

바닐라 자바스크립트

chapter02

개발 환경 구성

제공된 자료는 훈련생의 수업을 돕기 위한 것으로, 타인과 공유하시면 안됩니다.

Contents

part.1

Node.js 설치

part.2

VSCODE 플러그인 설치

part.3

자바스크립트 작성 방법

part.4

스코프

노드란?

Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임 도구입니다.

JavaScript는 프로그래밍 언어이고, Node.js는 JavaScript 코드를 실행하는 도구입니다.

Node.js는 Java의 JRE(자바 런타임 환경)와 비슷한 역할을 합니다.

노드의 특징

- JavaScript는 주로 클라이언트 쪽에서 언어지만, Node.js를 사용하면 JavaScript로 서버 기능까지 구현할 수 있습니다.
- 노드를 사용하면 브라우저나 HTML 없이도 JavaScript를 실행할 수 있습니다.



Node.js 설치

노드란?

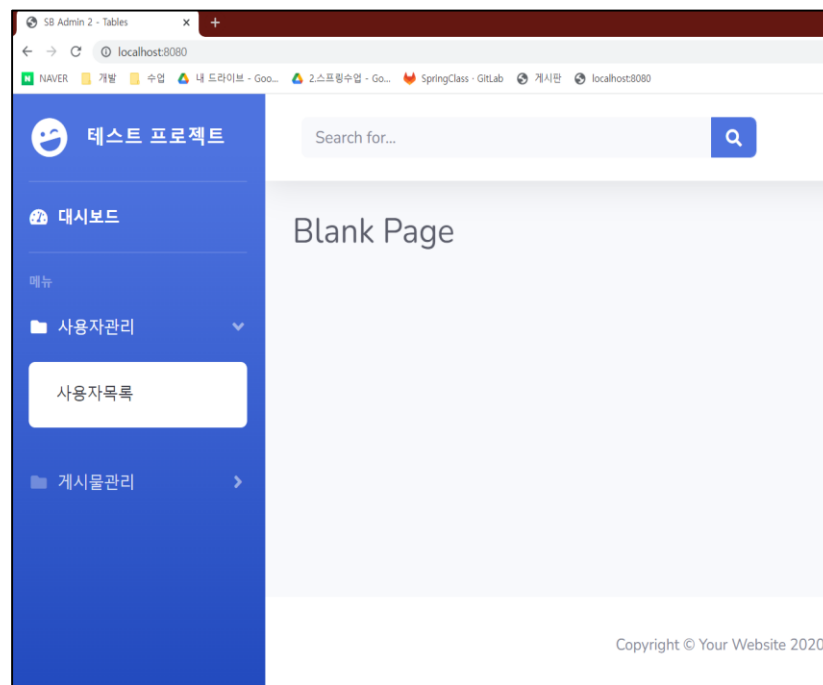
왜 노드를 사용해야 할까?

JavaScript는 원래 브라우저에서 실행되도록 설계된 클라이언트 프로그래밍 언어입니다.

JavaScript는 HTML과 CSS와 함께 화면을 구성하는 역할을 합니다.

하지만 Node.js를 사용하면 JavaScript를 서버에서 실행할 수 있어서, 브라우저 없이도 JavaScript만으로 기능을 구현할 수 있습니다.

*수업에서는 Node.js로 서버를 구축하지 않고, HTML 없이 빠르게 기능을 구현하기 데 사용할 예정입니다.



브라우저에서 HTML, CSS, JavaScript를 함께 실행

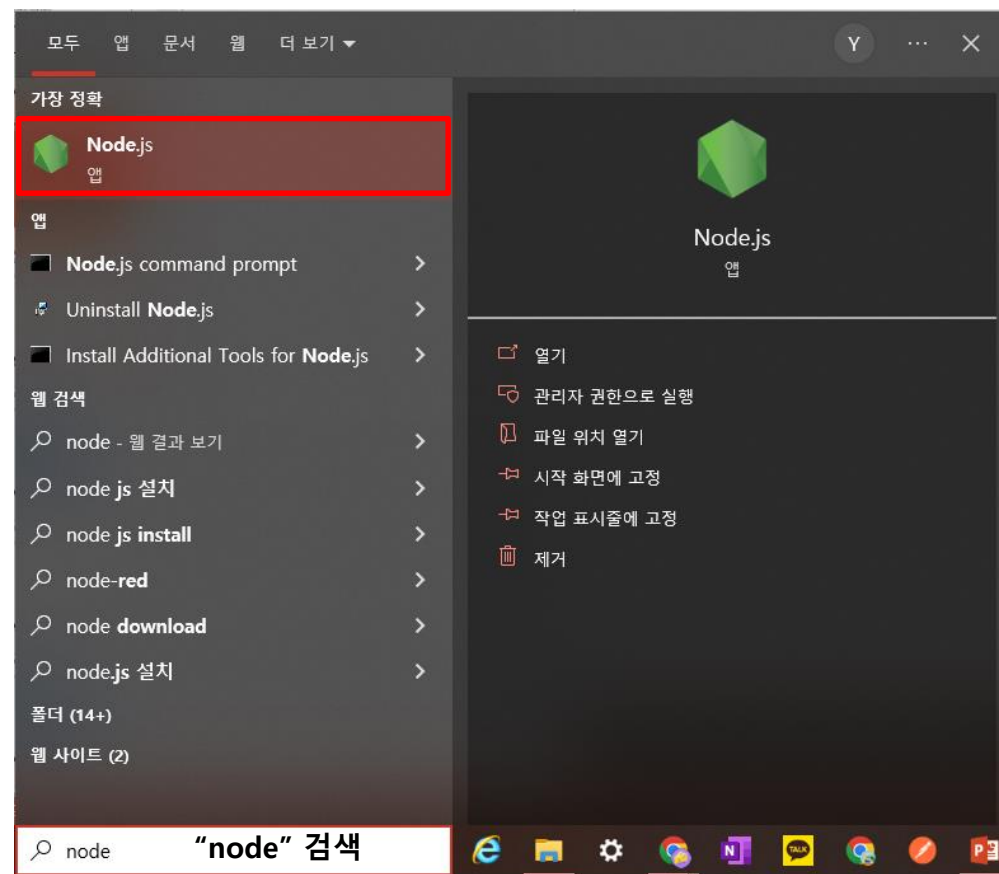
```
<script type="text/javascript">
  $(document).ready(function() {
    var modifyResult = '<c:out value="${modifyResult}"/>';
    checkModifyModal(modifyResult);
    function checkModifyModal(result) {
      if (result === '') {
        return;
      }
      if(result == 'true'){
        $(".modal-body").html("수정을 성공하였습니다");
      }else{
        $(".modal-body").html("수정을 실패하였습니다");
      }
      $("#myModal").modal("show");
    }
  })
</script>
```

JavaScript는 HTML과 함께 동작함

Node.js 설치

노드 설치하기

1. 이미 Node.js가 설치되어 있는지 확인하고, 설치되어 있다면 삭제하세요.
삭제한 경우, 컴퓨터 재부팅 후 진행하세요.



