



사용자 인증을 통한 게시판 만들기

|INTRODUCE



김시훈

- 세종대학교 컴퓨터공학과
- shuni93@naver.com



황성영

- 광운대학교 컴퓨터공학과
- sy930503@naver.com

|INTRODUCE

<https://drive.google.com/drive/folders/1sIP2XBtCLq-6mmjkKd1IXP8pSmthiA2K>

|INDEX

- ▶ #1 - Node.js의 소개 및 환경설정 / 간단한 웹 애플리케이션 만들기
- ▶ #2 - Express 모듈을 통한 프로젝트 생성 / Route / rest API
- ▶ #3 - Database(Mysql) 연동 / 회원가입 & 로그인
- ▶ #4 - CRUD 게시판 구현

|NodeJS



|NodeJS

1

단일 스레드 기반의 비동기 I/O처리에 의한 매우 빠른 고성능 서버

- CPU가 I/O응답을 기다리는 시간이 필요 없고, 대부분의 연산 작업에 사용되기 때문에 높은 효율성을 가질 수 있다.

2

자바스크립트를 이용한 생산성 향상

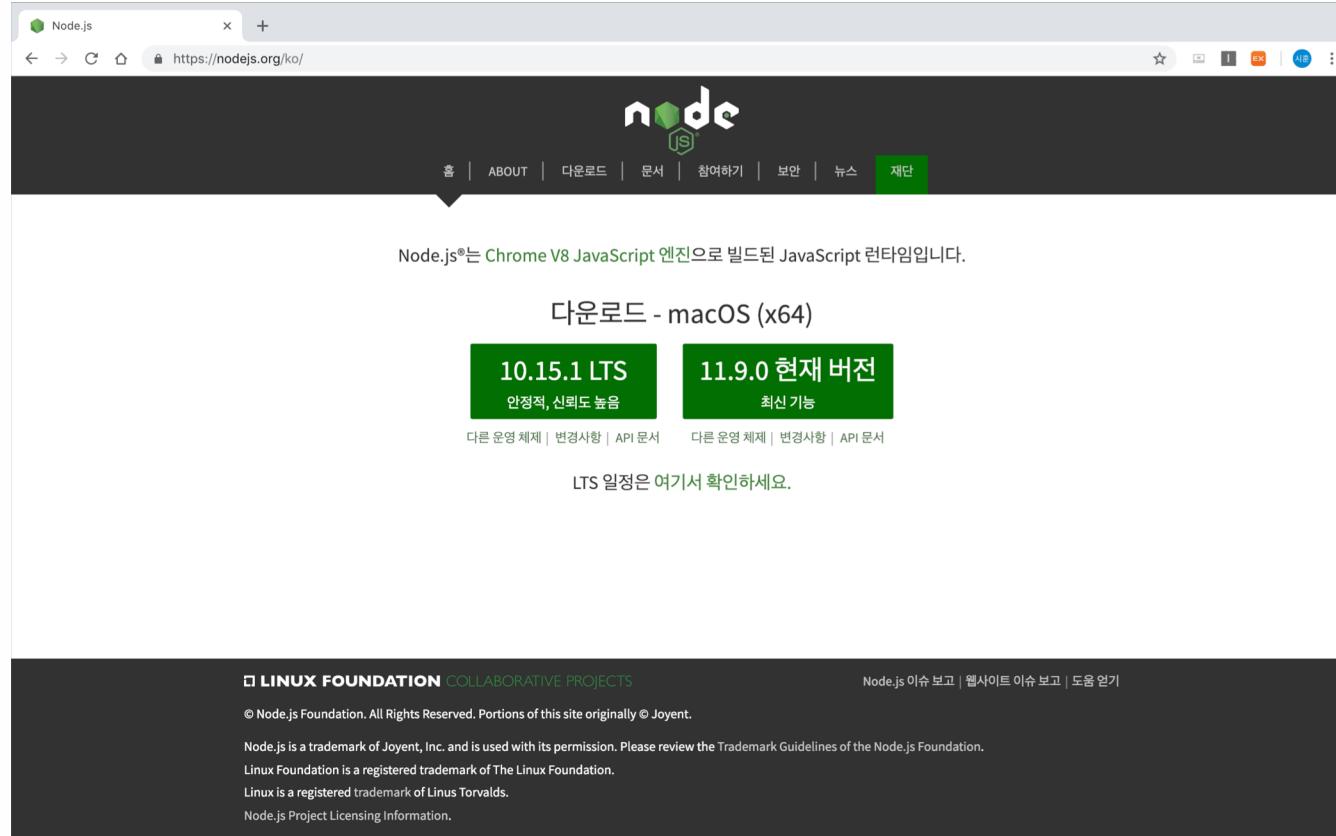
- 멀티 스레드가 아니므로 서버 프로그래밍이 단순하며, Javascript로 Backend 개발 가능

