

Node.js

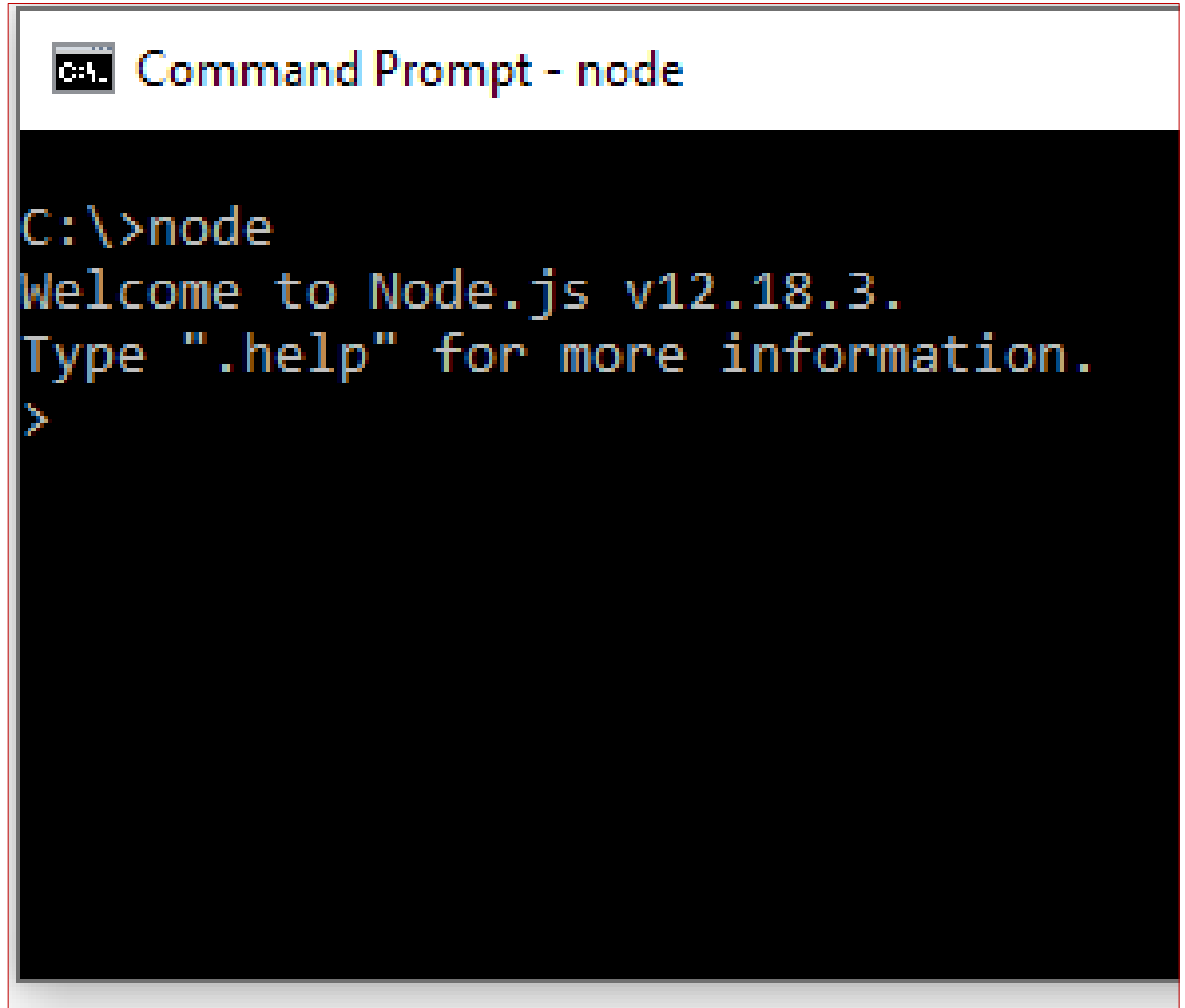
What's Node.js?

