

Usage and example

Usage

```
node [options] [V8 options] [script.js | -e "script" | - ] [arguments]
```

Please see the [Command-line options](#) document for more information.

Example

An example of a [web server](#) written with Node.js which responds with `'Hello, World!'` :

Commands in this document start with `$` or `>` to replicate how they would appear in a user's terminal. Do not include the `$` and `>` characters. They are there to show the start of each command.

Lines that don't start with `$` or `>` character show the output of the previous command.

First, make sure to have downloaded and installed Node.js. See [Installing Node.js via package manager](#) for further install information.

Now, create an empty project folder called `projects` , then navigate into it.

Linux and Mac:

```
$ mkdir ~/projects
$ cd ~/projects
```

Windows CMD:

```
> mkdir %USERPROFILE%\projects
> cd %USERPROFILE%\projects
```

Windows PowerShell:

```
> mkdir $env:USERPROFILE\projects
> cd $env:USERPROFILE\projects
```

Next, create a new source file in the `projects` folder and call it `hello-world.js` .

Open `hello-world.js` in any preferred text editor and paste in the following content:

```
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}/`);  
});
```

Save the file, go back to the terminal window, and enter the following command:

```
$ node hello-world.js
```

Output like this should appear in the terminal:

```
Server running at http://127.0.0.1:3000/
```

Now, open any preferred web browser and visit `http://127.0.0.1:3000` .

If the browser displays the string `Hello, World!` , that indicates the server is working.