3

Socket.io 모듈을 사용한 브라우저 호환성 증가

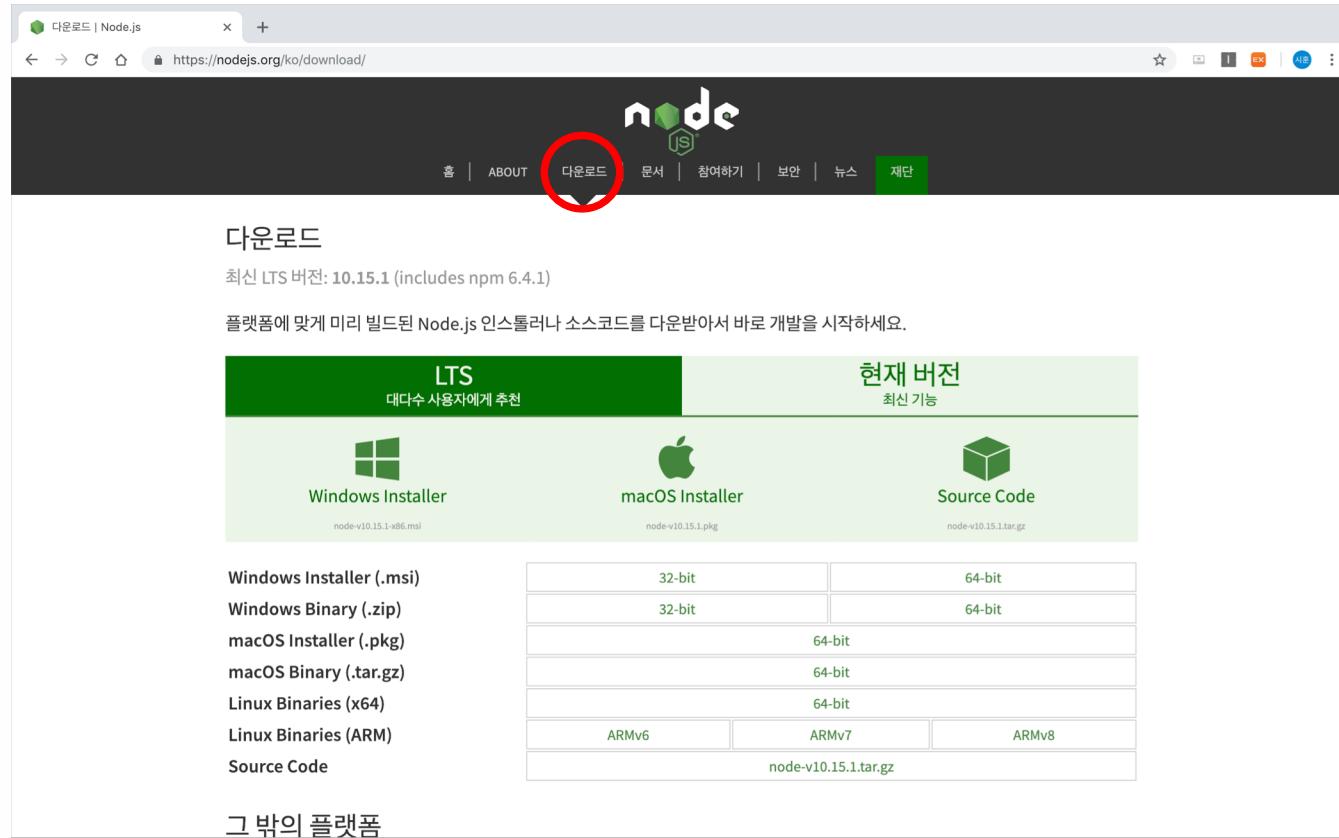
- 브라우저에 상관없이 개발자가 쉽게 푸시 구현

|SETTING



<https://nodejs.org/ko>

SETTING



다운로드 선택 후 OS에 맞게 설치

|SETTING

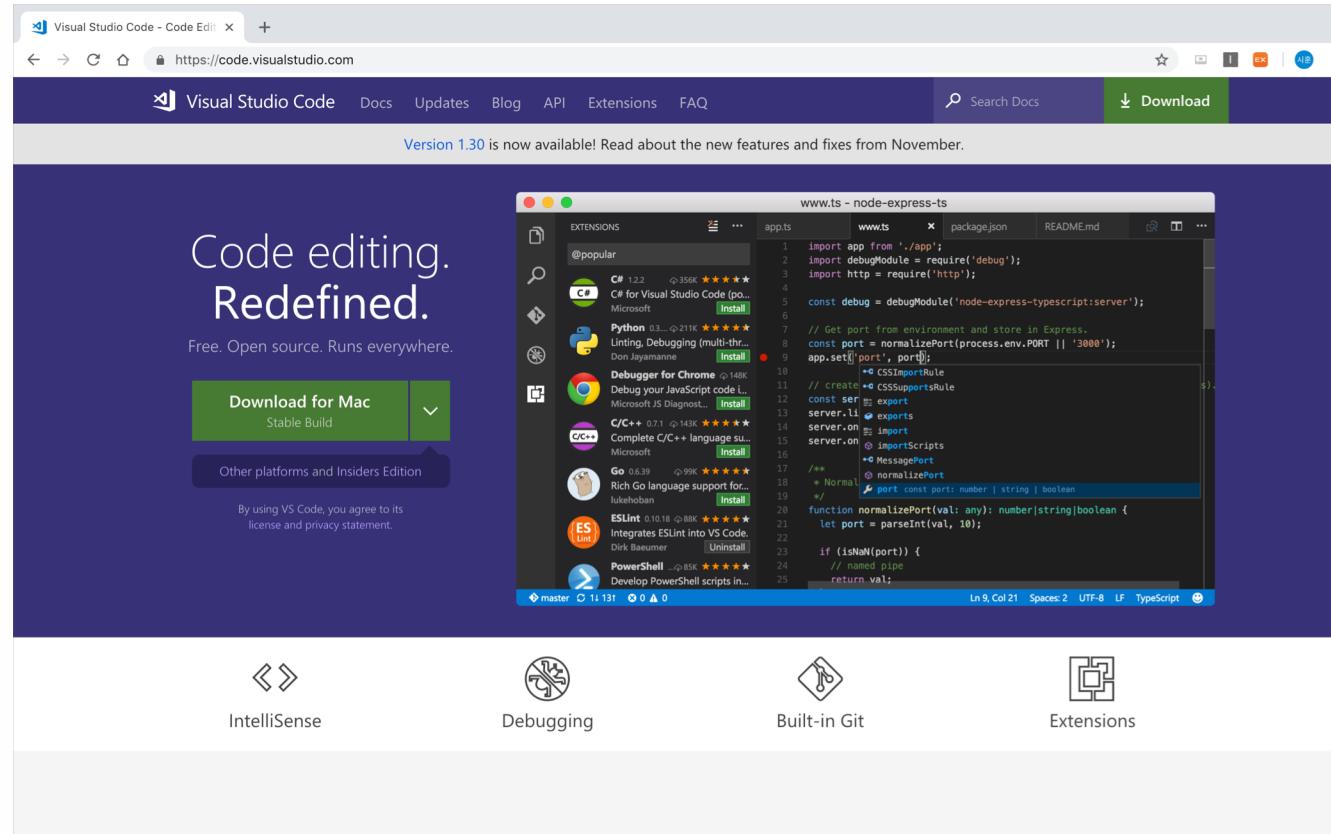
```
$node --version
```

버전이 나오면 설치완료

```
$mkdir nodejs
```

기본 경로에 nodejs라는 폴더 생성

SETTING



<https://code.visualstudio.com>

A screenshot of a Mac OS X terminal window. The window has a dark grey header bar with three colored window control buttons (red, yellow, green) on the left. The main body of the window is black and contains white text. At the top, it shows the command '\$node test.js'. Below that, the output 'Hello World!!' is displayed, preceded by a right-pointing arrow '>'.

```
$node test.js  
>Hello World!!
```

nodejs라는 폴더에 test.js파일을 만든 후 Node.js를 통해 실행시켜 cmd(terminal)창에 Hello World!! 띄우기

|ANSWER

A screenshot of a terminal window. The title bar shows three colored window control buttons (red, yellow, green). The main area has a dark background. On the left, there is a grey sidebar containing a yellow icon with 'JS' and the text 'test.js'. To the right of the sidebar, the code 'console.log("Hello World!!");' is written in white text.