프로그램 제거 또는 변경



Node.js 설치

노드 설치하기

2. Node.js 설치 파일 다운로드 하고 설치하세요.



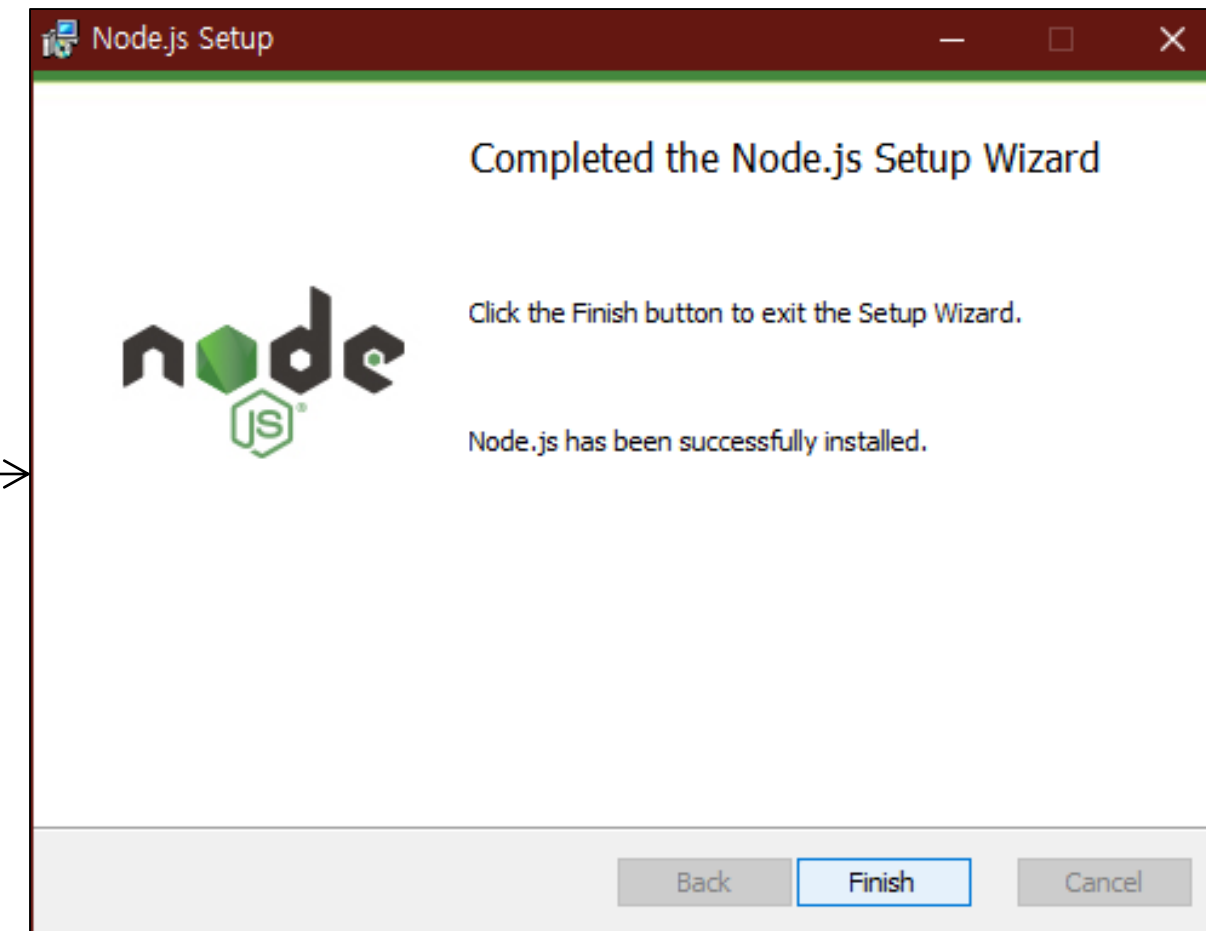
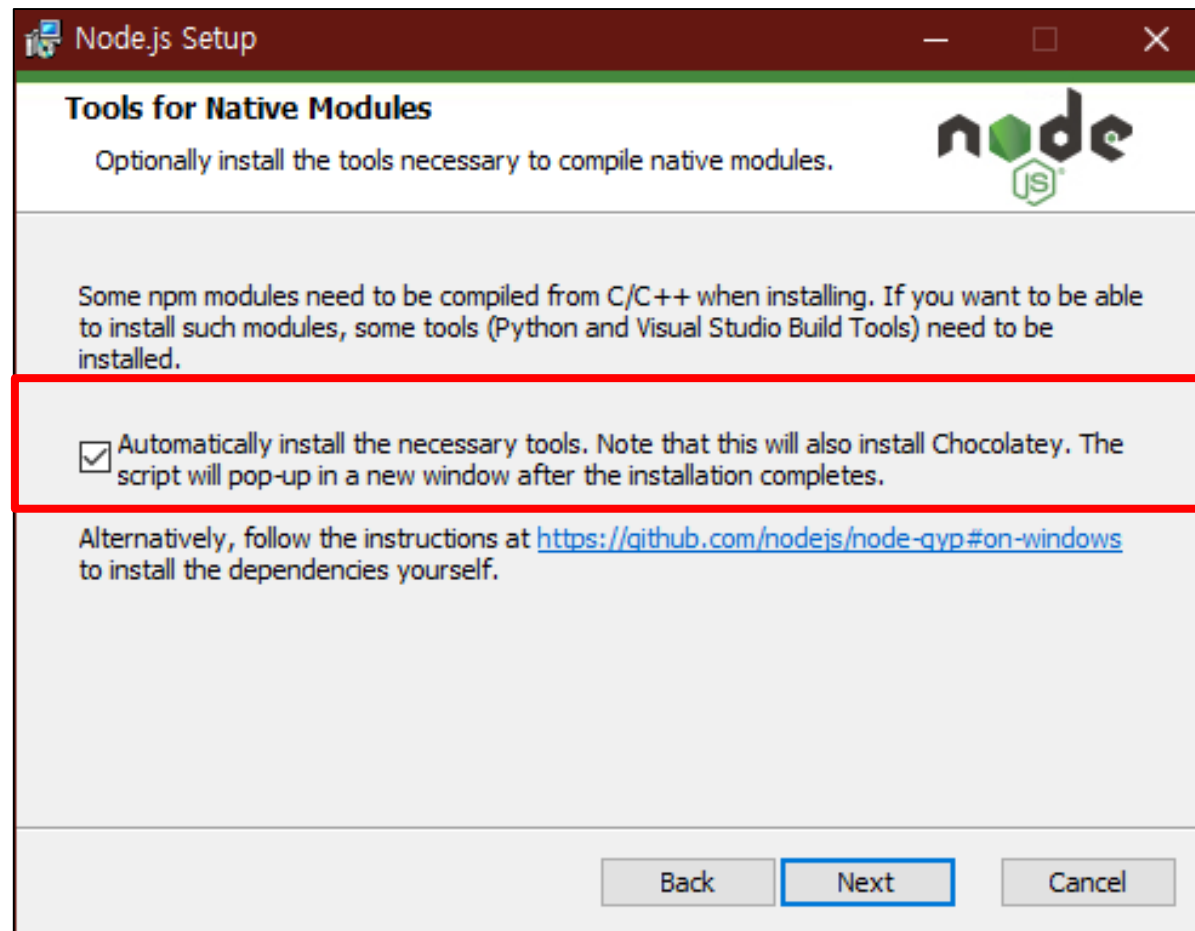
<https://nodejs.org>



Node.js 설치

노드 설치하기

3. 설치 과정 중간에 나오는 체크박스를 선택합니다.
체크하면 부가적인 프로그램들이 함께 설치됩니다.



