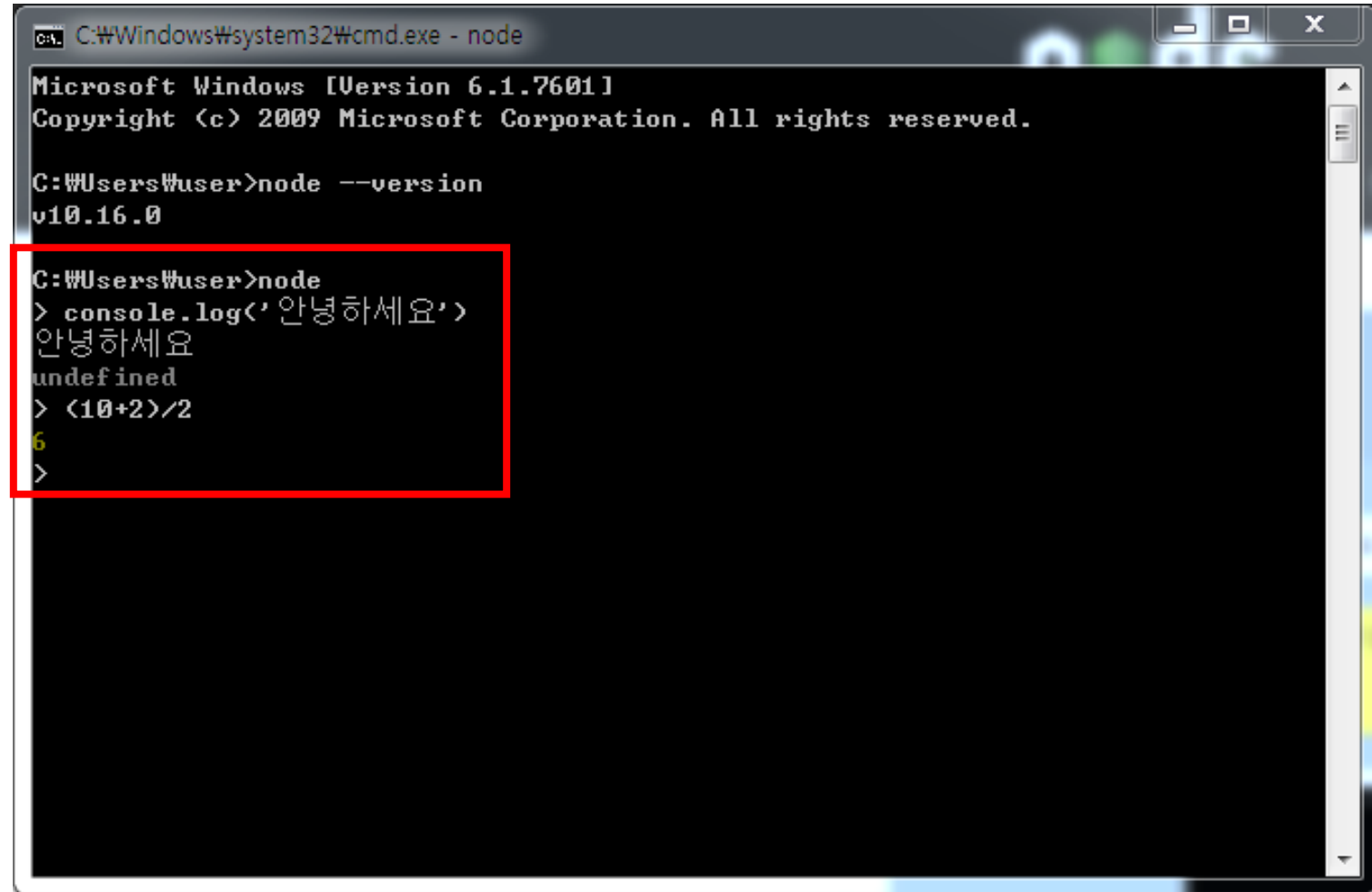
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>node --version
v10.16.0

C:\Users\User>
```

NodeJS 실행

- node

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe - node". The window shows the output of several Node.js commands. The first command is "node --version", which outputs "v10.16.0". The second command is "node", which starts an interactive Node.js shell. In this shell, the command "> console.log('안녕하세요');" is entered, followed by the output "안녕하세요" and "undefined". Then, the command "> (10+2)/2" is entered, followed by the output "6". The prompt ">" is visible at the bottom of the red box.

```
C:\Windows\system32\cmd.exe - node
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