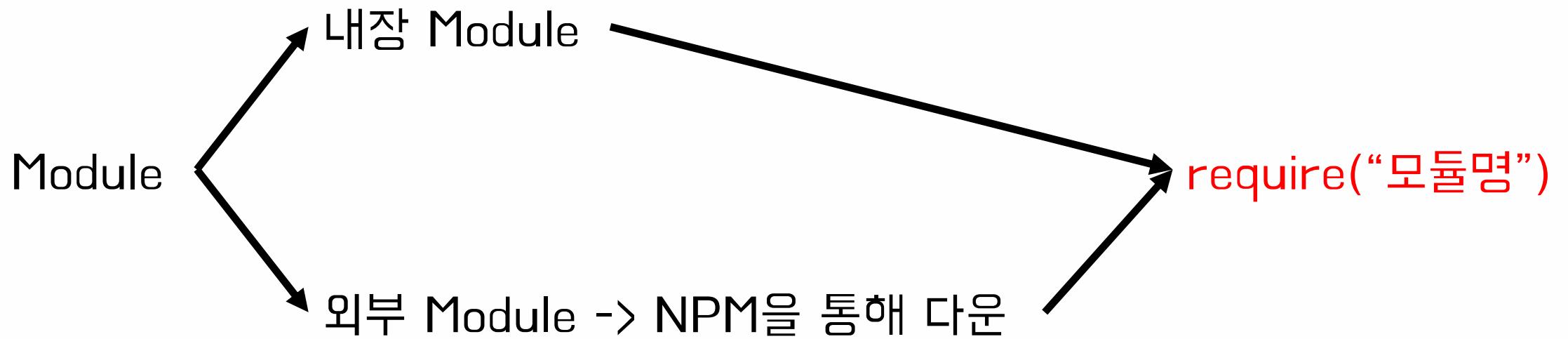
```
JS test.js
console.log("Hello World!!");
```

A screenshot of a terminal window. The title bar shows three colored window control buttons (red, yellow, green). The main area has a dark background. It displays a command '\$node test.js' followed by the output '>Hello World!!' in white text.

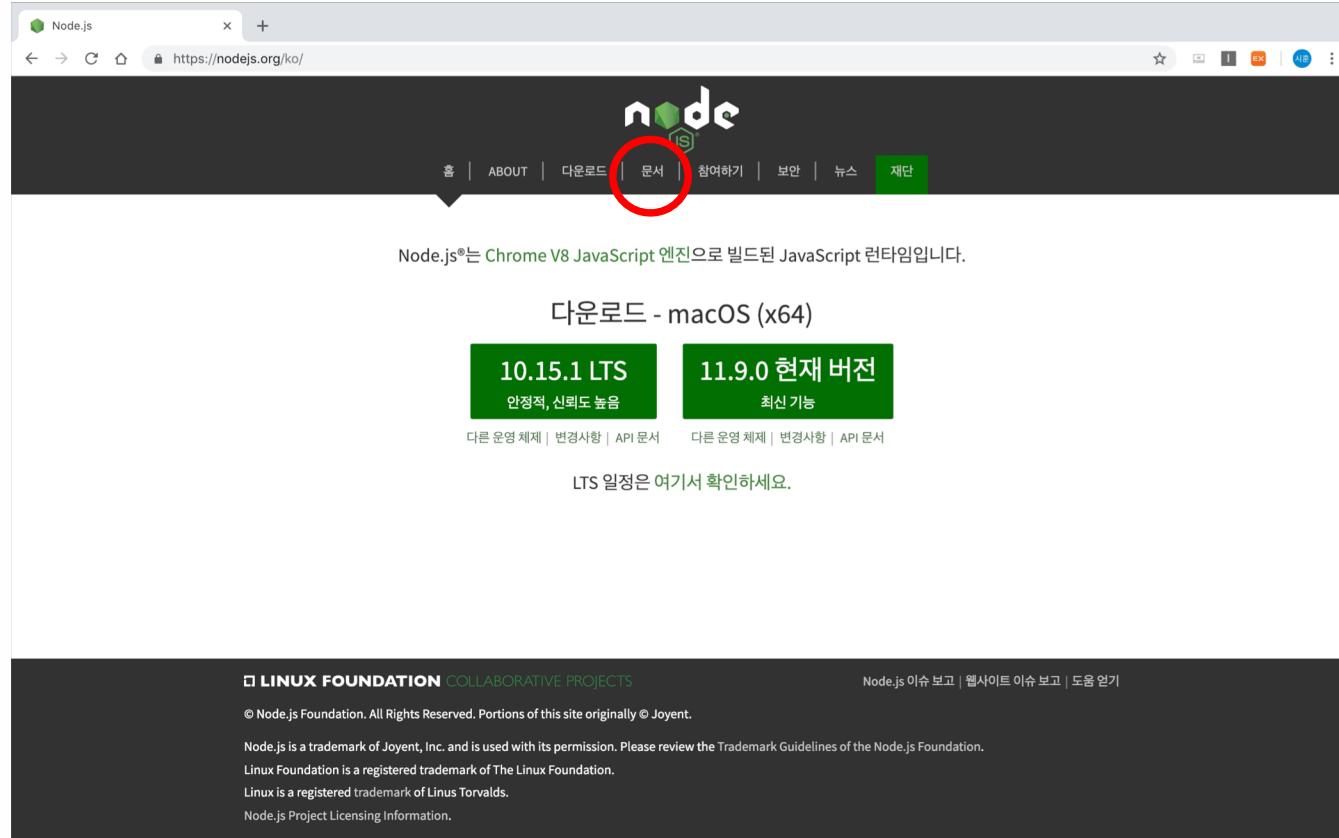
```
$node test.js
>Hello World!!
```

| LEARN

자바 -> 미리 만들어 놓고 사용하는 라이브러리 = API
Node.Js -> **Module(모듈)**



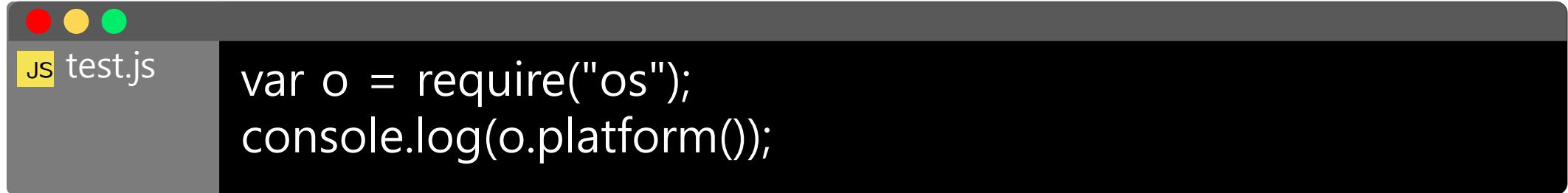
| LEARN



<https://nodejs.org/ko> -> 문서(DOCS)선택
버전별로 Module리스트 & Module 상세설명

OS Module 문서를 보고
test.js에 모듈을 추가하고 node실행을 통해
운영체제 플랫폼 출력 해보기

| ANSWER



A screenshot of a terminal window on a Mac OS X system. The window title bar is dark grey with red, yellow, and green close buttons. The main area of the terminal is black. On the left, there is a grey sidebar containing the text "JS test.js". To the right of the sidebar, the code is displayed:

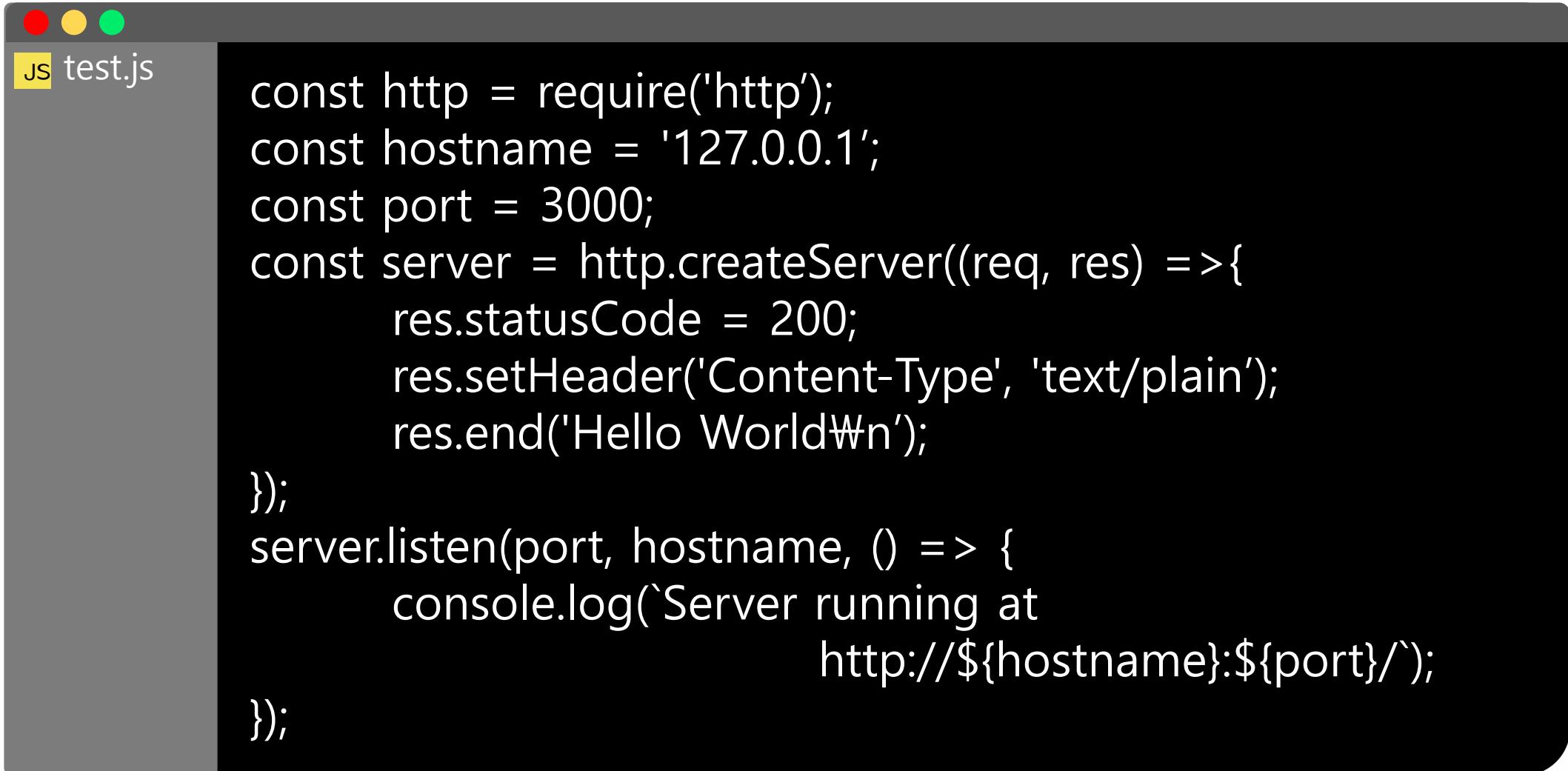
```
var o = require("os");
console.log(o.platform());
```



A screenshot of a terminal window on a Mac OS X system. The window title bar is dark grey with red, yellow, and green close buttons. The main area of the terminal is black. The command "\$node test.js" is entered, followed by the output ">darwin".