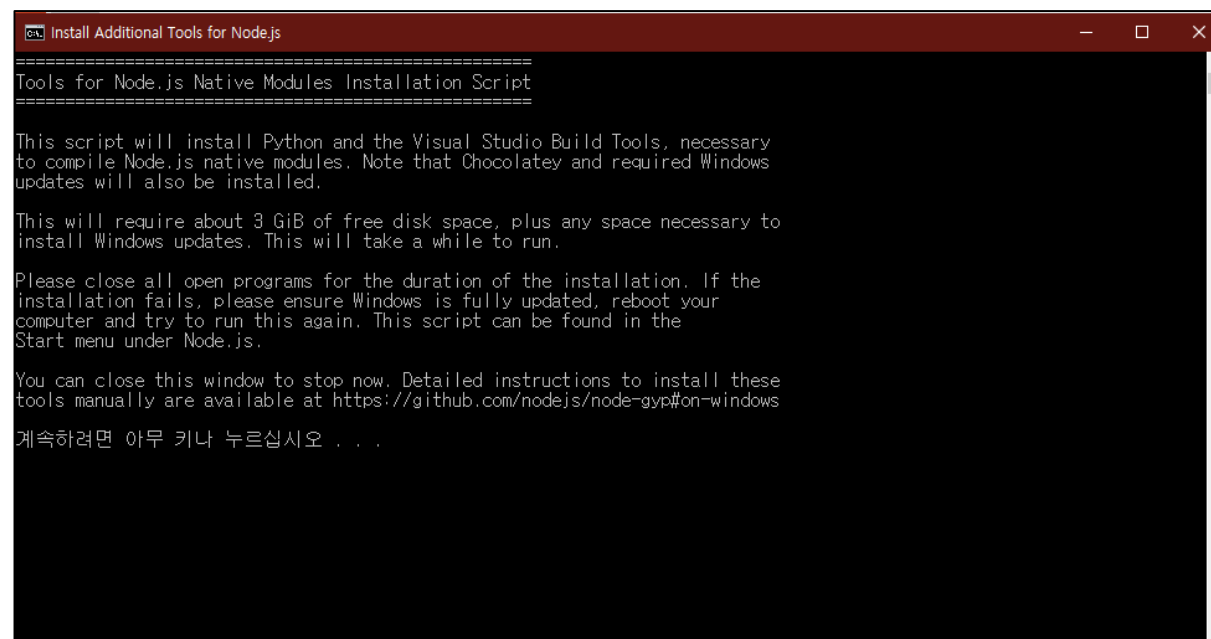
*아직 설치 안끝남

Node.js 설치

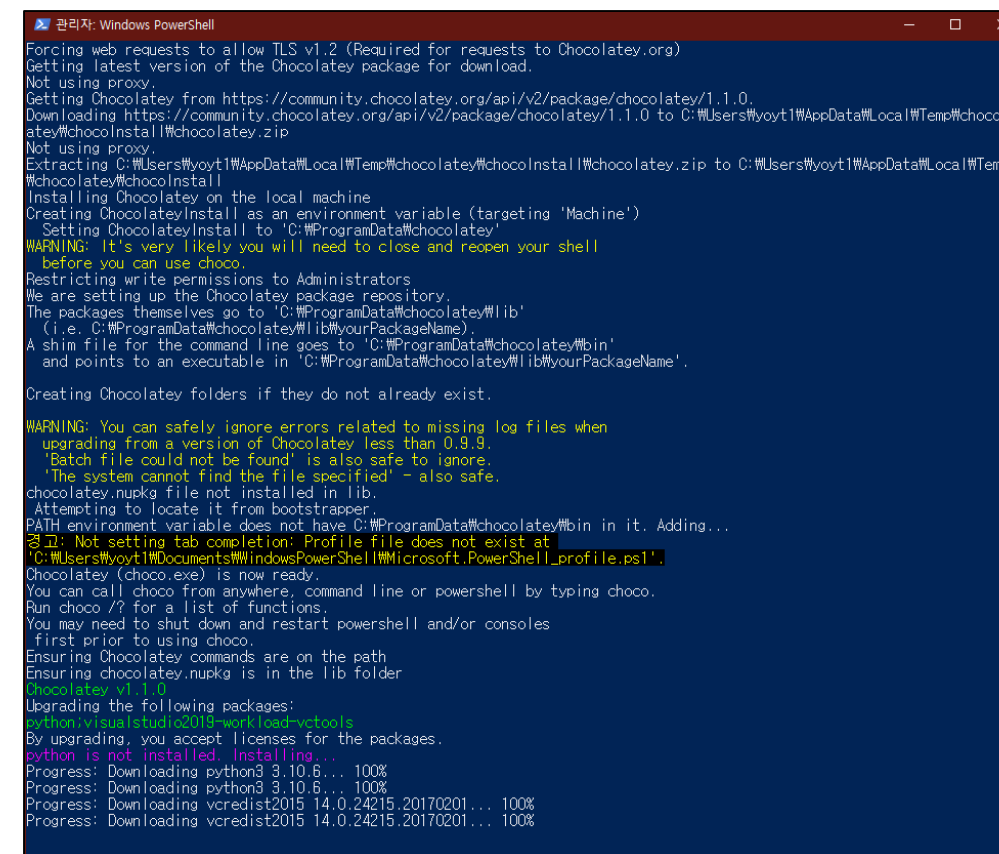
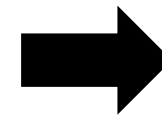
노드 설치하기

4. CMD창이 열리면, 엔터를 두 번 눌러주세요.

이후, 파워셸이 실행되면 필요한 프로그램들이 자동으로 설치됩니다.