- ▢ A JavaScript runtime environment based on Google Chrome's V8 JavaScript engine
- ▢ Allowed JavaScript to be used out of the browser.
- ▢ Download and install here:
 - ▢ <https://nodejs.org/en/>
 - ▢ Download the LTS option as the most recent one may not be as stable yet.
- ▢ Helpful documentation can be found here:
 - ▢ <https://nodejs.org/en/docs/>

Node.js

Is a stand-alone ,
command line tool

Open your
command
prompt/terminal
and type node, you
can start typing
node commands

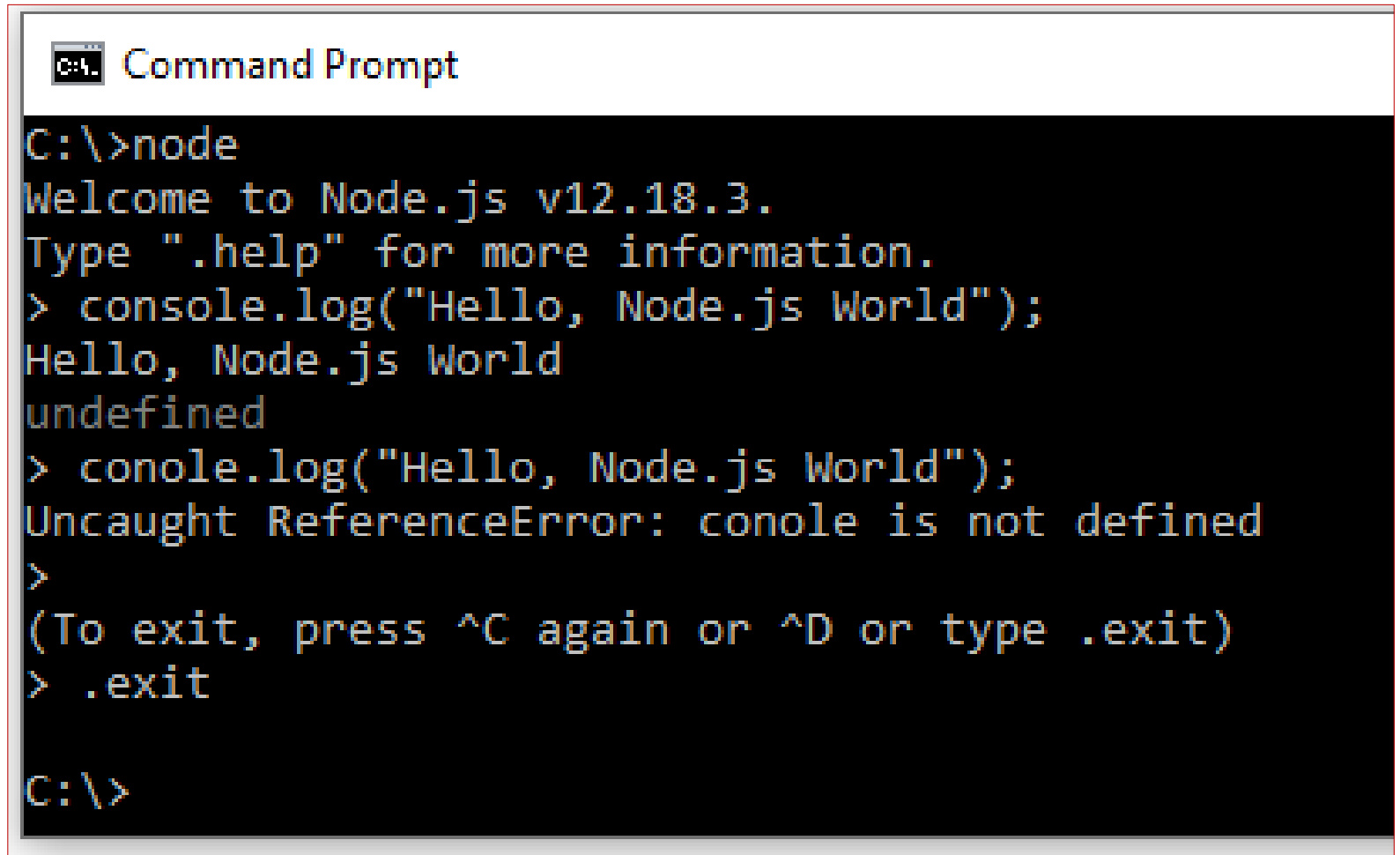


```
C:\>node
Welcome to Node.js v12.18.3.
Type ".help" for more information.
>
```