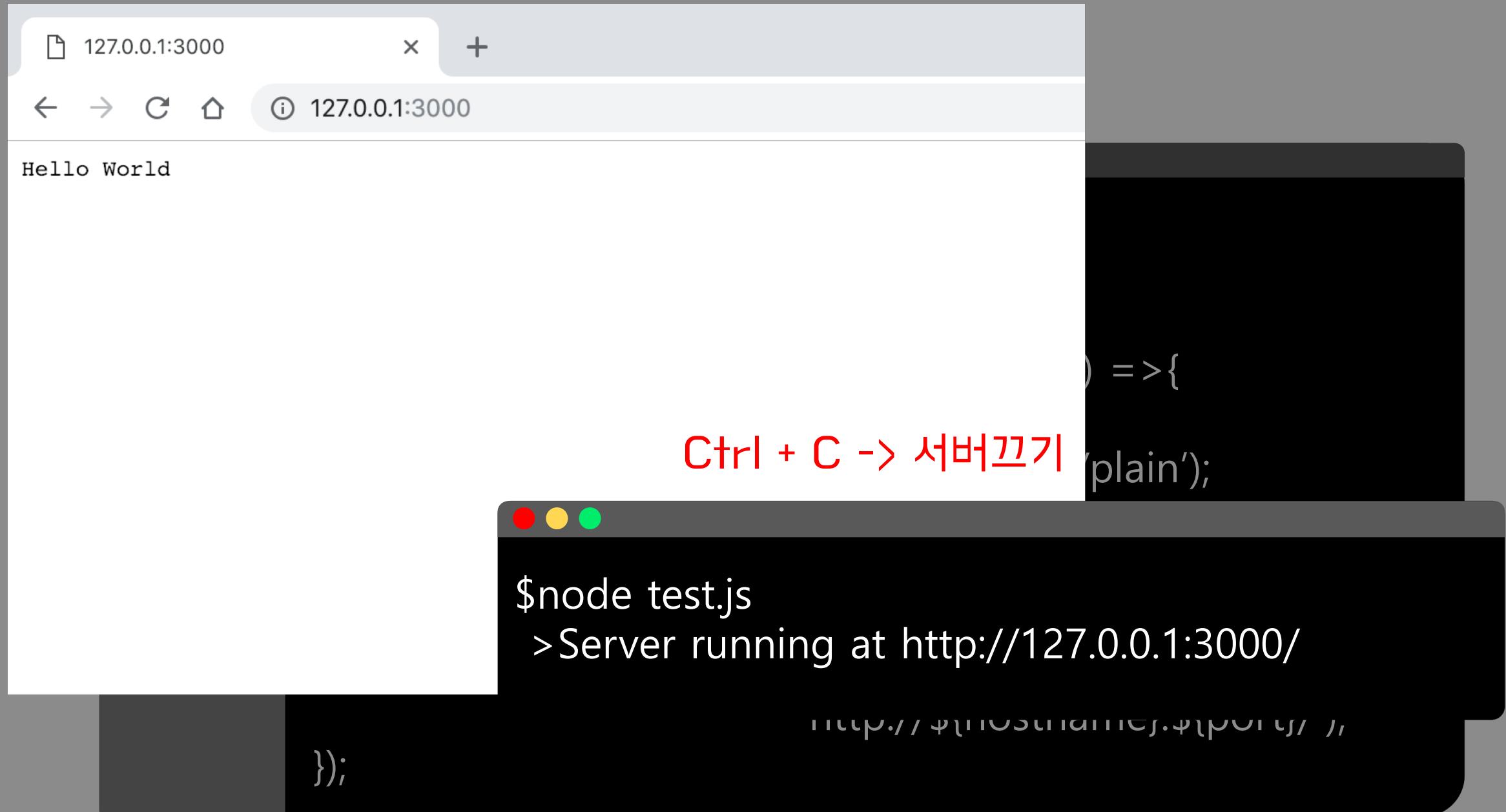
```
$node test.js
>darwin
```

|LEARN

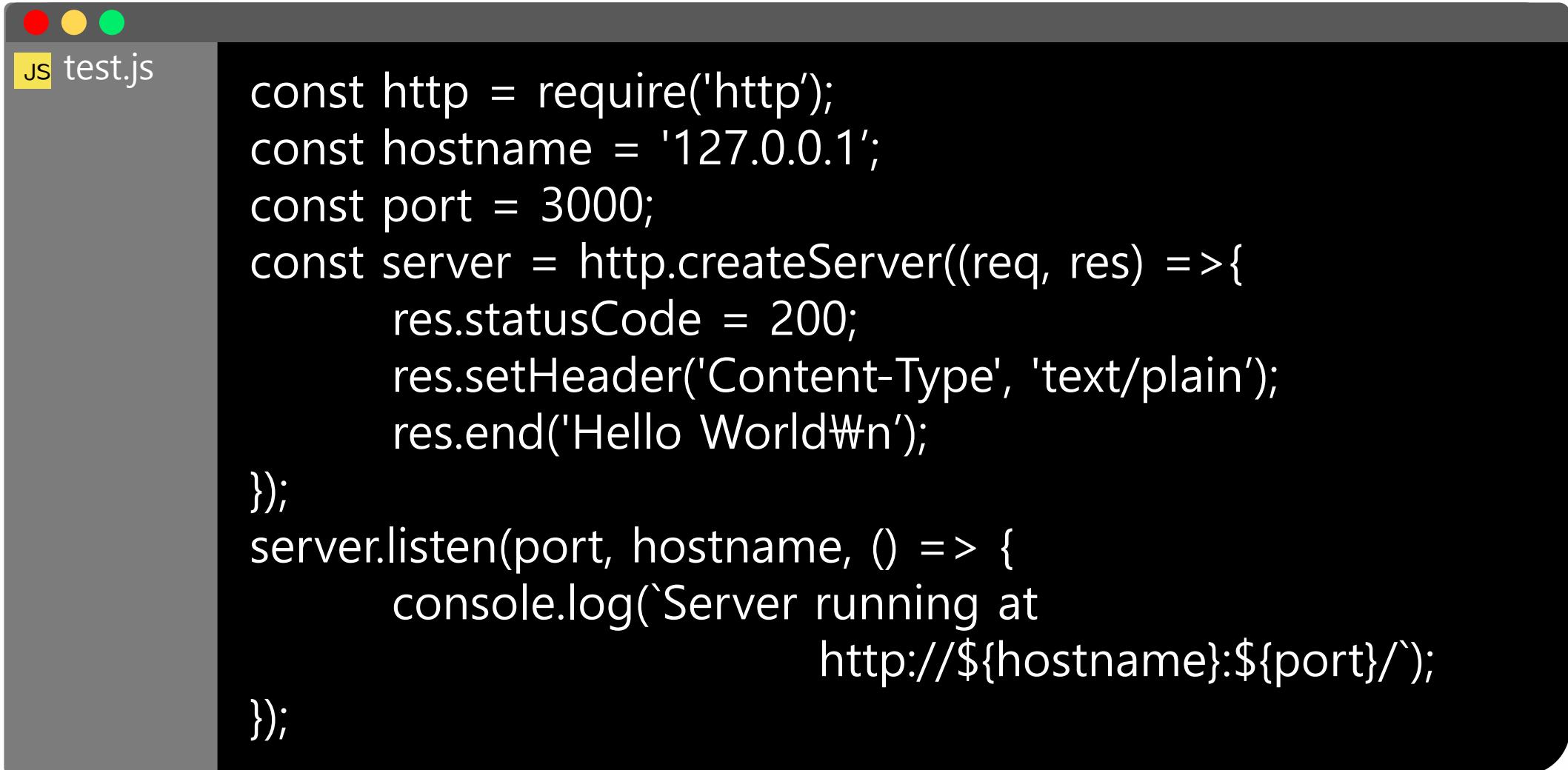


A screenshot of a code editor window titled "test.js". The code editor has a dark theme with syntax highlighting. The file contains the following JavaScript code:

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) =>{
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Hello World\n');
});
server.listen(port, hostname, () => {
    console.log(`Server running at
        http://${hostname}:${port}/`);
});
```



|LEARN



A screenshot of a code editor window titled "test.js". The code editor has a dark theme with syntax highlighting. The file contains the following JavaScript code:

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) =>{
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Hello World\n');
});
server.listen(port, hostname, () => {
    console.log(`Server running at
        http://${hostname}:${port}/`);
});
```