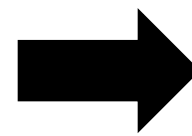
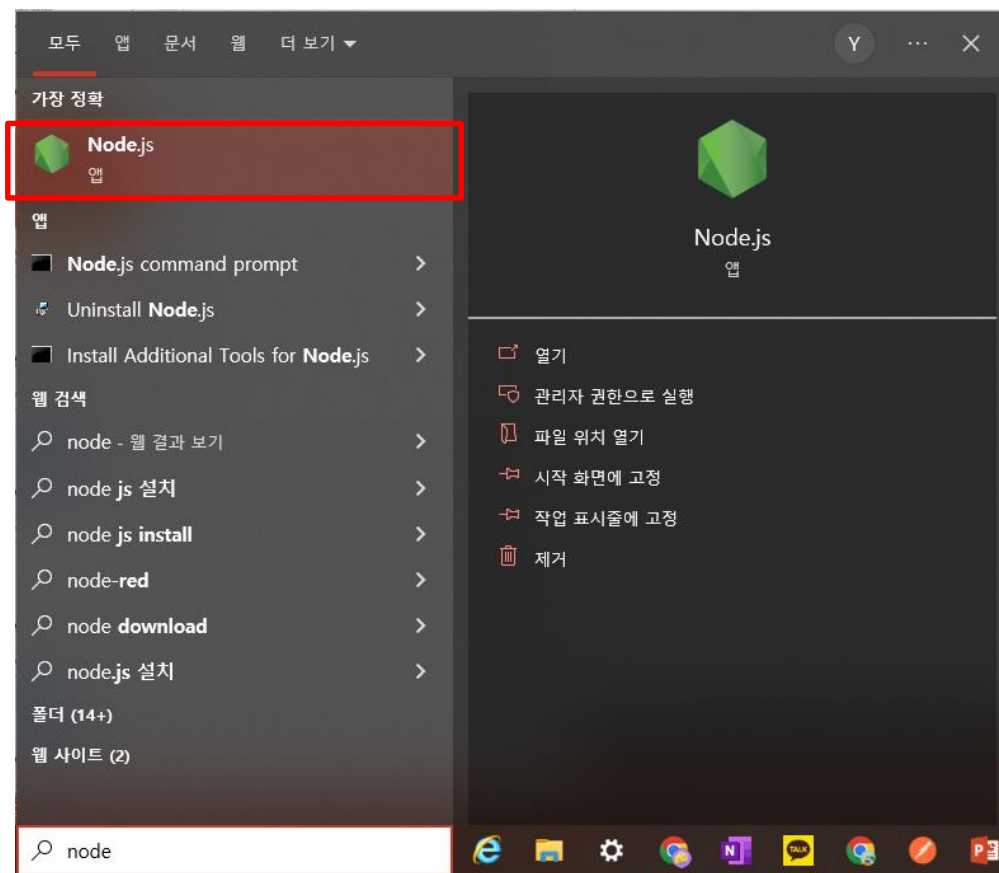
엔터.. 한번더 엔터..



Node.js 설치

노드 설치하기

5. 서비스 메뉴에서 Node.js가 정상적으로 설치되었는지 확인하세요.
그리고 CMD창을 열고 "node -v" 명령어를 입력해 버전을 확인하세요.
Node.js의 버전이 출력되면, 설치가 정상적으로 완료된 것입니다.



```
명령 프롬프트
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\imjiyeon>node -v
v20.16.0

C:\Users\imjiyeon>
```

CMD창

Node.js 설치

테스트

빈 폴더(프로젝트)를 생성한 후, VS Code에서 폴더를 여세요.
JavaScript 파일 하나를 생성하고, 간단한 코드를 작성하세요.
그런 다음 터미널에서 "node test" 명령어를 입력하세요.
그러면 JavaScript 파일이 실행됩니다.
또는 F5 단축키를 눌러 실행할 수도 있습니다.

```
JS test.js  U X
2.개발 환경 구성 > JS test.js
1  // node.js 테스트
2
3  console.log("node.js 테스트");
```

```
문제  출력  디버그 콘솔  터미널  포트
PS D:\OneDrive\8.자바스크립트\vs-workspace\front-study\JS\2.개발 환경 구성 > node test
node.js 테스트
```

디버깅 시작	F5
디버깅 없이 실행	Ctrl+F5

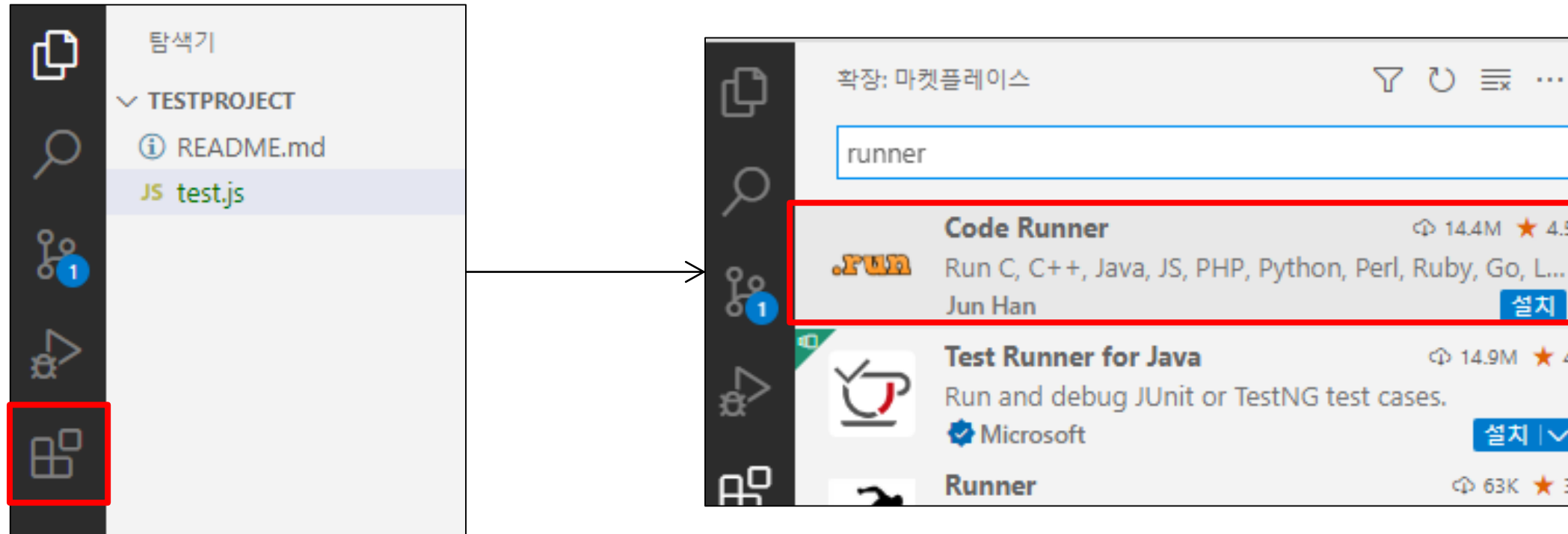
플러그인 설치

Runner 플러그인 설치하기

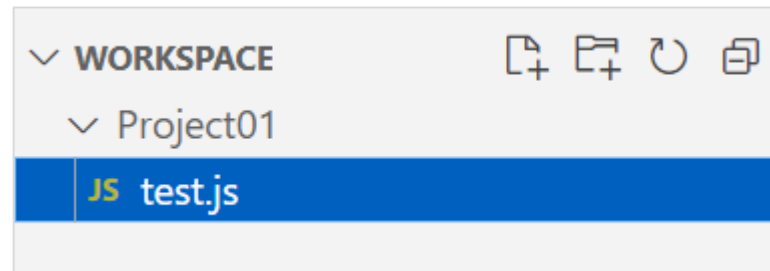
Node.js 대신 VSCode의 Runner 플러그인을 사용하셔도 됩니다.