Node.js

Read-Eval-Print-Loop(REPL)

feature allows you to type JavaScript statements directly in the command line



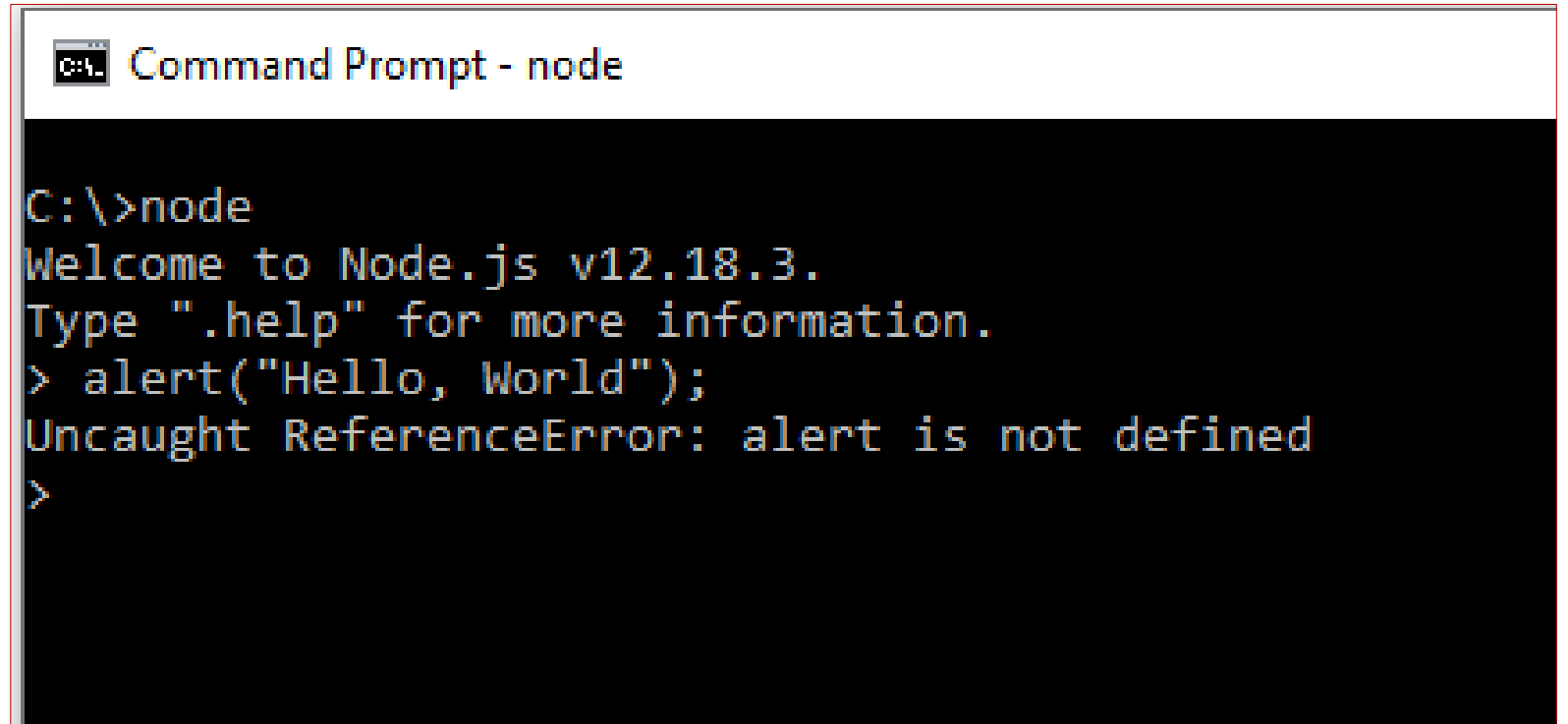
```
C:\>node
Welcome to Node.js v12.18.3.
Type ".help" for more information.
> console.log("Hello, Node.js World");
Hello, Node.js World
undefined
> conole.log("Hello, Node.js World");
Uncaught ReferenceError: conole is not defined
>
(To exit, press ^C again or ^D or type .exit)
> .exit

C:\>
```

Node.js

Read-Eval-Print-Loop(REPL)

feature allows you to type JavaScript statements directly in the command line



```
C:\>node
Welcome to Node.js v12.18.3.
Type ".help" for more information.
> alert("Hello, World");
Uncaught ReferenceError: alert is not defined
>
```

Not all built-in code available in the browser will be available in Node.js and vice versa