index.html파일을 만들고 fs내부 모듈을
사용하여 페이지 띄우기

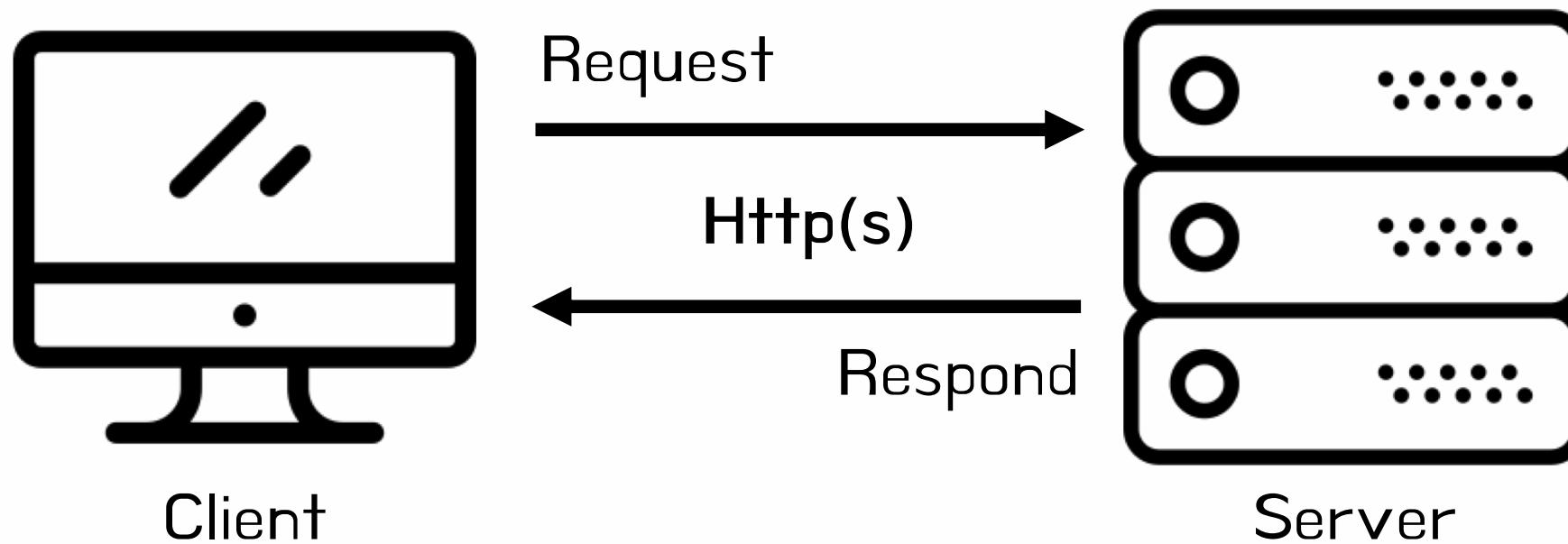
ANSWER



JS test.js

```
const http = require('http');
const fs = require('fs');
const hostname = '127.0.0.1';
const port = 3000;
fs.readFile('index.html', function (err, html) {
    if (err) { throw err; }
const server = http.createServer(function(request, response) {
    response.writeHead(200, {"Content-Type": "text/html;charset=utf-8;"});
    response.write(html);
    response.end();
});
server.listen(port, hostname, () => {
    console.log(`Server running at http://${hostname}:${port}/`);
});
});
```

|LEARN



| LEARN

Callback 함수란 ?

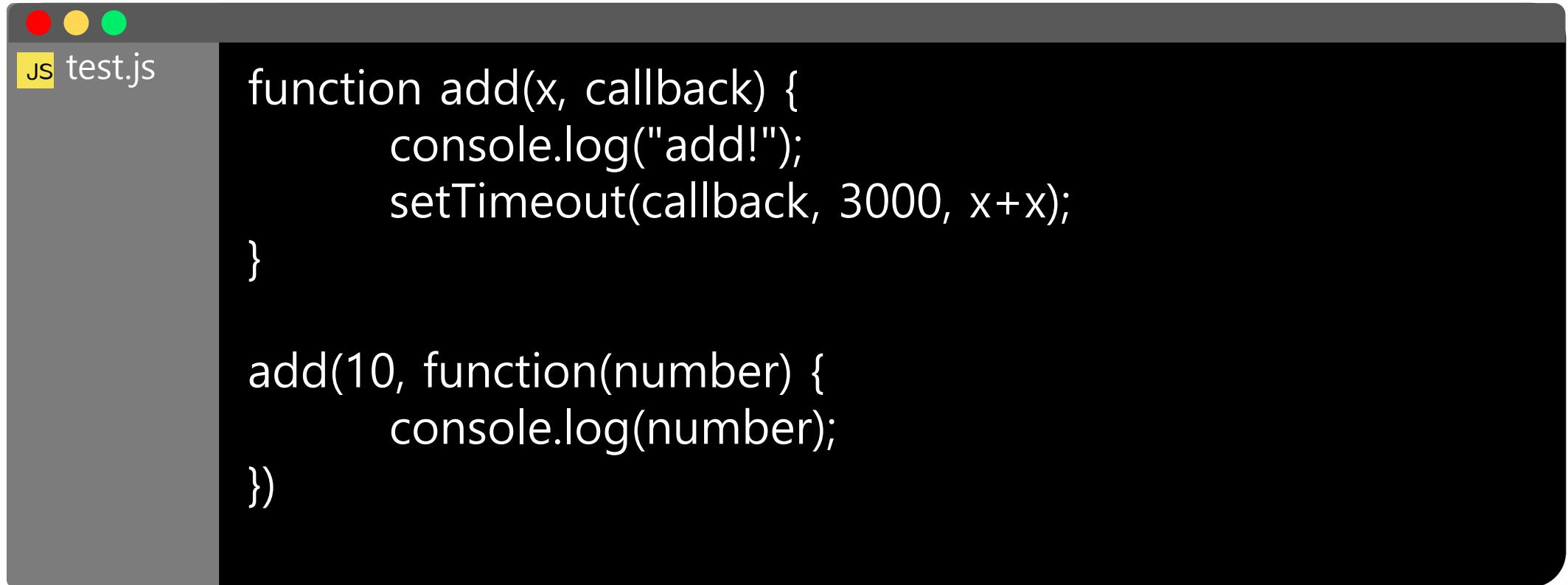
다른 함수의 인자로서 사용되는 함수

어떤 이벤트에 의해 호출되는 함수

| LEARN

비동기 프로그래밍을 위해서!!

|LEARN



A screenshot of a terminal window titled "test.js". The window has three colored window control buttons (red, yellow, green) at the top left. The code inside the terminal is as follows:

```
function add(x, callback) {
    console.log("add!");
    setTimeout(callback, 3000, x+x);
}

add(10, function(number) {
    console.log(number);
})
```