이 플러그인은 다양한 언어를 실행할 수 있는 기능을 제공하며, JavaScript도 실행할 수 있습니다.

1. 마켓 플레이스에서 "runner"를 검색한다.
2. runner 플러그인을 설치한다.



1. test.js 파일을 열고, 메시지를 변경합니다.
2. 버튼을 클릭하여 JavaScript를 실행합니다.
3. 콘솔창에 메시지가 출력되었는지 확인하세요.

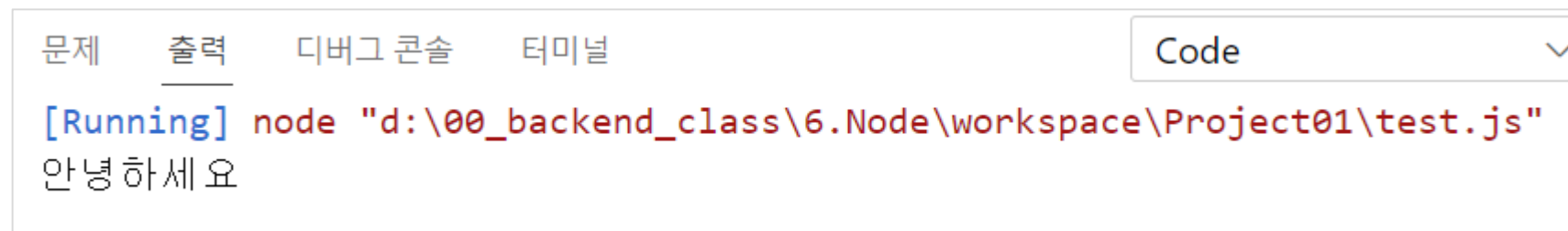


```
JS test.js  U X
2.개발 환경 구성 > JS test.js
1  // runner 테스트
2
3  console.log("runner 테스트");
```

코드 실행



결과



작성 방법

코드 작성하고 파일 실행하기

HTML에서 자바스크립트를 작성할 수 있는 방법은 두가지 입니다.

1. head 태그 안에 script 태그를 추가하고, script 태그 안에 작성한다.

```
<html lang="ko">
  <head>
    <script>
      |   console.log("안녕하세요");
    </script>
  </head>
```

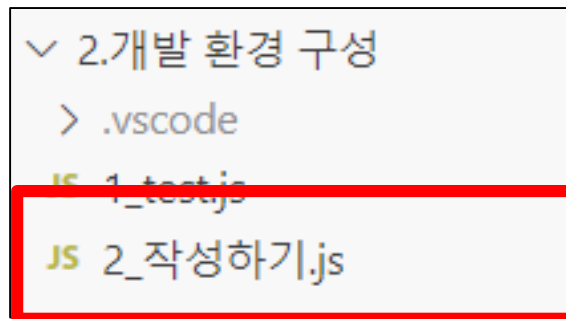
2. 자바스크립트 파일을 별도로 만들고, script 태그로 연결한다.

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Document</title>
  <script src="./2_작성하기.js"></script>
</head>
```

2_작성하기.js

```
console.log("안녕하세요");
```

프로젝트에 "2_JS작성하기.js" 파일을 추가하세요.
그리고 JS 파일에 다음과 같이 내용을 입력하세요.



```
console.log('안녕하세요');
```

그리고 index.html 파일을 추가하세요.
<script> 태그를 만들고, src 속성에 JS 파일의 경로를 작성하여 연결하세요.