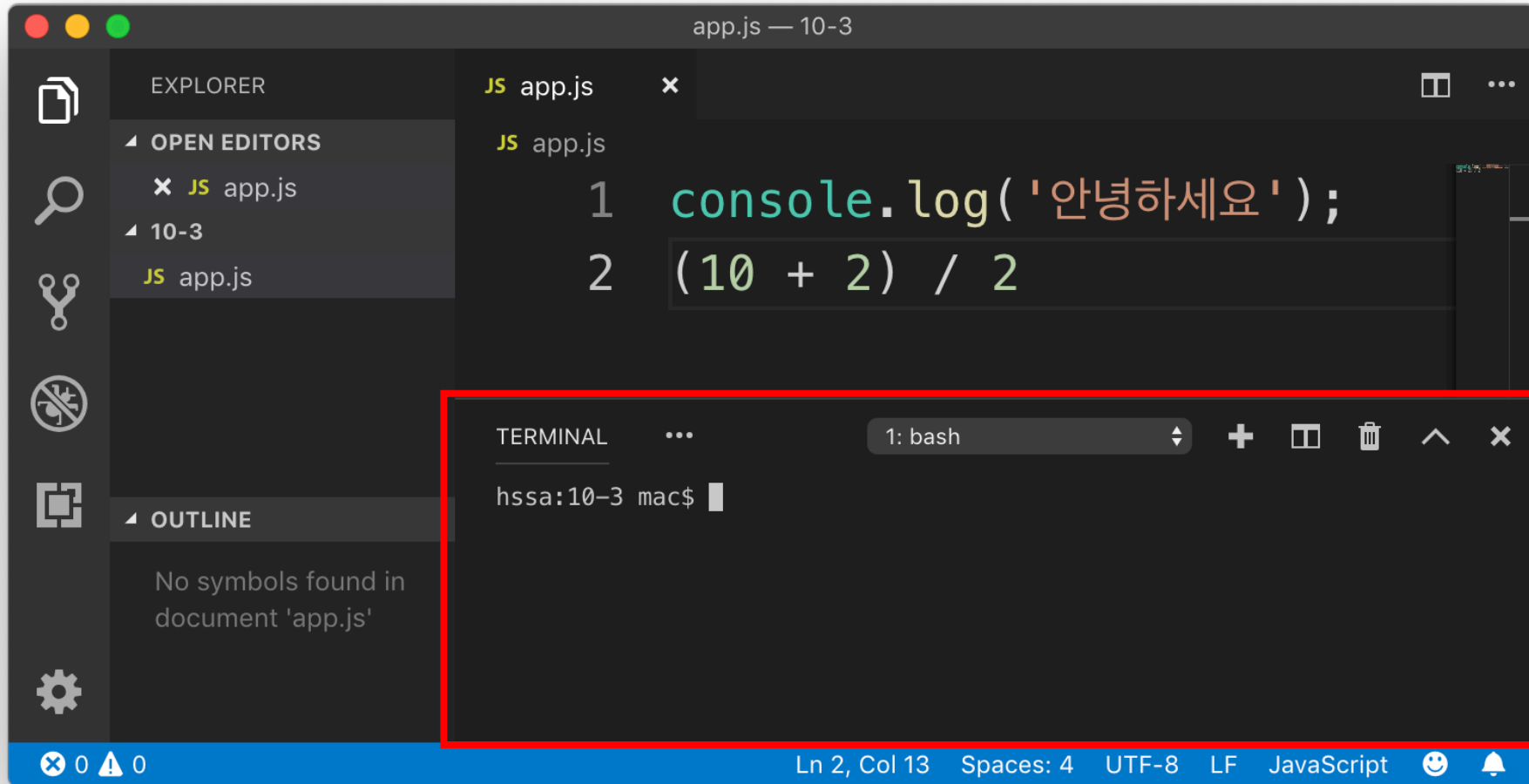
C:\Users\User>node --version
v10.16.0

C:\Users\User>node
> console.log('안녕하세요')
안녕하세요
undefined
> (10+2)/2
6
>
```

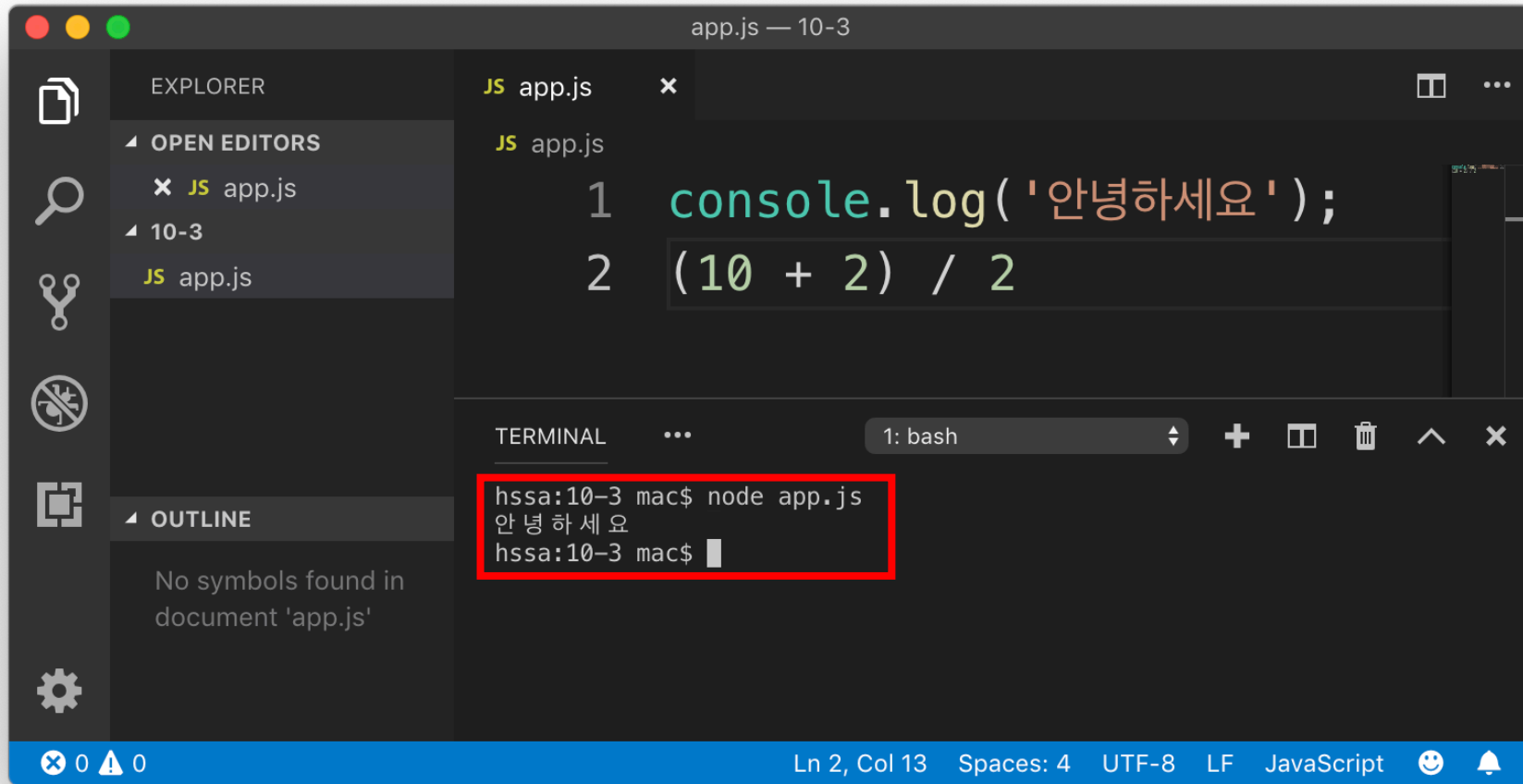
JS 파일에서 NodeJS 실행

- 콘솔에서 개발한 코드는 따로 저장되지 않으므로 재현이 불가능
- 웹 페이지를 개발할 때와 마찬가지로 파일에 스크립트를 저장할 필요 있음
- `node {파일명}` 명령어 사용

JS 파일에서 NodeJS 실행



JS 파일에서 NodeJS 실행



The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left shows the file structure with 'app.js' open. The main editor area displays the contents of 'app.js' with two lines of code: `console.log('안녕하세요');` and `(10 + 2) / 2`. The terminal at the bottom shows the command `node app.js` being executed, resulting in the output `안녕하세요`. The terminal output is highlighted with a red box. The status bar at the bottom indicates the current line and column (Ln 2, Col 13), the number of spaces (4), the encoding (UTF-8), the line ending (LF), and the language (JavaScript).

```
app.js — 10-3
```

EXPLORER

- OPEN EDITORS
 - JS app.js
- 10-3
 - JS app.js

OUTLINE

No symbols found in document 'app.js'

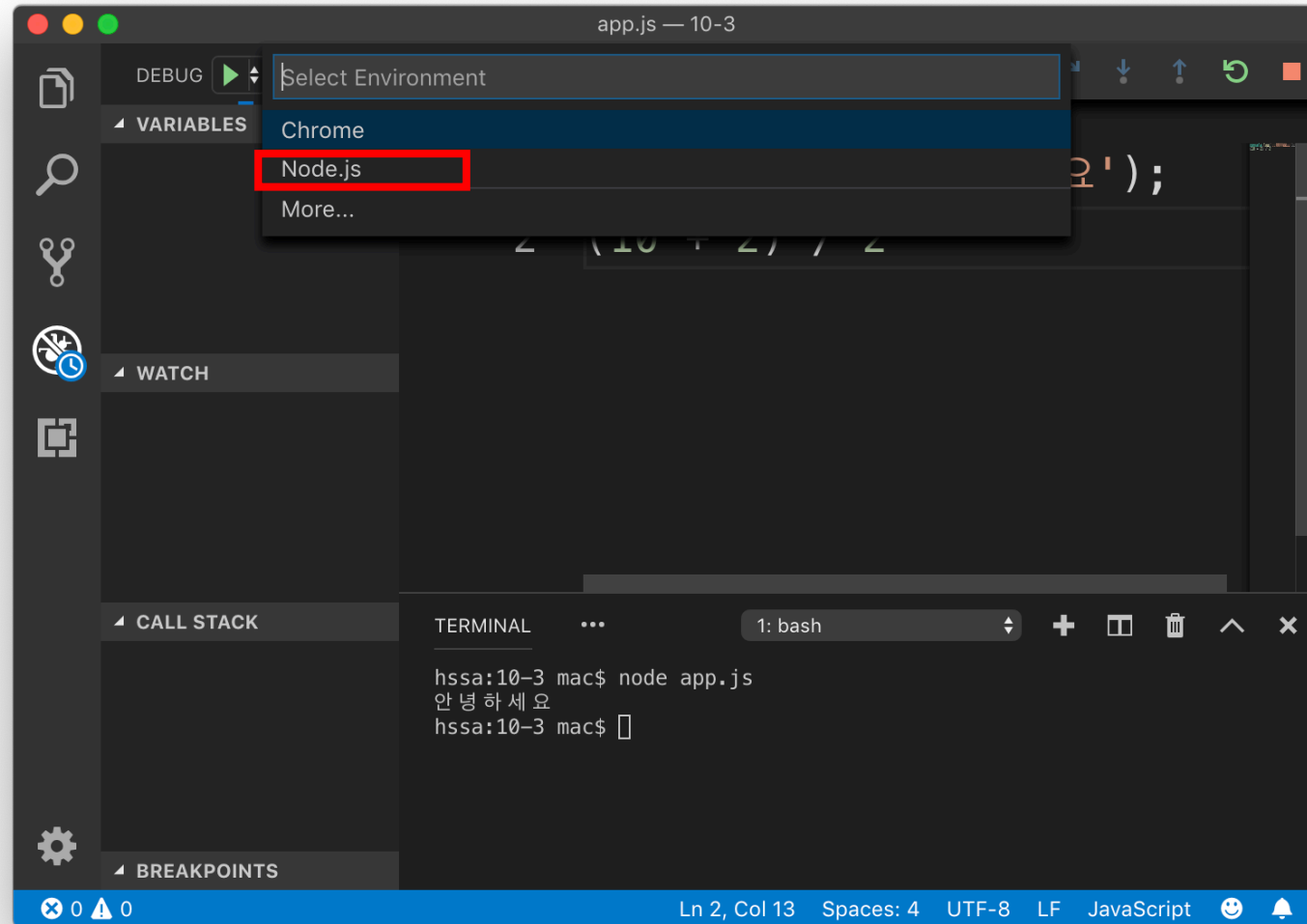
```
1 console.log('안녕하세요');
2 (10 + 2) / 2
```

TERMINAL 1: bash

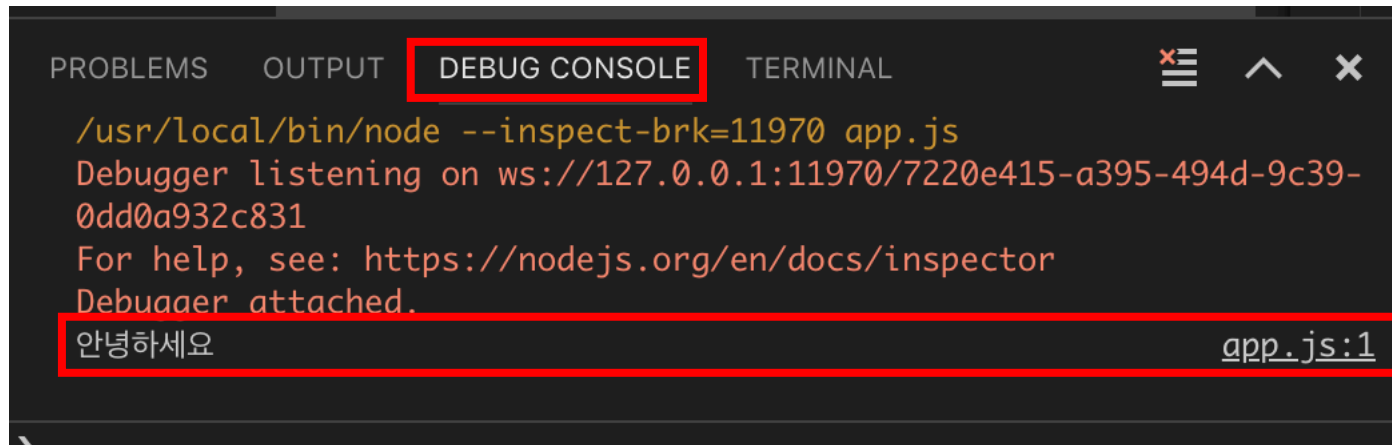
```
hssa:10-3 mac$ node app.js
안녕하세요
hssa:10-3 mac$
```

Ln 2, Col 13 Spaces: 4 UTF-8 LF JavaScript

Visual Studio Code 에서 NodeJS 개발



Visual Studio Code 에서 NodeJS 개발



The image shows the 'DEBUG CONSOLE' tab in Visual Studio Code. The console output includes the command to start Node.js with the inspector, the debugger listening message, and a confirmation that the debugger is attached. The final line of output, '안녕하세요', is highlighted with a red box, along with the file and line reference 'app.js:1' on the right side of the same line.

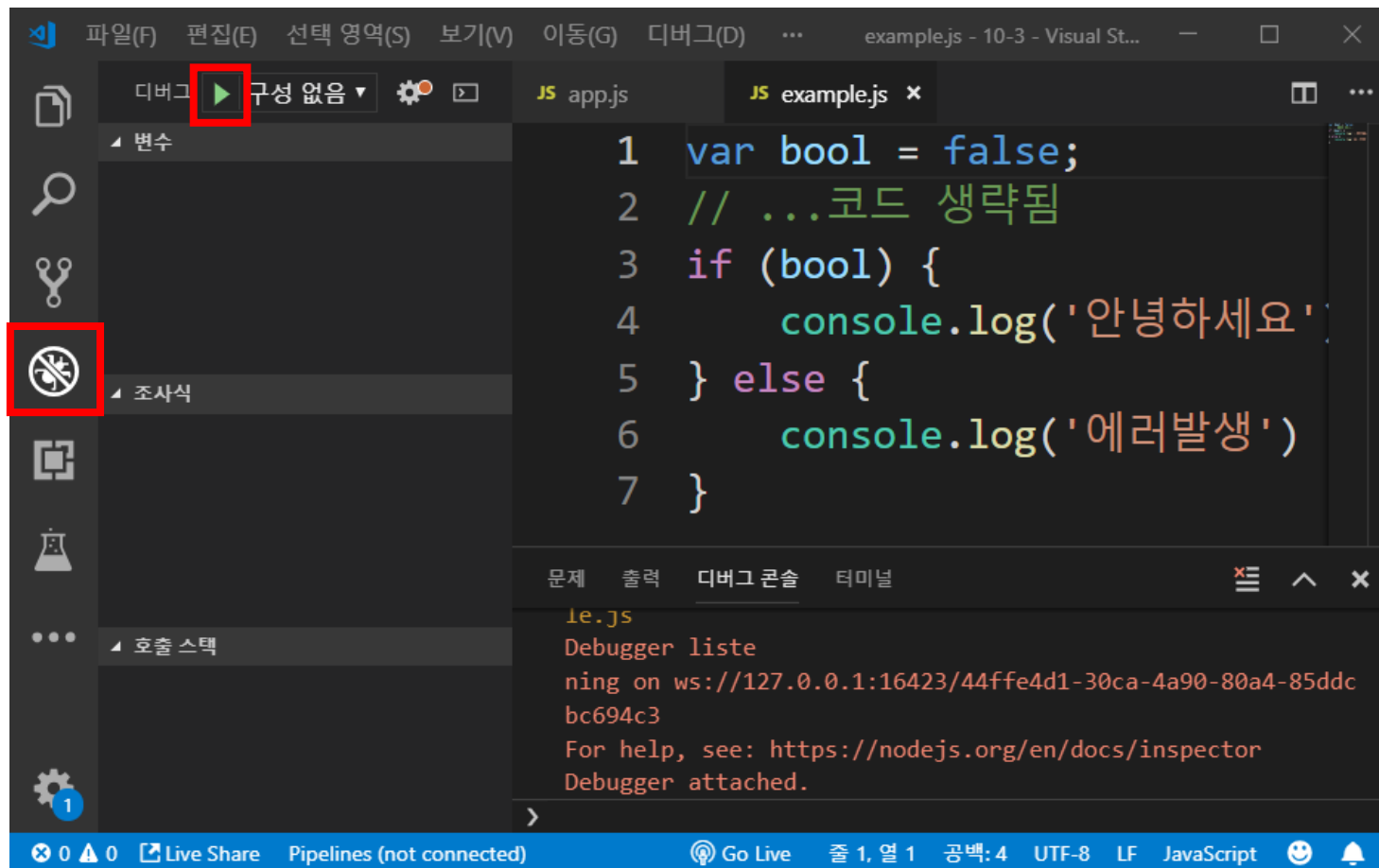
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
```

```
/usr/local/bin/node --inspect-brk=11970 app.js  
Debugger listening on ws://127.0.0.1:11970/7220e415-a395-494d-9c39-  
0dd0a932c831  
For help, see: https://nodejs.org/en/docs/inspector  
Debugger attached.  
안녕하세요 app.js:1
```