|LEARN

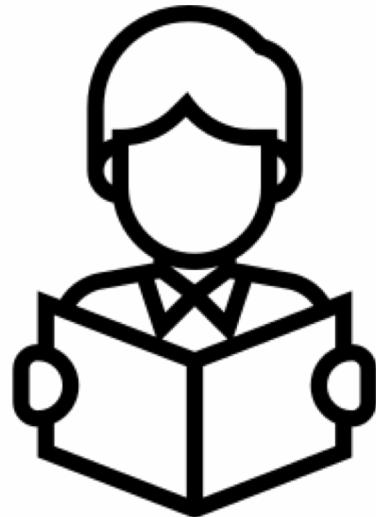


```
$node test.js  
>add!  
>20
```

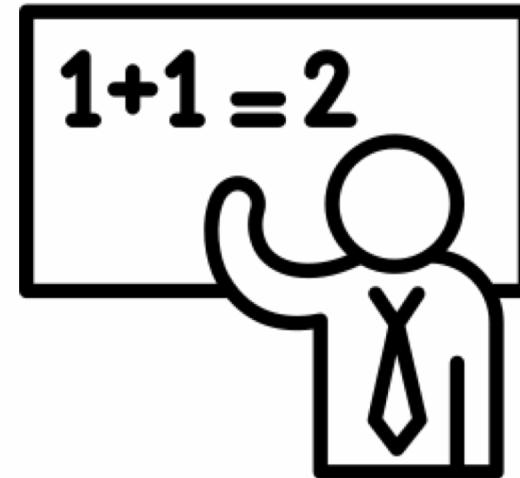
| LEARN

동기프로그래밍, 비동기 프로그래밍

|LEARN



학생



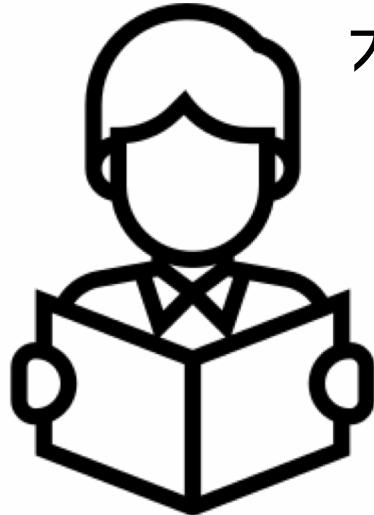
선생님

|LEARN

공부한다

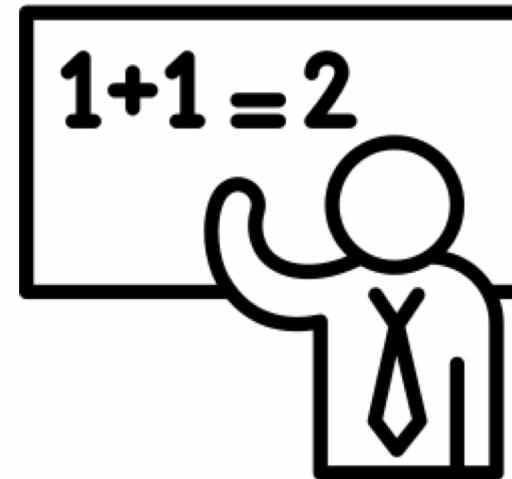
VS

기다린다



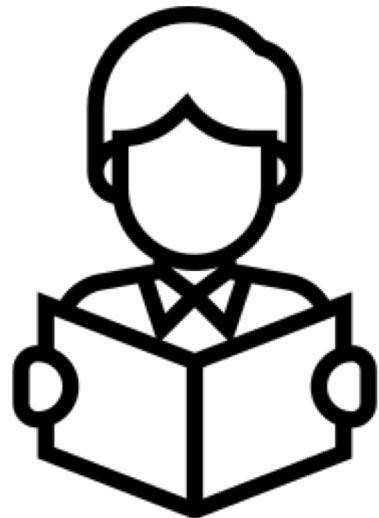
학생

채점중...

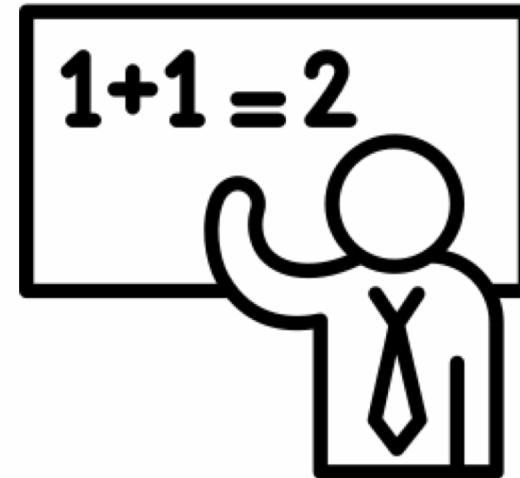


선생님

|LEARN

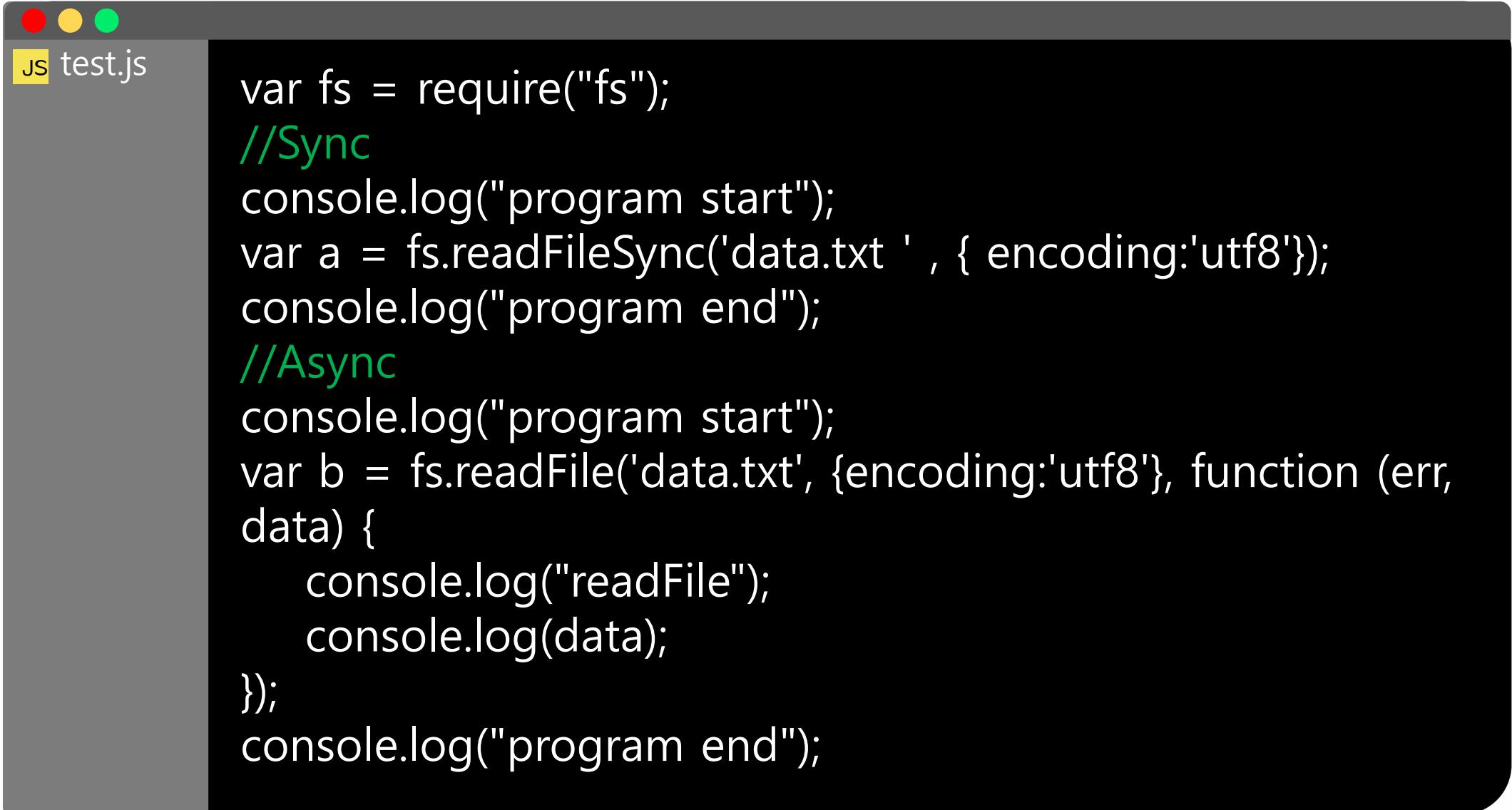


학생



선생님

|LEARN



The image shows a terminal window with a dark background. In the top left corner, there are three small colored circles (red, yellow, green). Below them, the file name "test.js" is displayed in a yellow box. The main area of the terminal contains the following code:

```
var fs = require("fs");
//Sync
console.log("program start");
var a = fs.readFileSync('data.txt' , { encoding:'utf8'});
console.log("program end");
//Async
console.log("program start");
var b = fs.readFile('data.txt', {encoding:'utf8'}, function (err,
data) {
    console.log("readFile");
    console.log(data);
});
console.log("program end");
```

| LEARN

NPM 이란 ?