```
<head>  
  <script src="./2_JS작성하기.js"></script>  
</head>
```

작성 방법

코드 작성하고 파일 실행하기

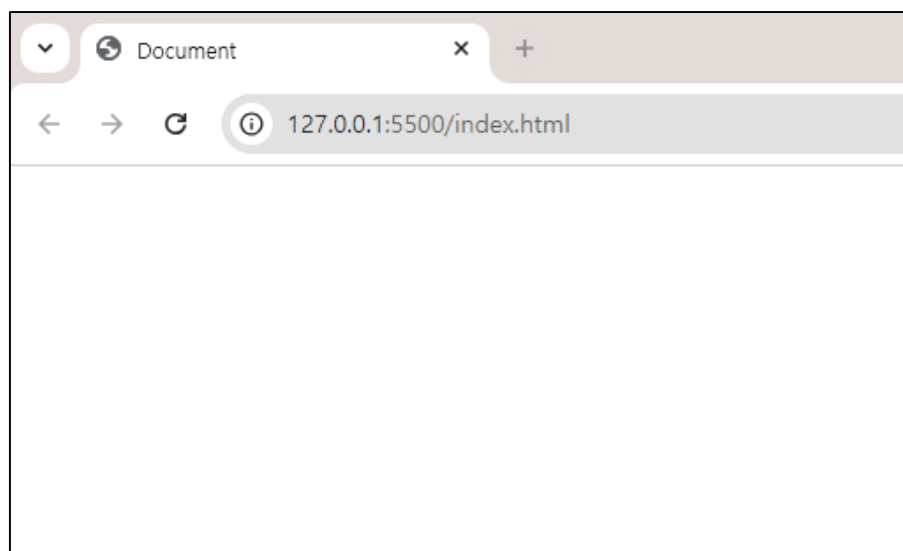
index.html을 선택하고 우클릭하세요.

그리고 'Open with Live Server' 메뉴를 선택하여 HTML 파일을 실행하세요.



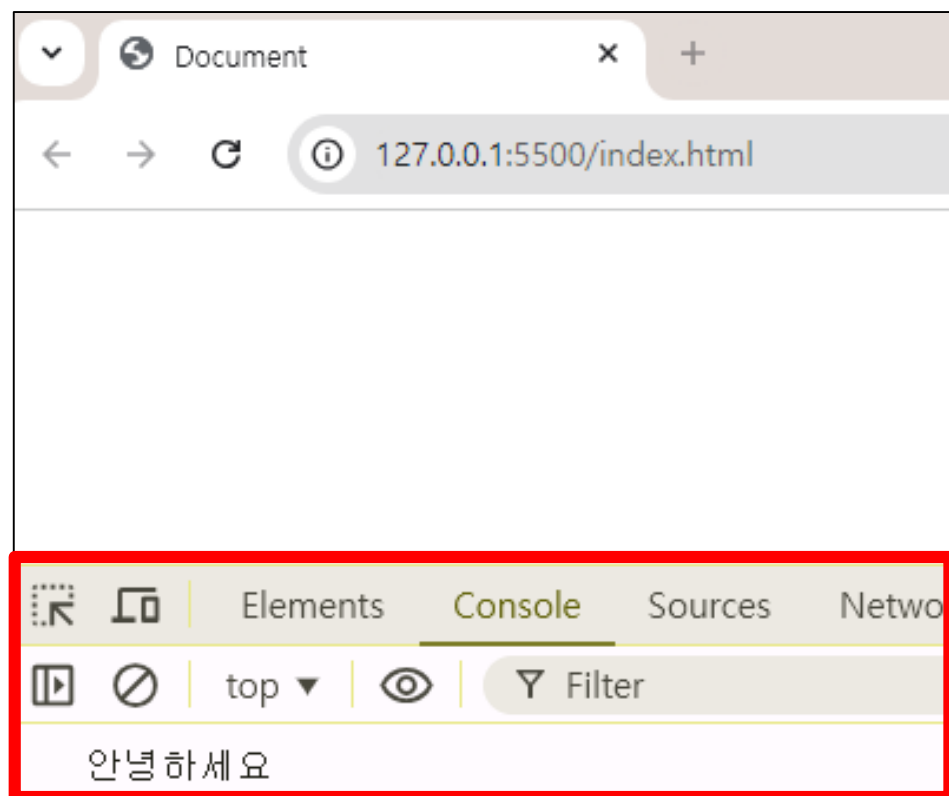
Open with Live Server

<Alt>+L <Alt>+O



브라우저에서 F12키를 눌러서 개발자 도구를 여세요.

그리고 Console탭을 선택하여 메시지가 정상적으로 출력되었는지 확인하세요.



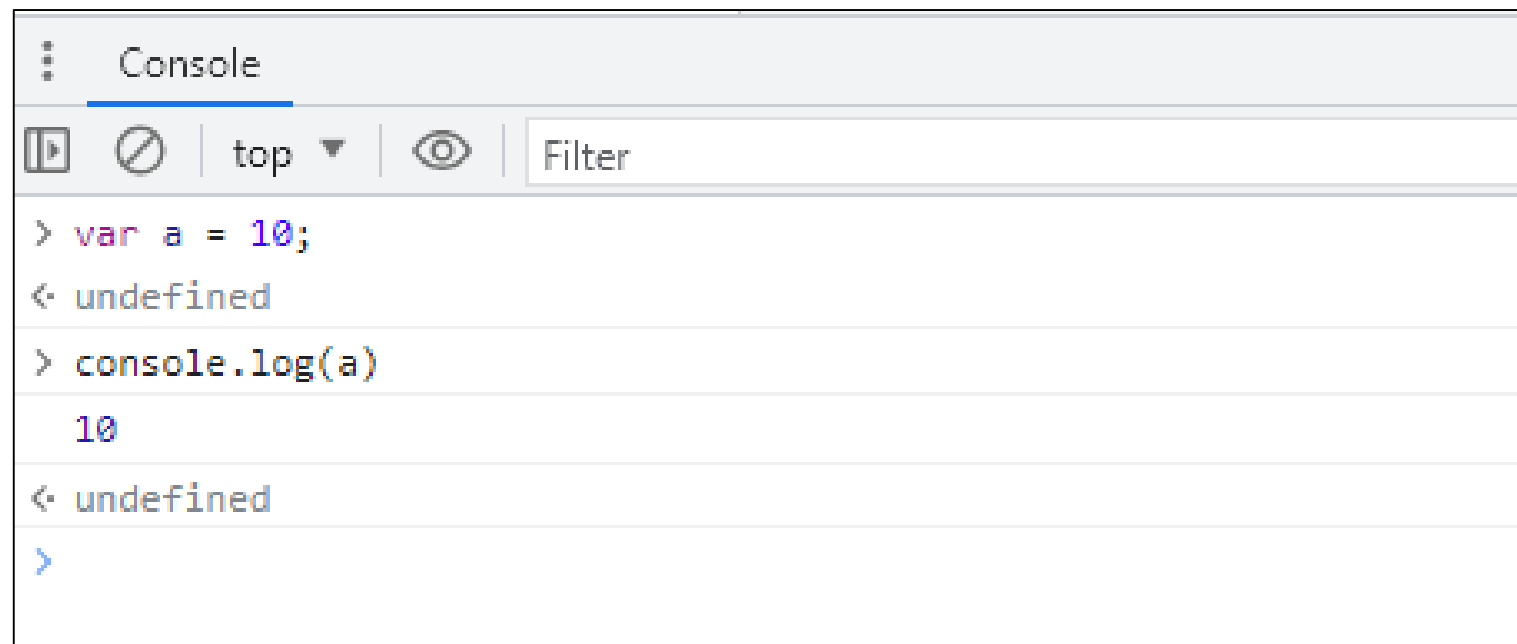
개발자도구에는 여러가지 탭이 있는데, 그중 Console탭을 사용하면 여러가지 기능을 사용할 수 있습니다.

코드 실행하기

콘솔탭을 보면 > 기호와 함께 커서가 깜박거리는 것을 볼 수 있습니다.

여기서 JavaScript 코드를 실행 할 수 있습니다.

다만 엔터키로 줄바꿈이 되지 않기 때문에 복잡한 코드를 작성하기 어렵습니다.



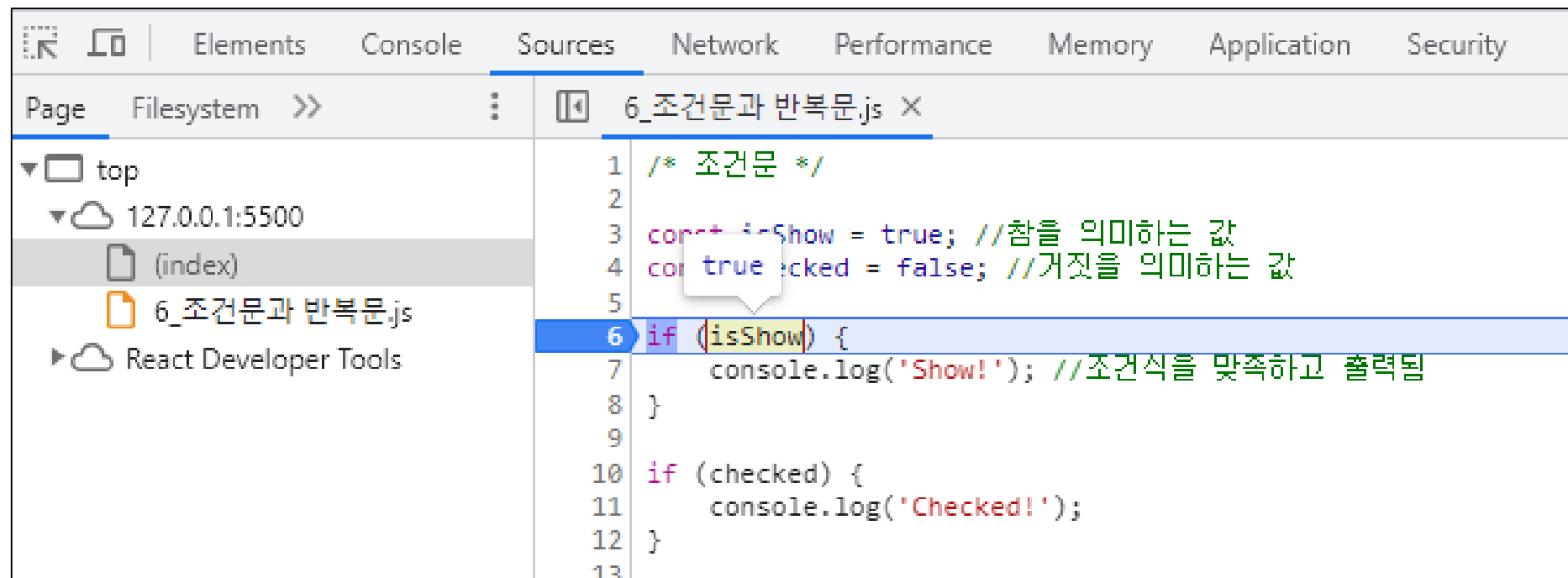
The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The interface includes a toolbar with icons for running, pausing, and filtering, along with a 'top' dropdown and a 'Filter' input. The console log shows the following sequence of operations:

```
> var a = 10;  
< undefined  
> console.log(a)  
10  
< undefined  
>
```

코드 디버깅

개발자도구에서는 JavaScript를 디버깅도 할 수 있습니다.

이 과정에서 코드를 디버깅하면서 변수의 값을 확인 할 수 있습니다.



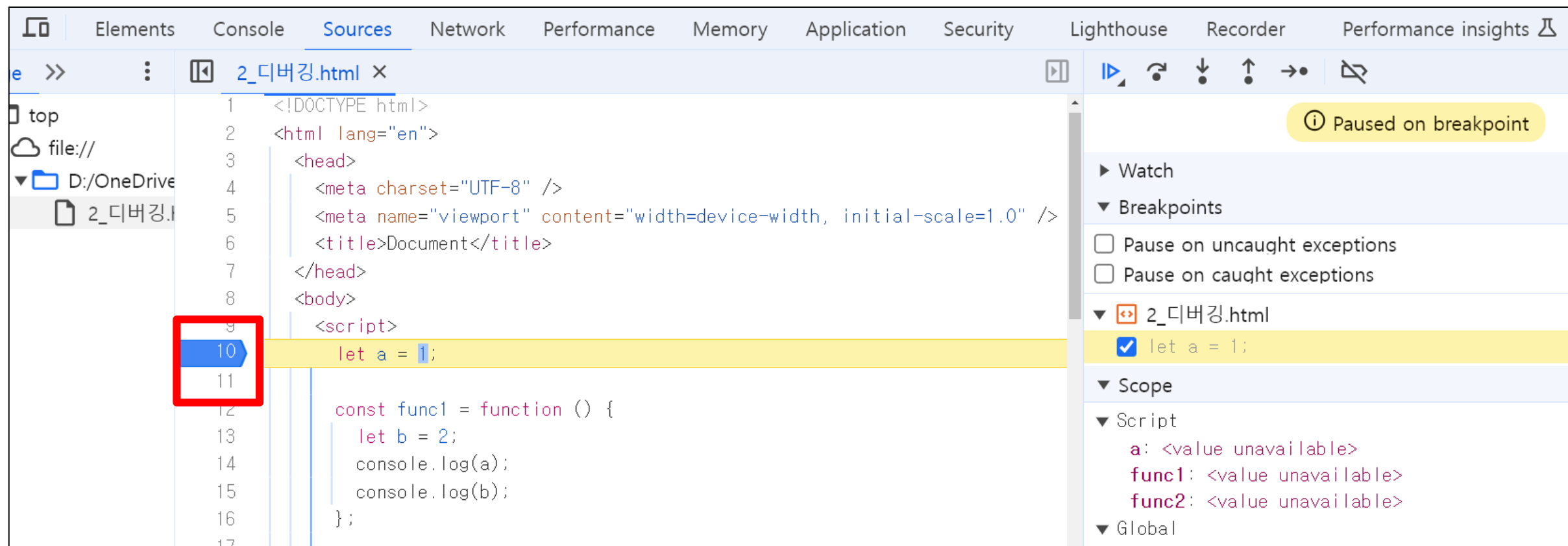
자세한 디버깅 방법

Source탭에서 자바스크립트 파일을 열어주세요.

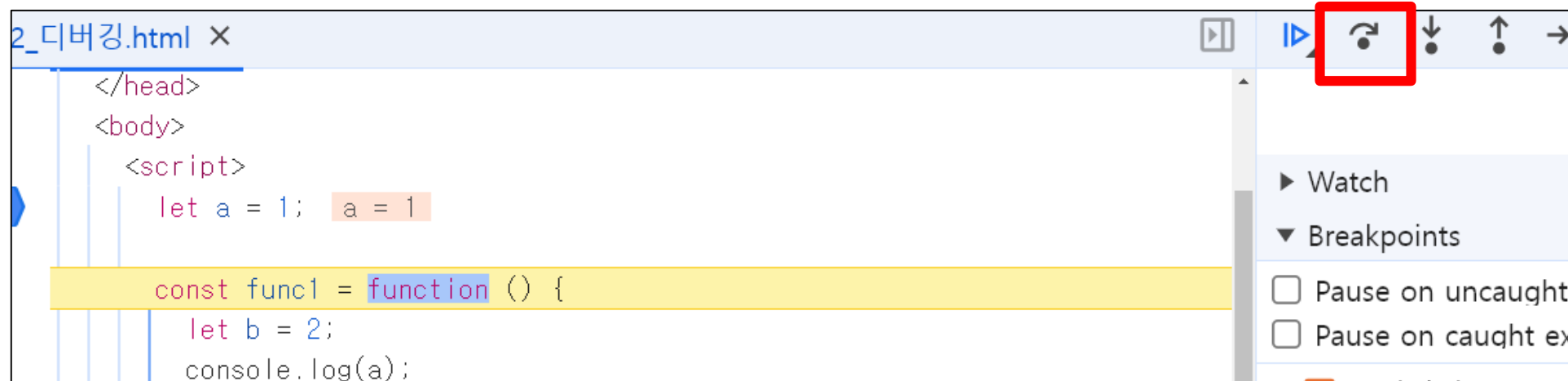
파일에서 라인을 클릭하여 브레이크 포인트를 설정하세요.

그리고 새로고침(F5)을 하여 화면을 다시 불러옵니다.

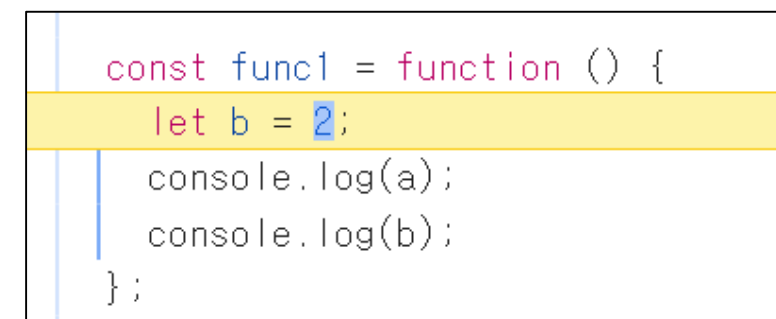
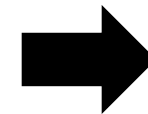
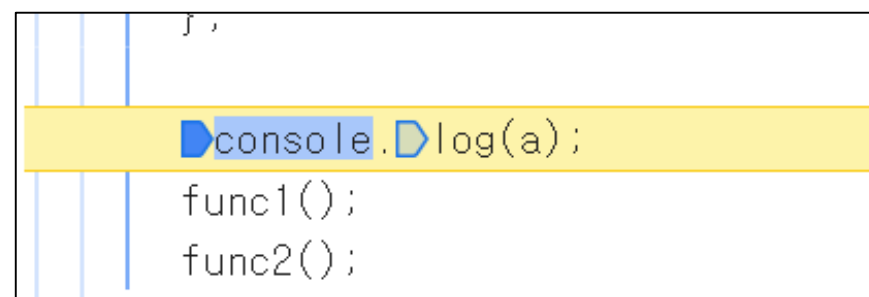
화면이 나타나면 브레이크 포인트가 작동합니다.



2번째 버튼을 클릭하면 다음 라인이 실행됩니다.



3번째 버튼을 클릭하면 함수 내부로 이동합니다.



JavaScript에는 스코프(scope)라는 개념이 있습니다.
스코프는 전역 스코프와 지역 스코프로 구분됩니다.