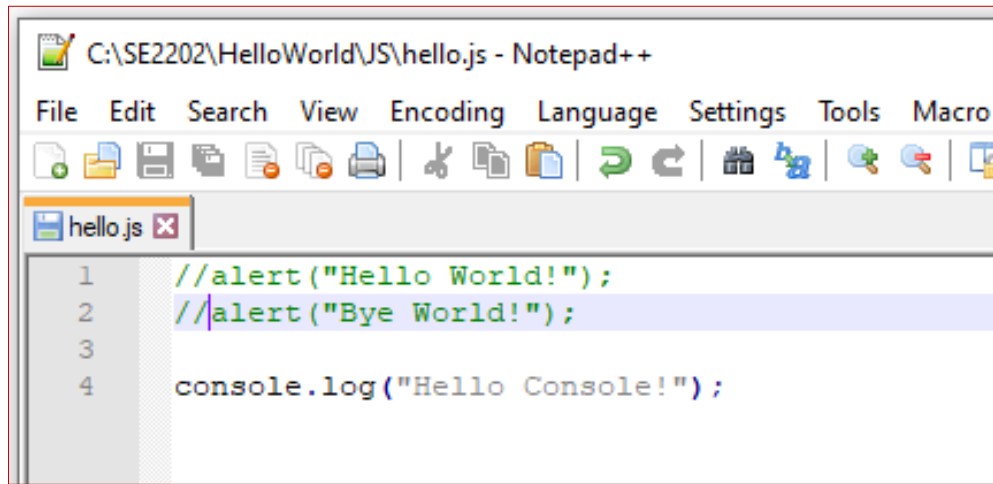
Node.js (REPL)

In addition to executing JS statements REPL has its own commands

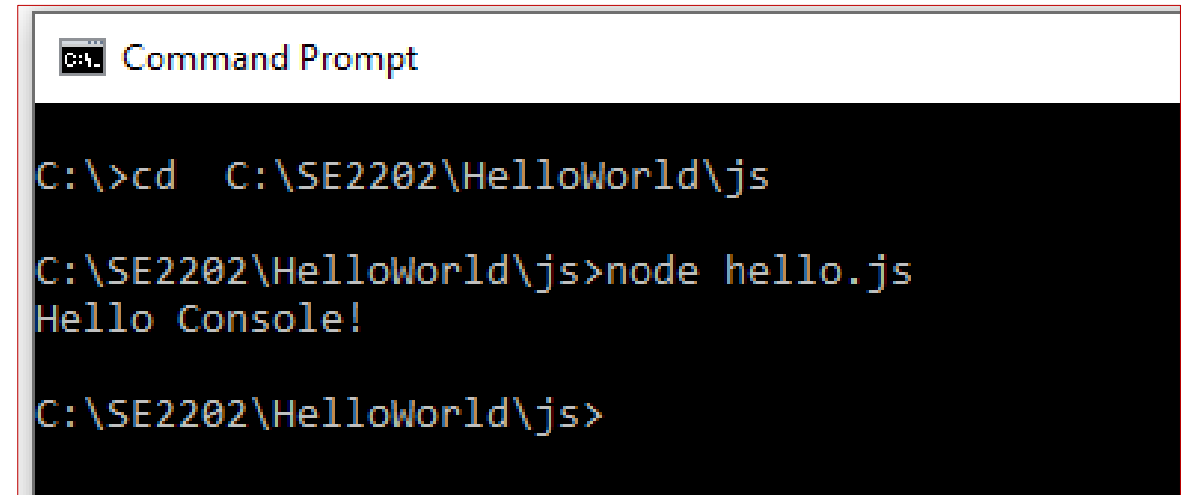
One of them is “.help” that lists all the possible commands

Command Prompt - node

```
Welcome to Node.js v12.18.3.  
Type ".help" for more information.  
> .help  
.break      Sometimes you get stuck, this gets you out  
.clear      Alias for .break  
.editor      Enter editor mode  
.exit        Exit the repl  
.help        Print this help message  
.load        Load JS from a file into the REPL session  
.save        Save all evaluated commands in this REPL session to a file  
  
Press ^C to abort current expression, ^D to exit the repl  
>
```



```
C:\SE2202\HelloWorld\JS\hello.js - Notepad++  
File Edit Search View Encoding Language Settings Tools Macro  
hello.js  
1 //alert("Hello World!");  
2 //alert("Bye World!");  
3  
4 console.log("Hello Console!");
```