Node Package Manager

node의 모듈을 관리해주는 패키지 매니저

다른 사람이 개발한 모듈 사용 가능

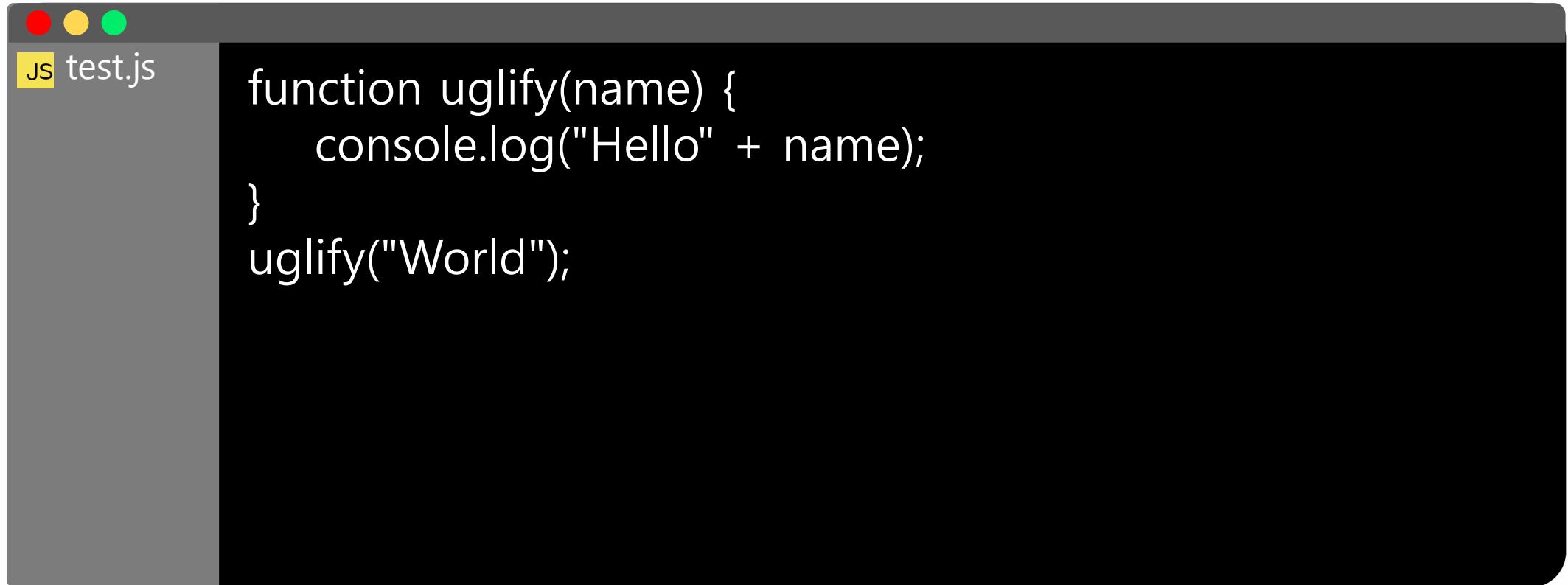
|LEARN



```
$npm install uglify-js
```

<https://www.npmjs.com/package/uglify-js>

|LEARN

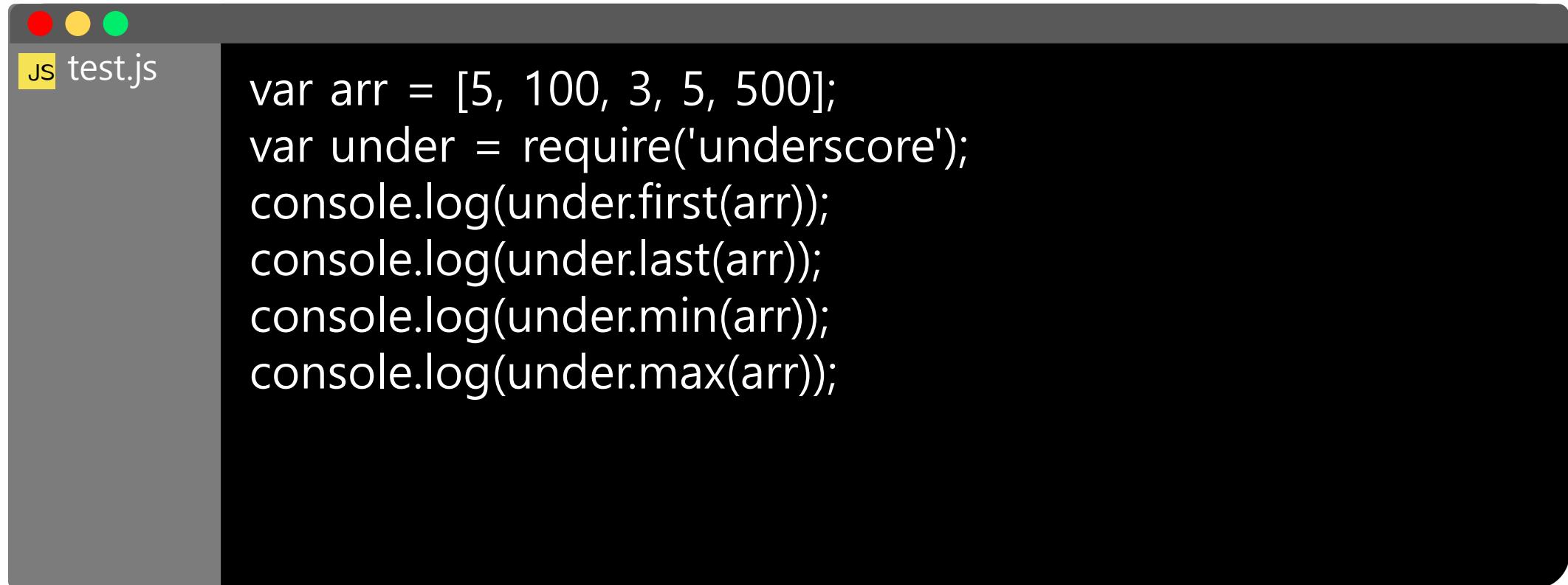


A screenshot of a terminal window with a dark theme. The window title is "test.js". The code inside the window is:

```
function uglify(name) {  
    console.log("Hello" + name);  
}  
uglify("World");
```

Underscore 모듈을 활용하여
var arr = [5, 100, 3, 5, 500] 을 선언
첫번째, 마지막, 최대, 최소 값을 출력하시오

ANSWER



A screenshot of a terminal window with a dark theme. The window title bar shows three colored dots (red, yellow, green) and the file name "test.js". The file content is displayed in white text on a black background:

```
var arr = [5, 100, 3, 5, 500];
var under = require('underscore');
console.log(under.first(arr));
console.log(under.last(arr));
console.log(under.min(arr));
console.log(under.max(arr));
```

THANK YOU