NodeJS 디버깅

- 코드에서 오류가 발생하거나 예상한 결과와 다를 때 코드를 고치는 작업
- 디버깅 도구의 중요성이 높아짐
- Visual Studio Code 에서 개발과 디버깅 통합

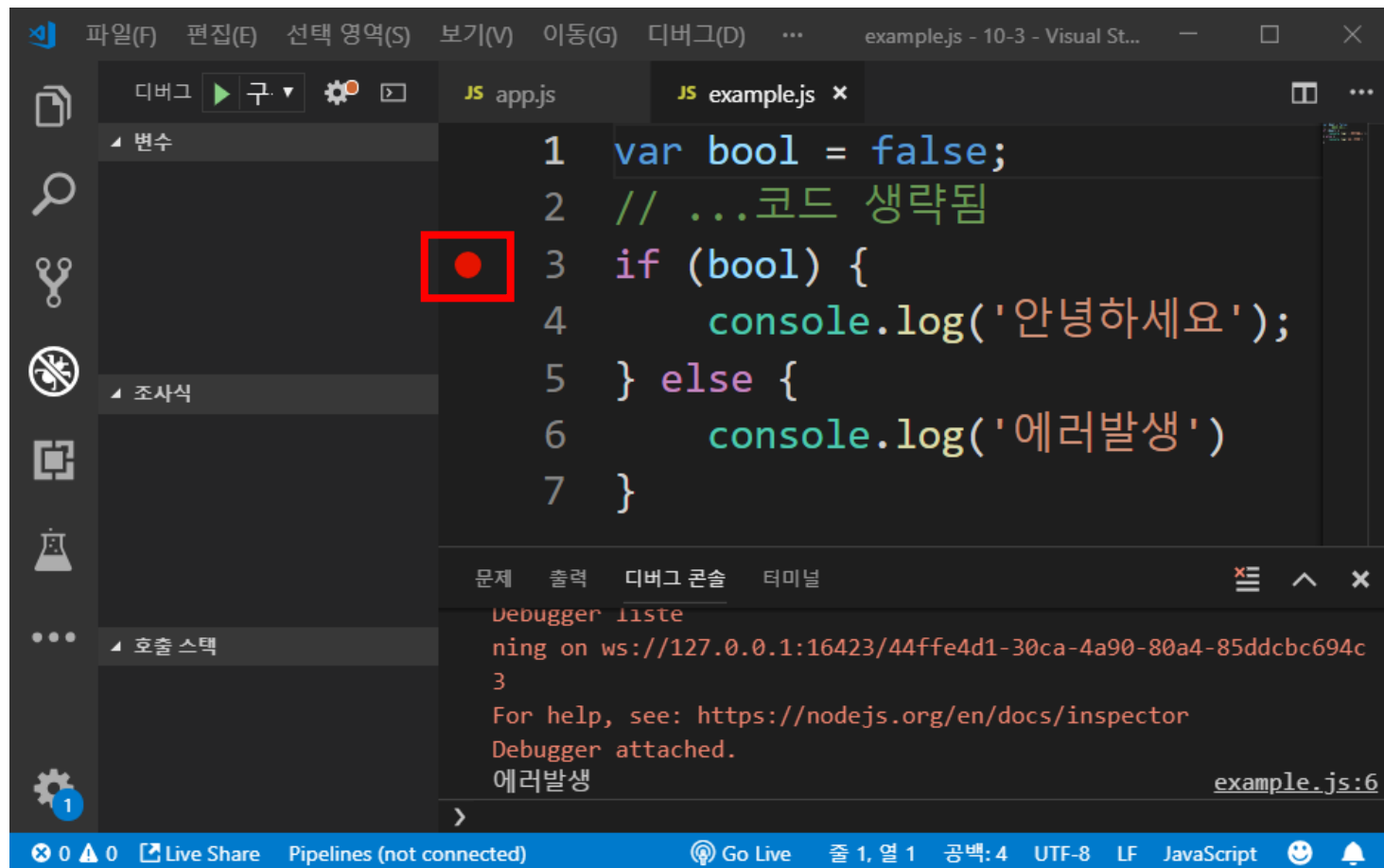
NodeJS 디버깅



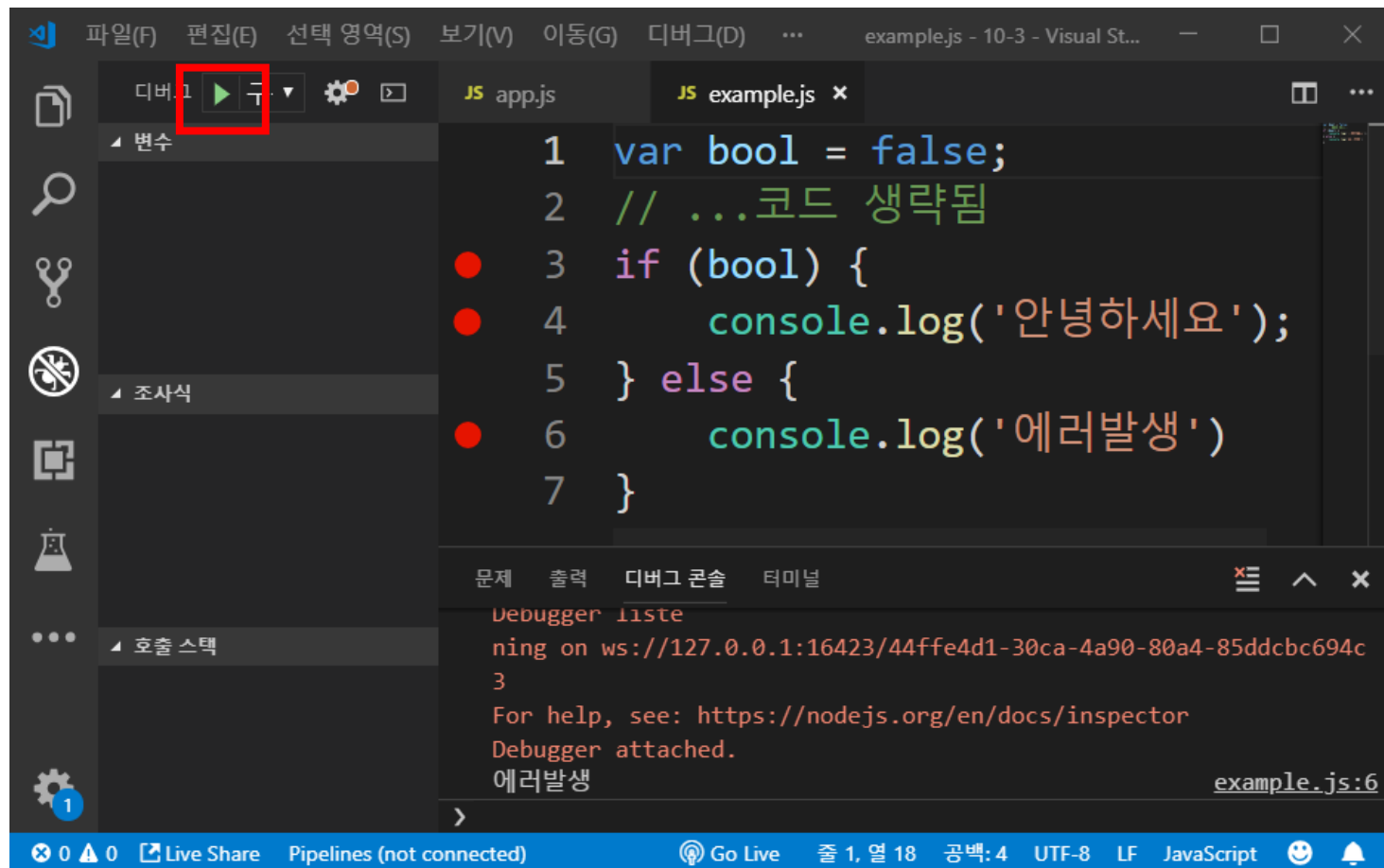
NodeJS 디버깅

```
문제  출력  디버그 콘솔  터미널
Debugger listening on ws://127.0.0.1:16423/44ffe4d1-30ca-4a90-80a4-85ddcbc694c3
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
에러발생 example.js:6
>
```