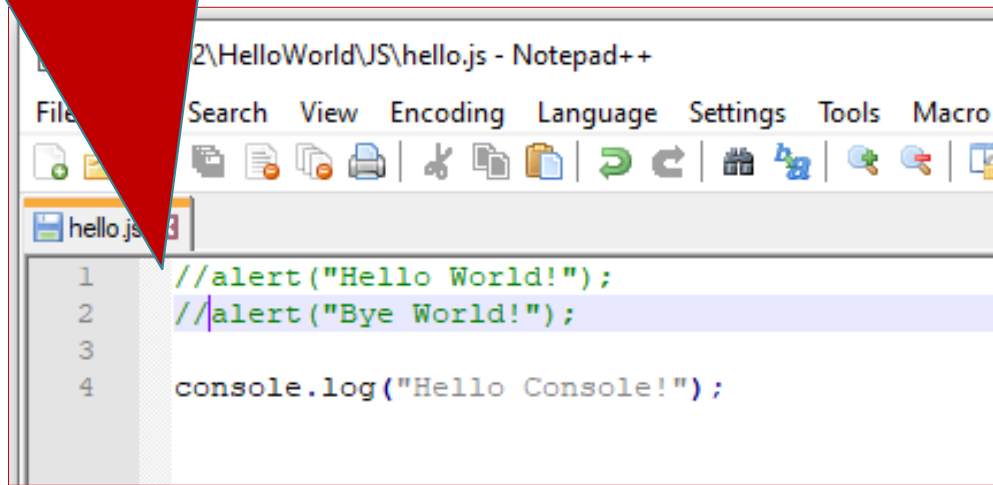
```
Command Prompt  
C:\>cd C:\SE2202\HelloWorld\js  
C:\SE2202\HelloWorld\js>node hello.js  
Hello Console!  
C:\SE2202\HelloWorld\js>
```

Node.js

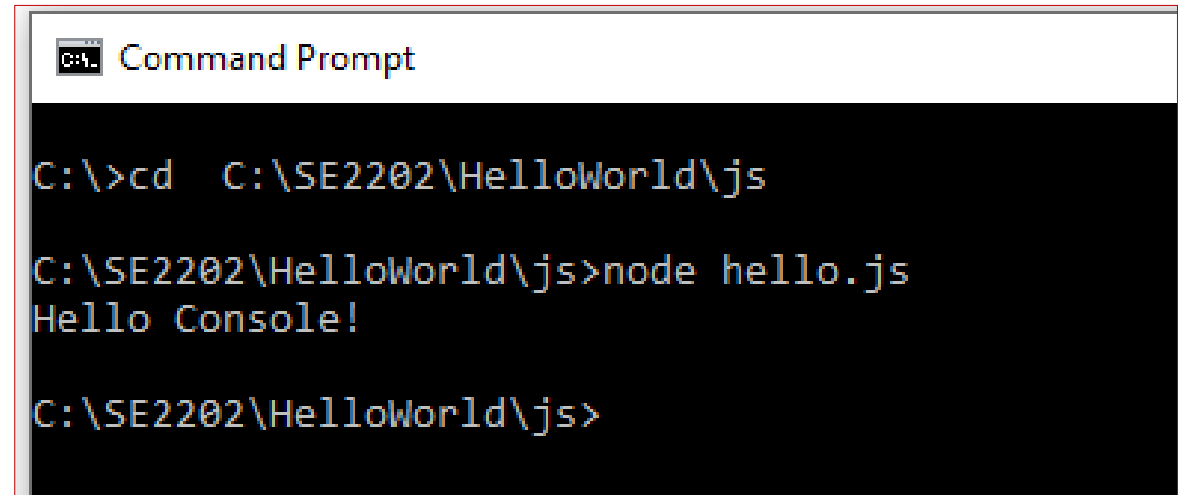
You can also execute a js file directly by giving the file name as a command line argument

JS Comments

- `//` one-line comment
- `/* ... */` multi-lines comment



```
1 //alert("Hello World!");
2 //alert("Bye World!");
3
4 console.log("Hello Console!");
```



```
C:\>cd C:\SE2202\HelloWorld\js

C:\SE2202\HelloWorld\js>node hello.js
Hello Console!

C:\SE2202\HelloWorld\js>
```

Node.js

You can also execute a js file directly by giving the file name as a command line argument