```
<script>
```

```
let a = 1;
```

전역스코프

```
const func1 = function () {
```

지역스코프

```
let b = 2;
```

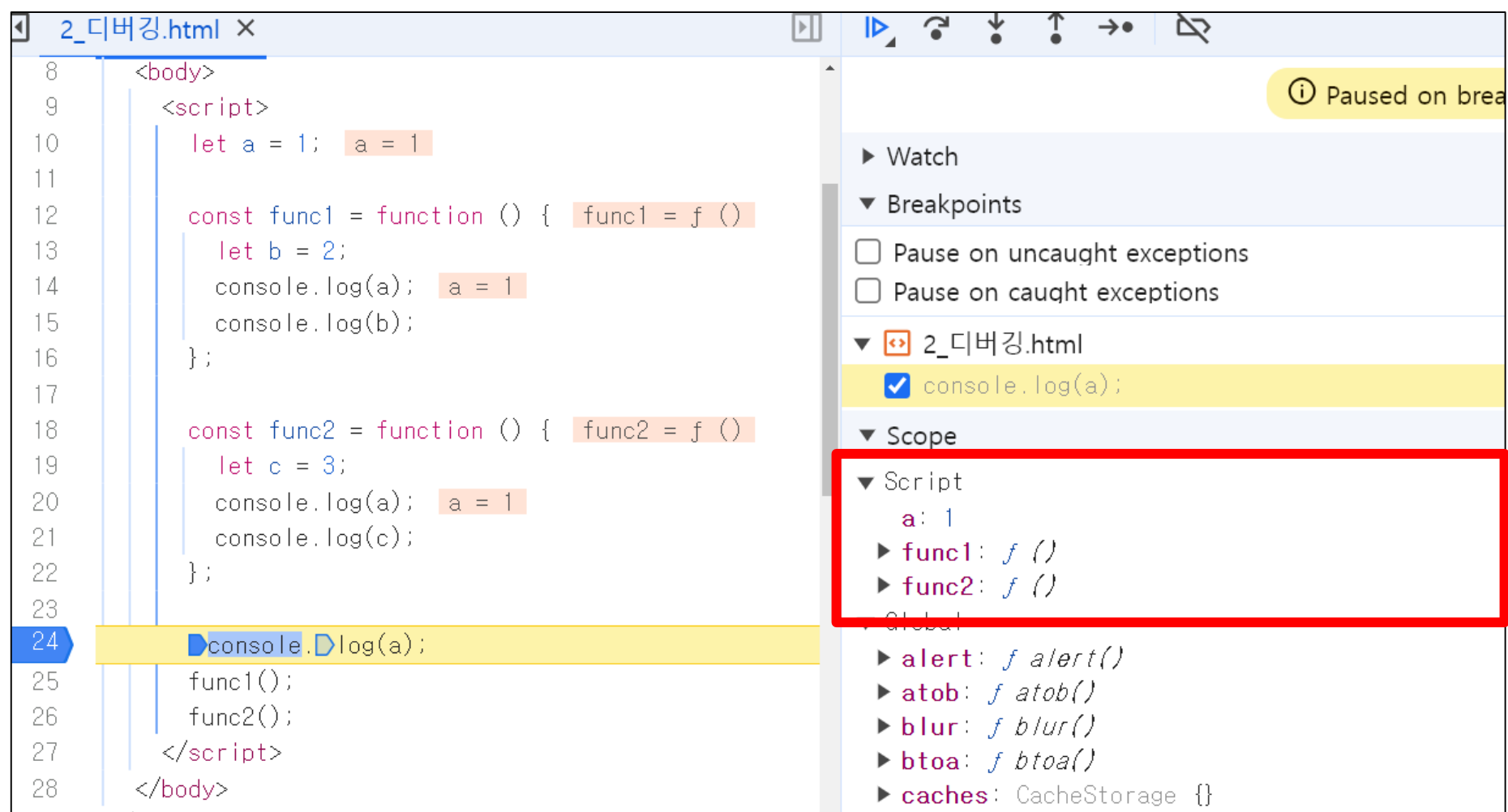
```
console.log(a);
```

```
console.log(b);
```

```
} .
```

스크립트는 전역 스코프이며, 전역 변수가 표시됩니다.

현재 전역변수로는 변수a, 함수func1, 함수func2가 있습니다.



Global도 전역 스코프이며, window객체가 표시됩니다.
브라우저에서 JavaScript를 실행하면 window객체가 생성됩니다.
이 객체가 가지고 있는 변수와 함수를 확인 할 수 있습니다.



스코프

지역 스코프

지역 스코프는 함수 스코프와 블록 스코프가 있습니다.
함수 내에 선언된 변수들은 지역 스코프에 생성됩니다.
지역 스코프의 this는 window객체를 가리킵니다.