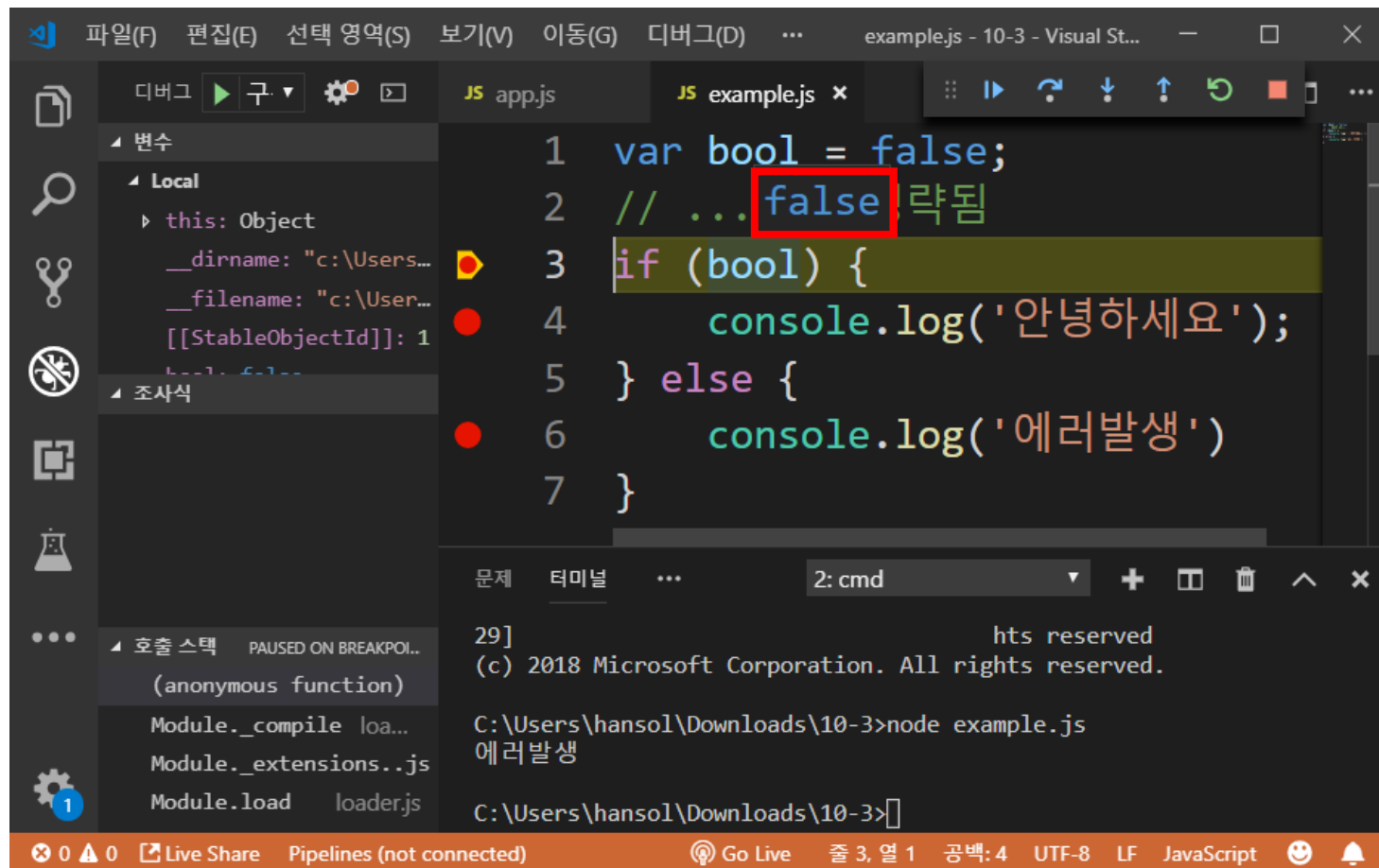
NodeJS 디버깅



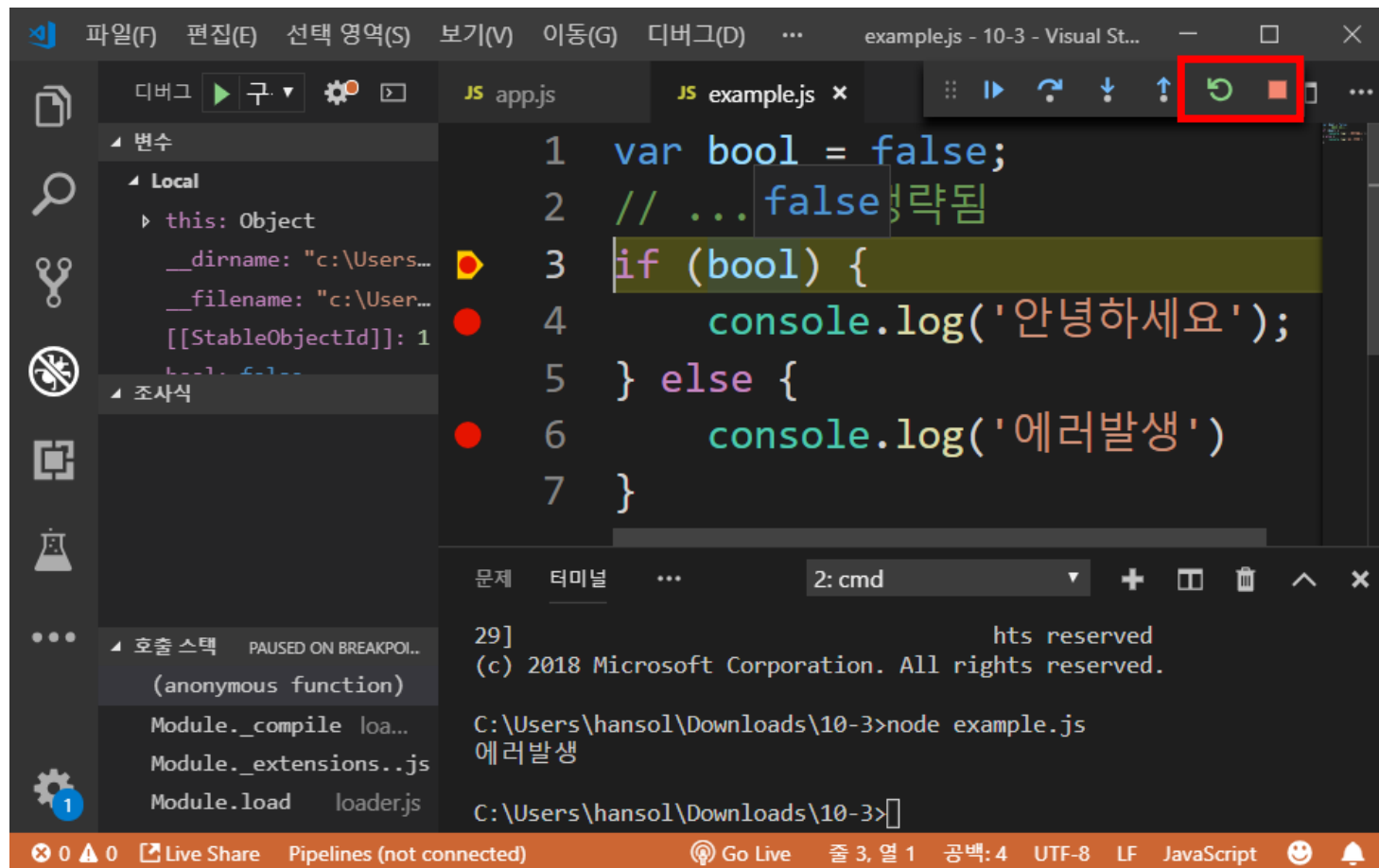
NodeJS 디버깅



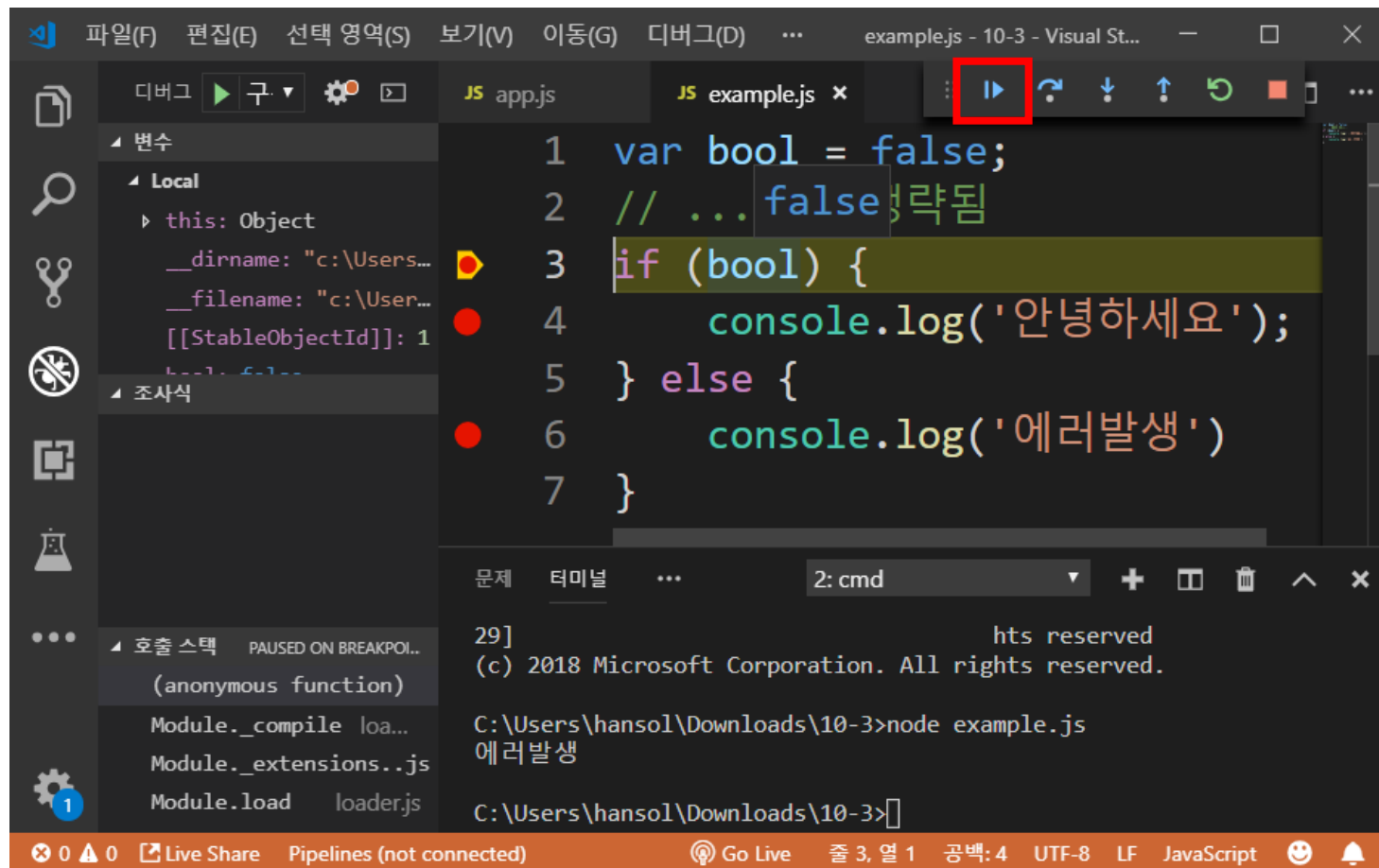
NodeJS 디버깅



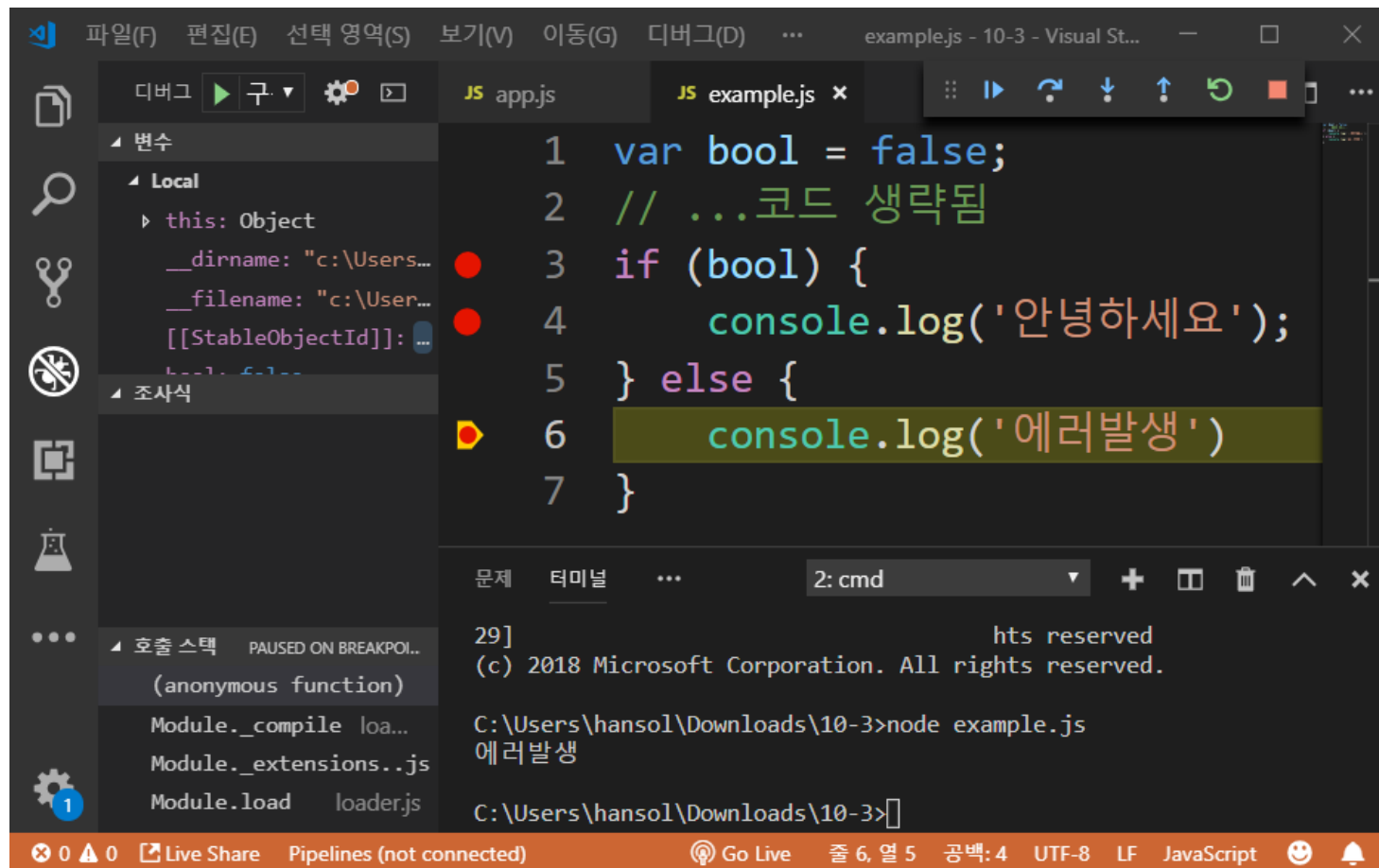
NodeJS 디버깅



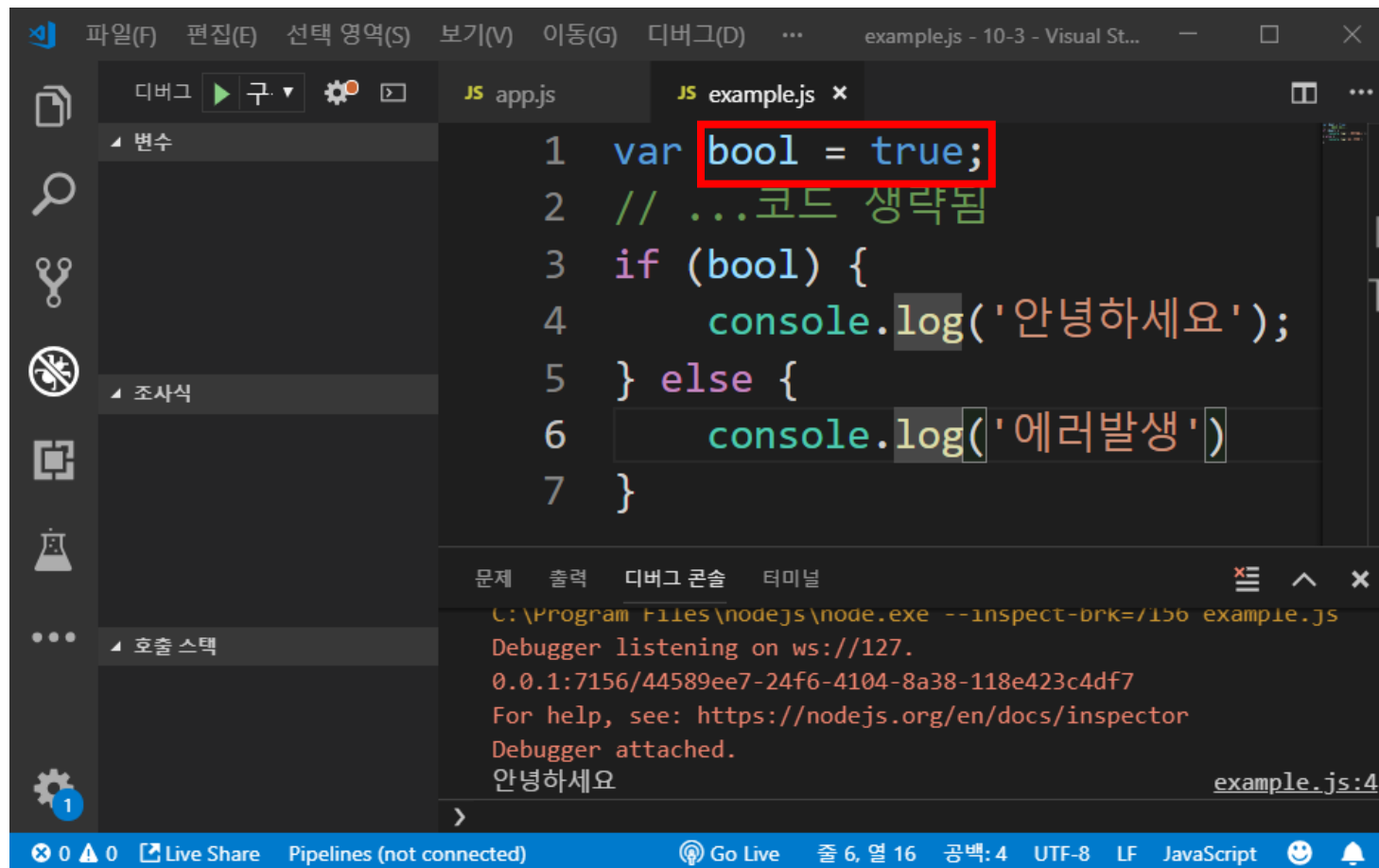
NodeJS 디버깅



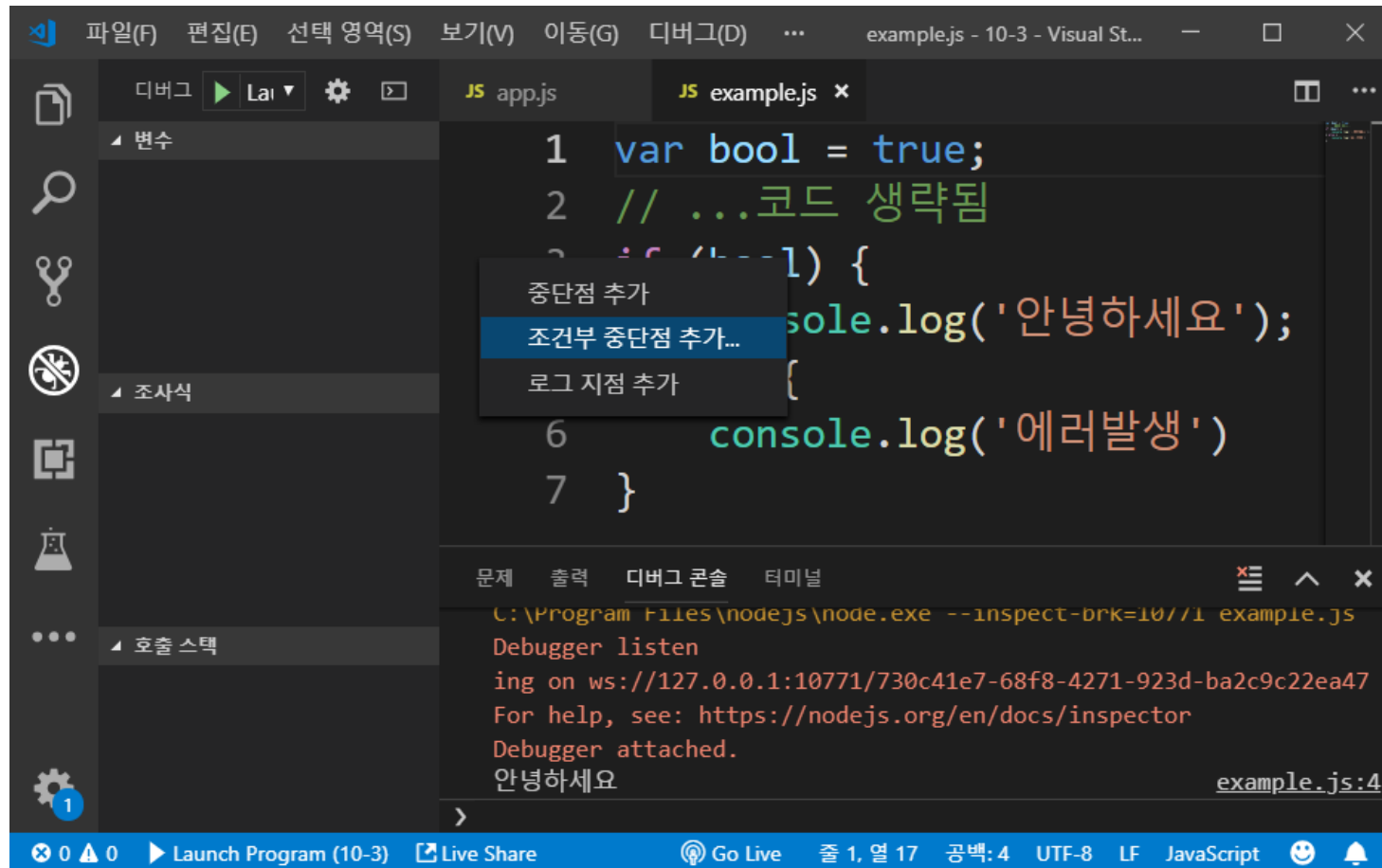
NodeJS 디버깅



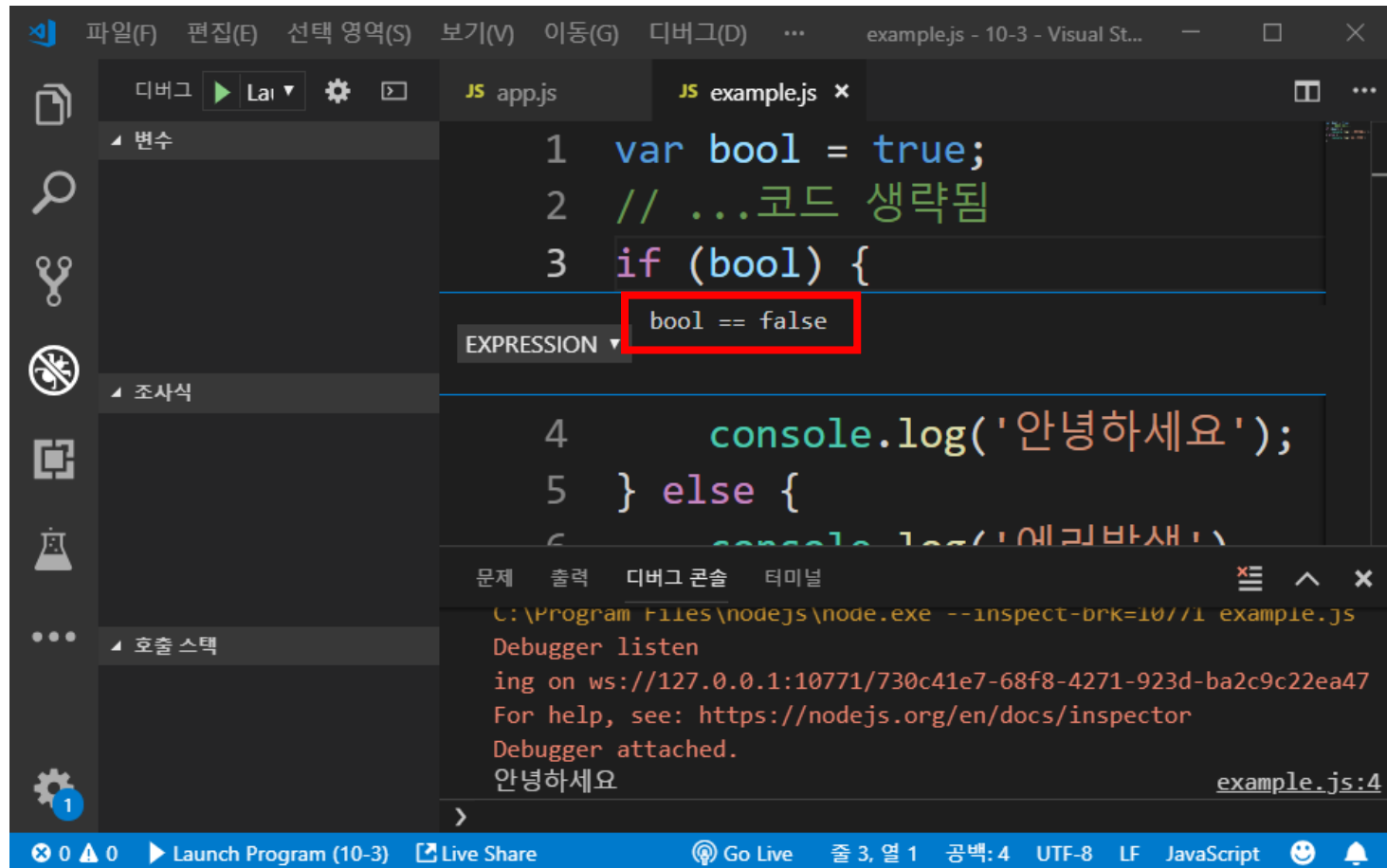