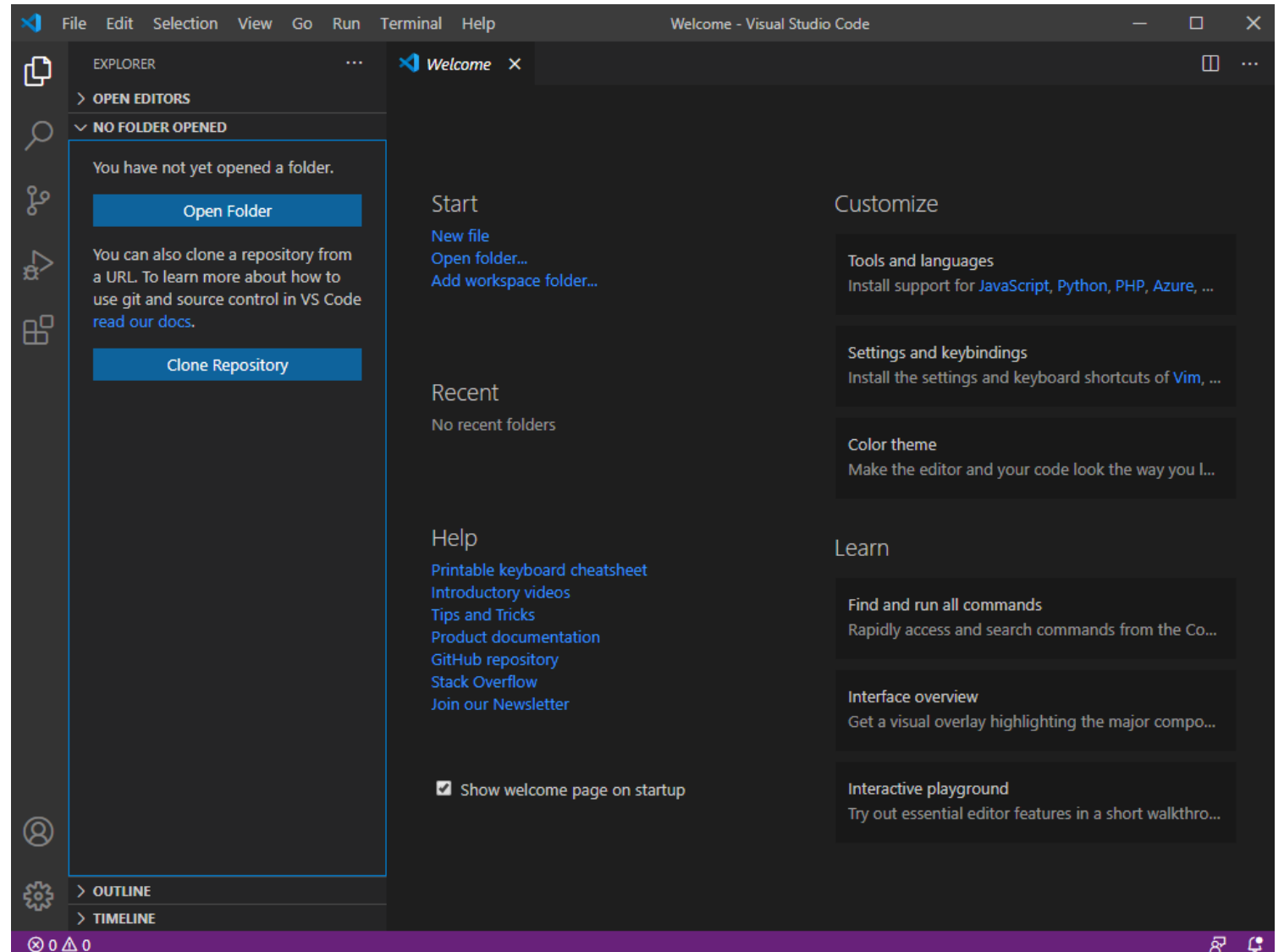


Visual Studio Code

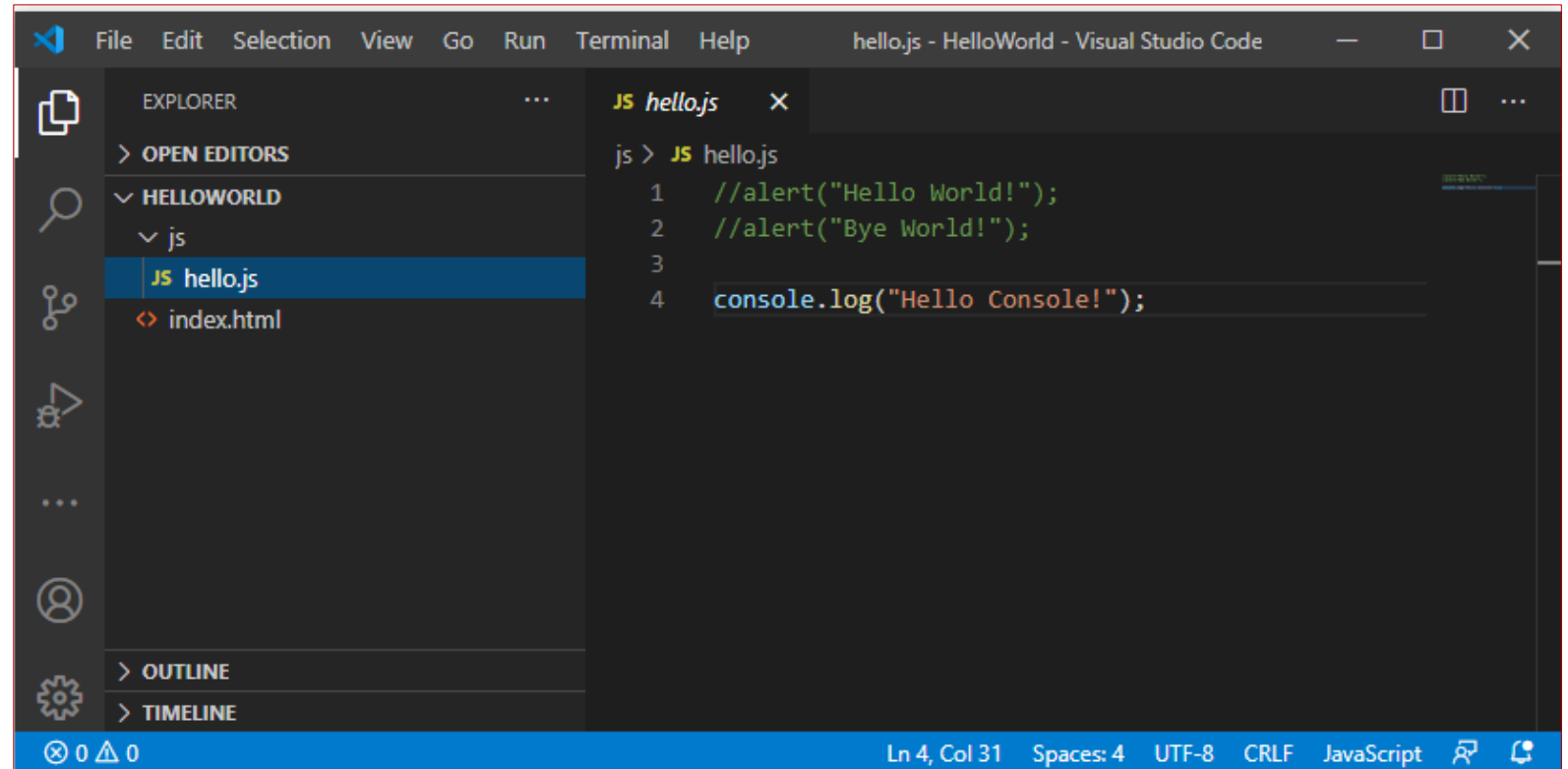