NodeJS 디버깅



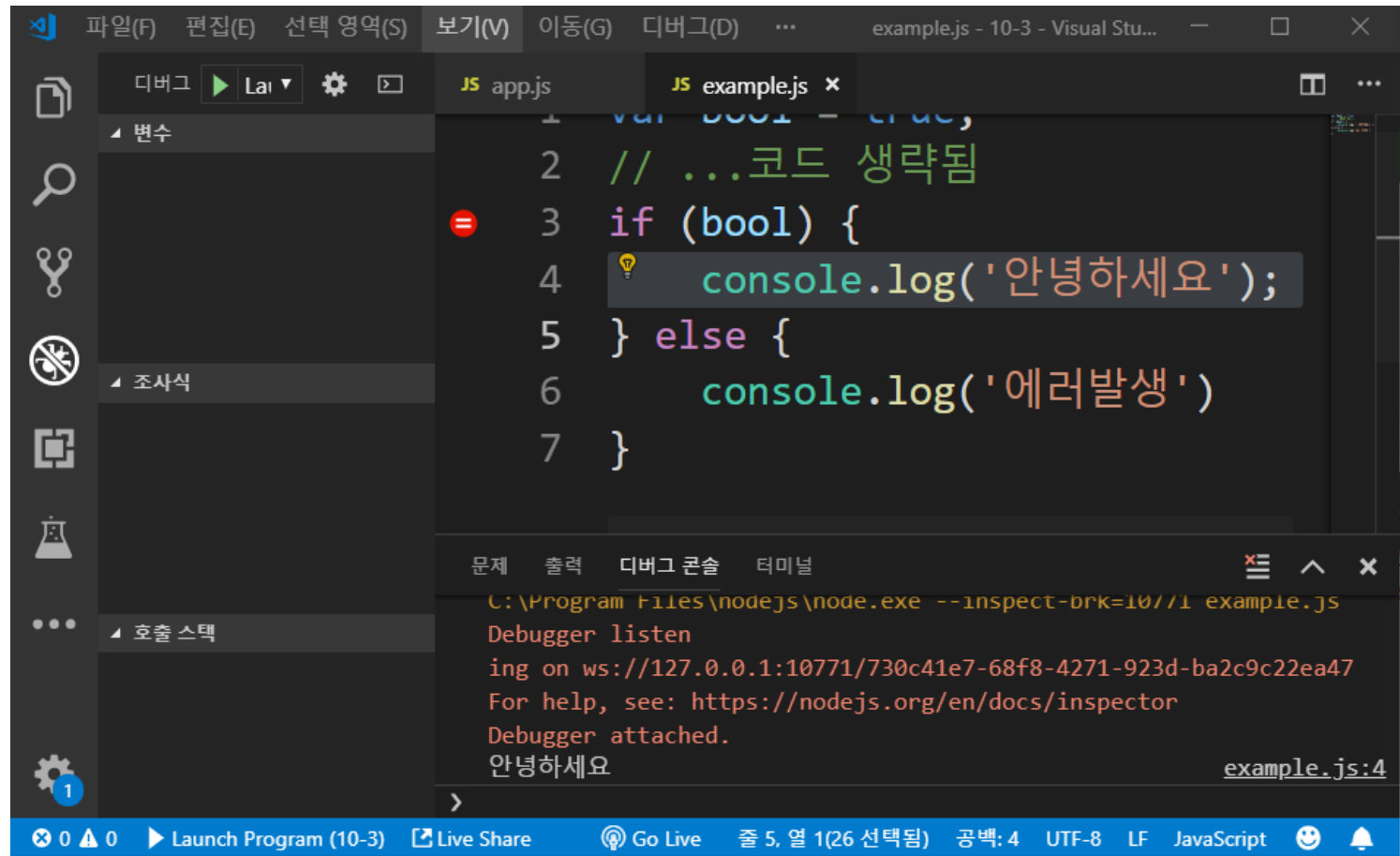
조건부 중단점



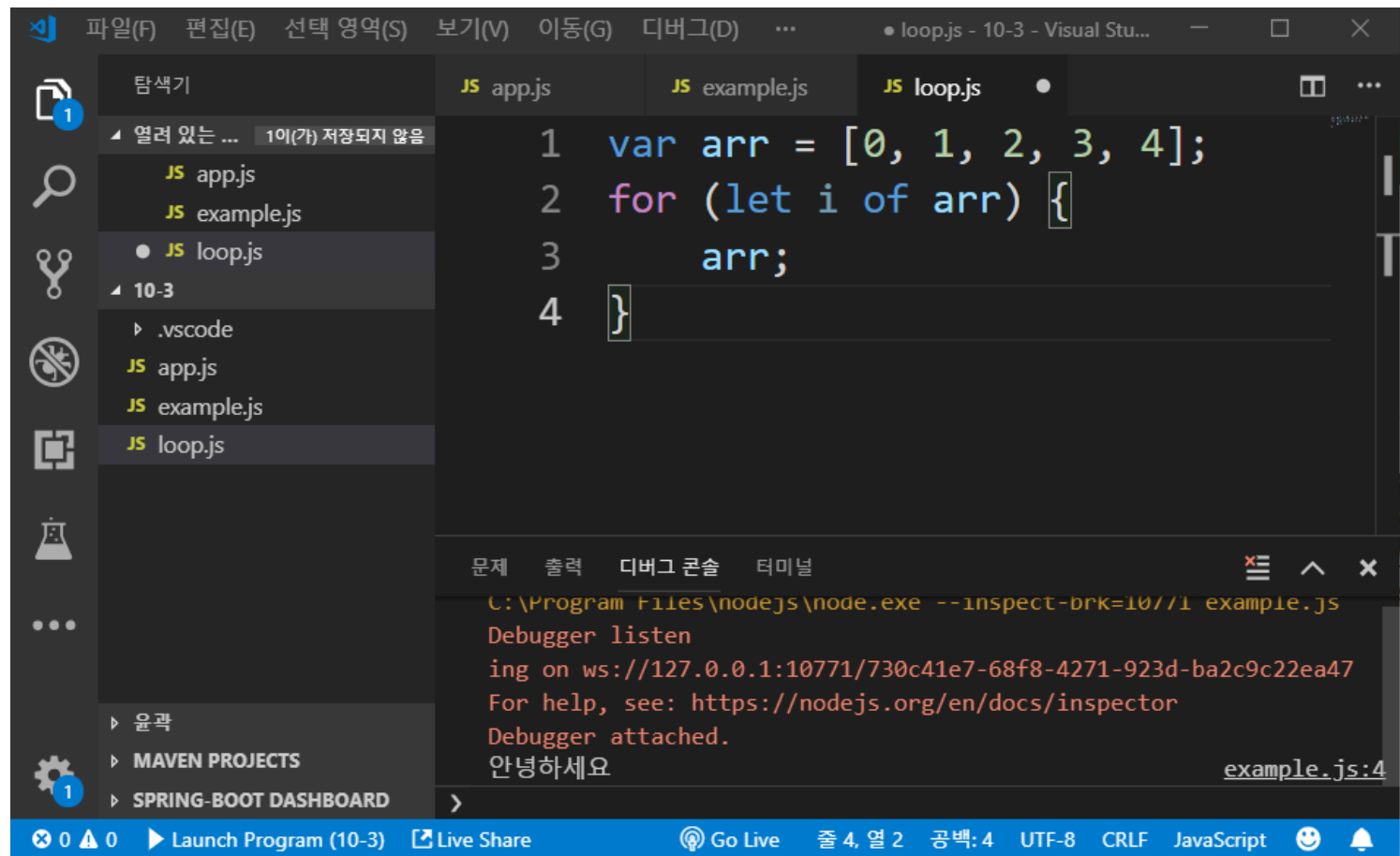
조건부 중단점



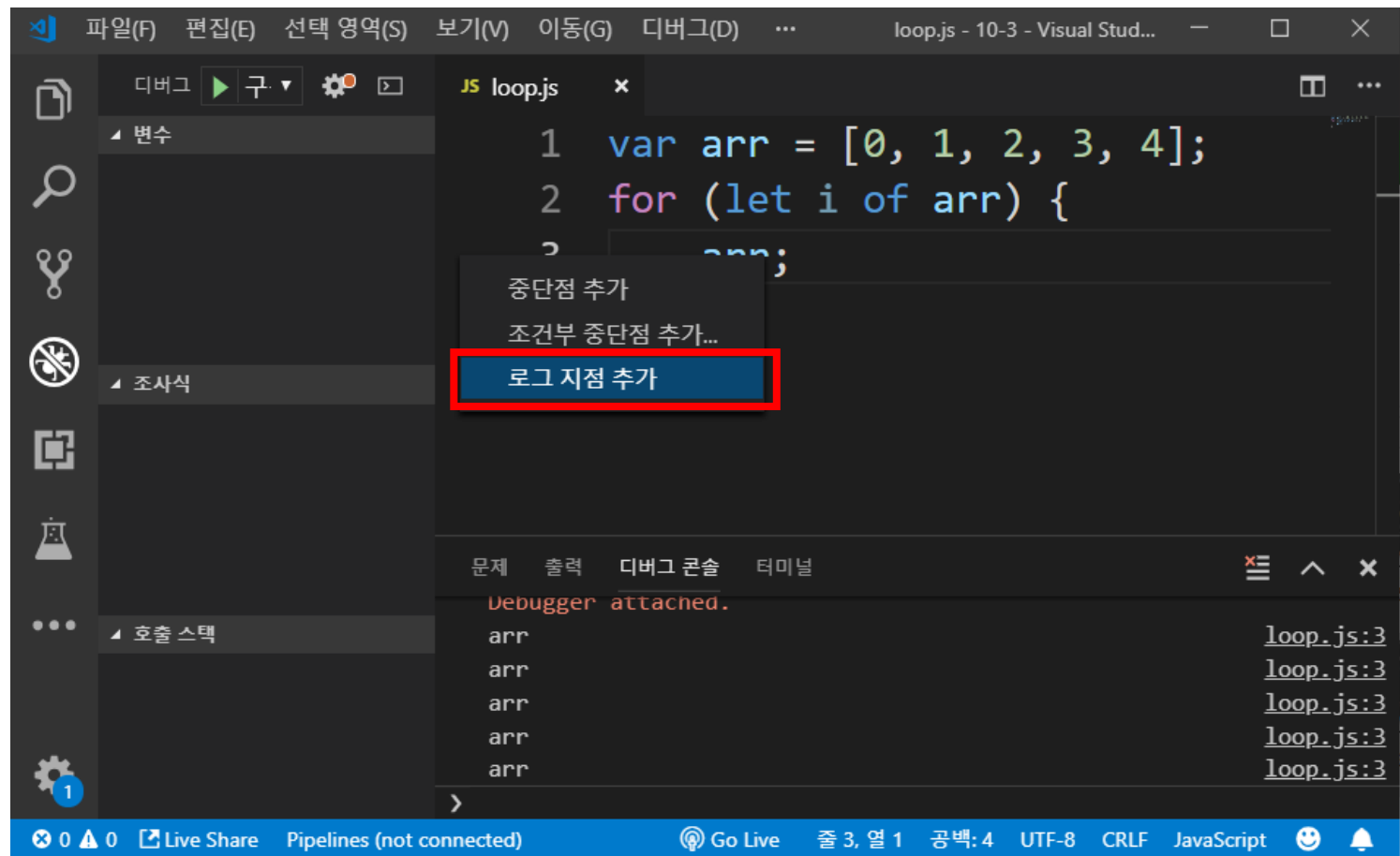
조건부 중단점



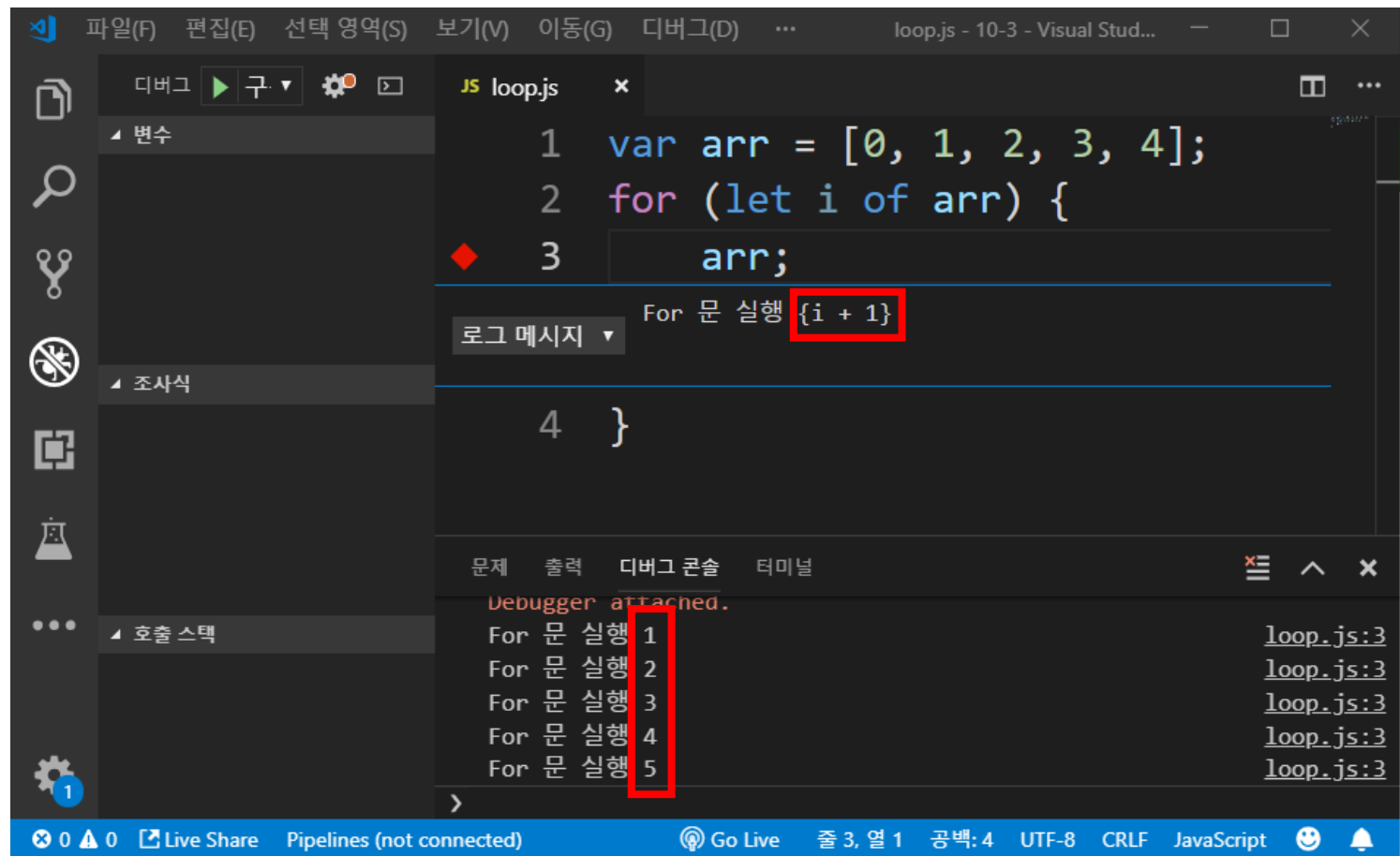
로그 프린트



로그 프린트



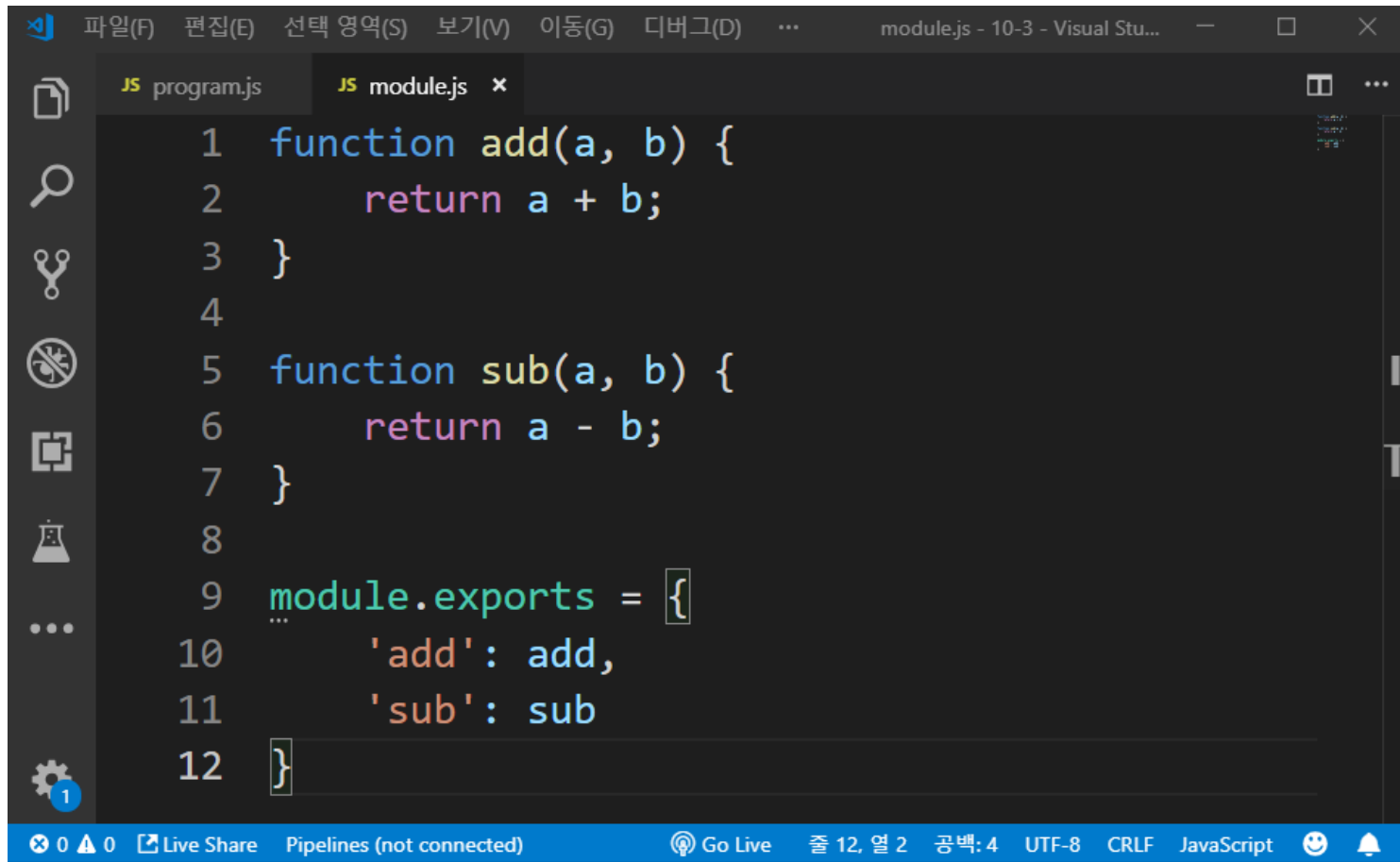
로그 프린트



모듈 프로그래밍

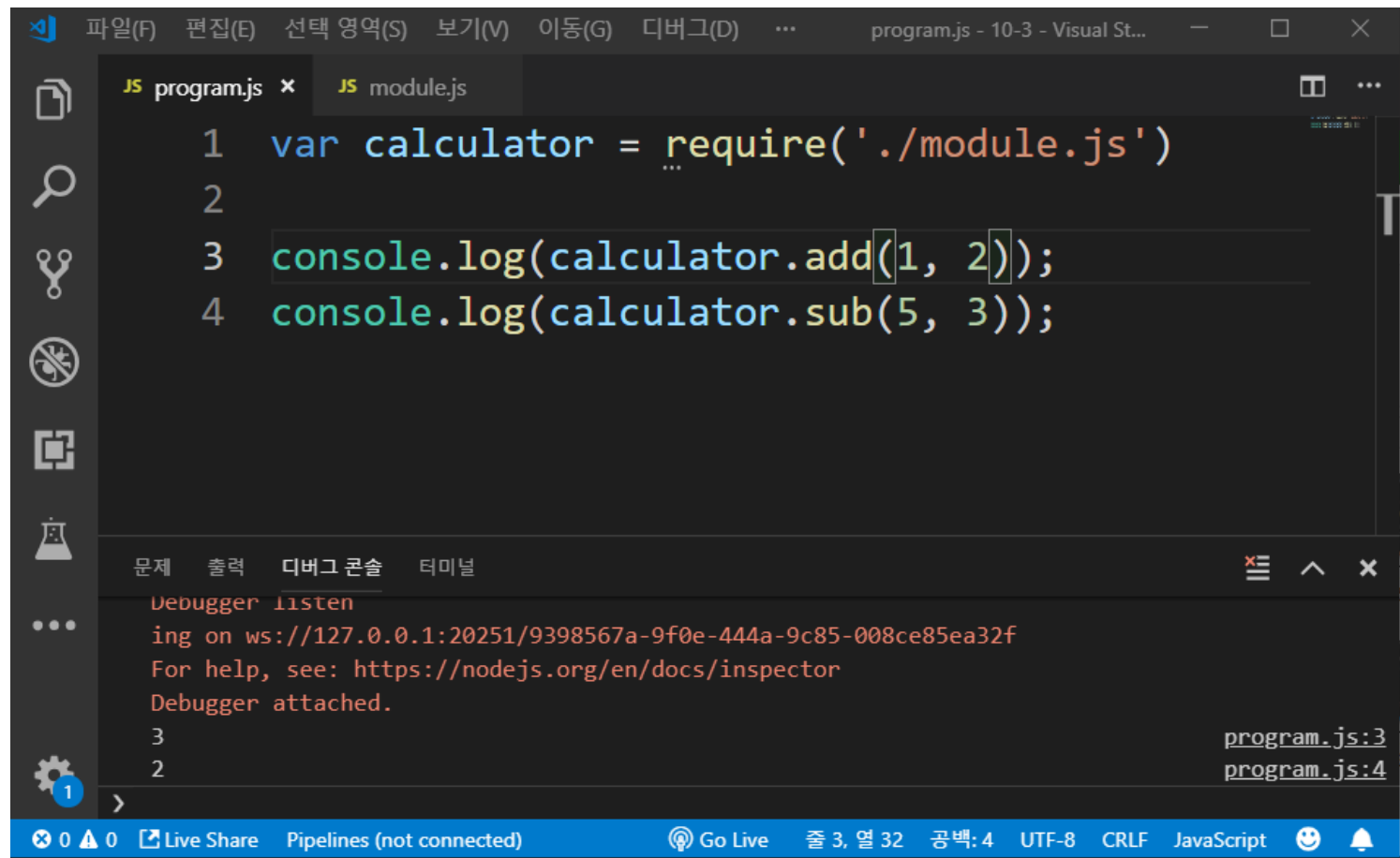
- 여러 파일에 기능을 쪼개서 개발
- 코드 가독성이 올라가고 관리가 쉬워짐
- `module.exports` 객체 이용
- `require('{파일명}')` 함수로 모듈 로드

모듈 프로그래밍



```
1 function add(a, b) {  
2     return a + b;  
3 }  
4  
5 function sub(a, b) {  
6     return a - b;  
7 }  
8  
9 module.exports = {  
10     'add': add,  
11     'sub': sub  
12 }
```

모듈 프로그래밍



The screenshot shows the Visual Studio Code interface with a file named `program.js` open. The code in the editor is as follows:

```
1 var calculator = require('./module.js')
2
3 console.log(calculator.add(1, 2));
4 console.log(calculator.sub(5, 3));
```

Below the editor, the Debug Console is visible, showing the output of the program:

```
Debugger listen
ing on ws://127.0.0.1:20251/9398567a-9f0e-444a-9c85-008ce85ea32f
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
3 program.js:3
2 program.js:4
```

The status bar at the bottom indicates the file is encoded in UTF-8, uses CRLF line endings, and is a JavaScript file. It also shows a "Live Share" button and a "Go Live" button.

요약

- NodeJS 소개 및 설치
- NodeJS 디버깅
- NodeJS 모듈 프로그래밍

차시 예고

- 11-1 : npm과 express 모듈 알아보기
 - Npm 로 모듈 설치
 - express 모듈 사용

강의를 마치겠습니다
수고하셨습니다

10주차_03 NodeJS