VSCode

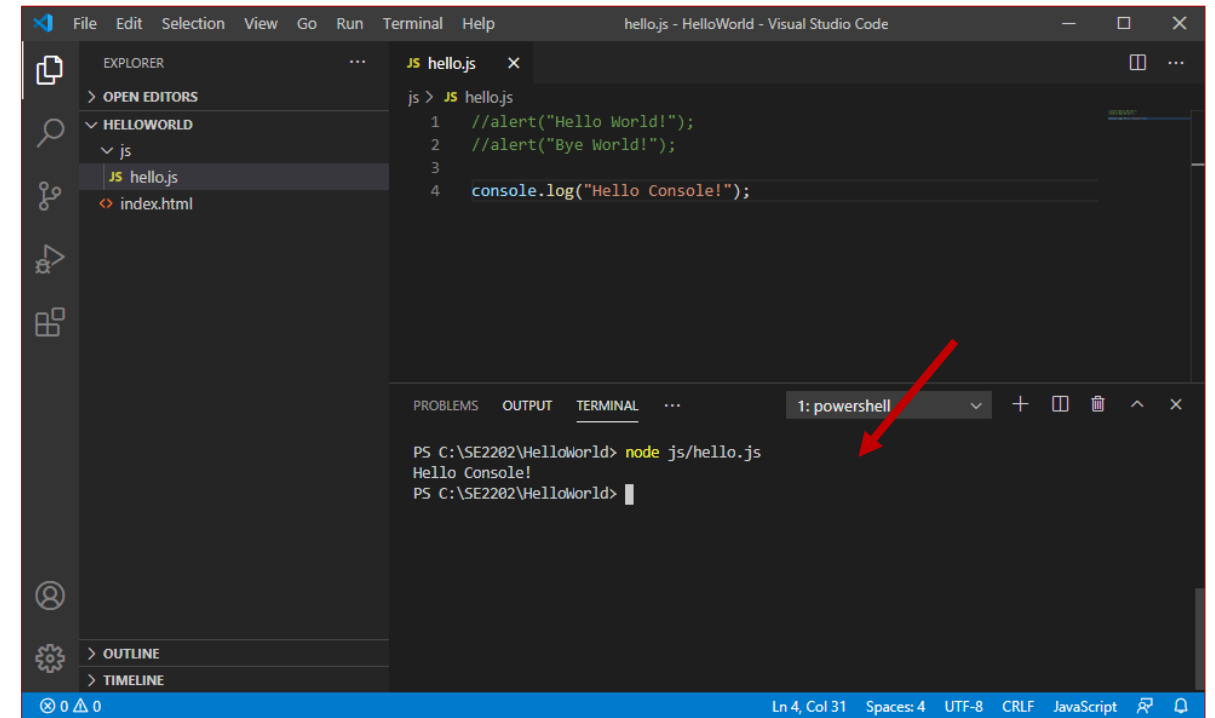
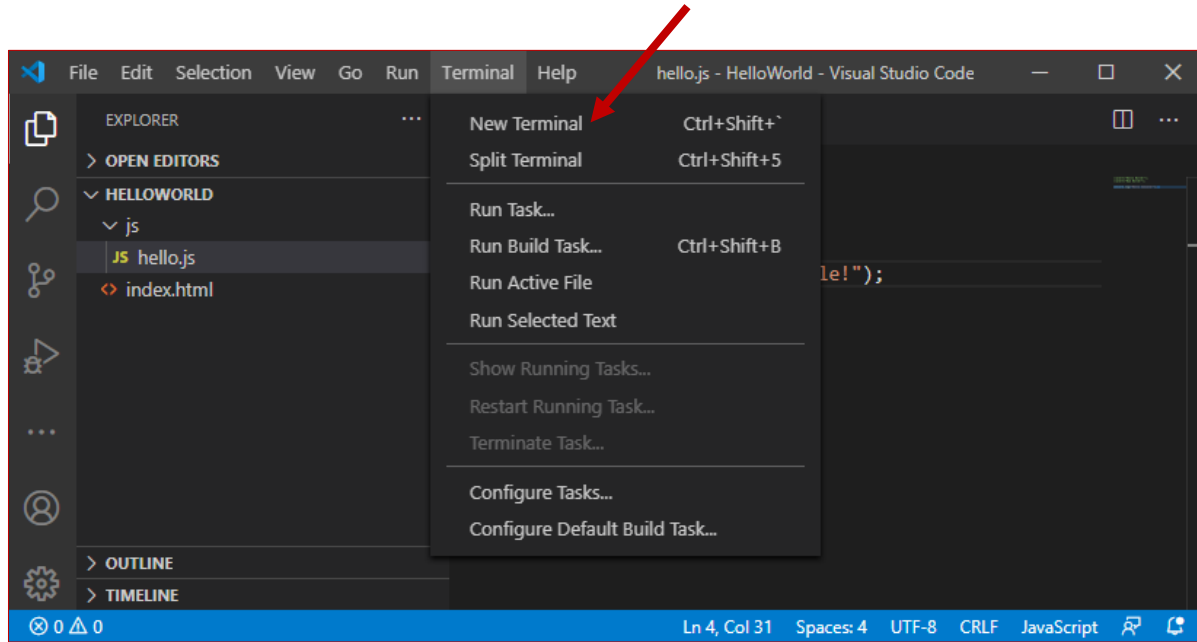
- ▮ Advanced “Code-Editor” or light IDE
- ▮ Provides tools to facilitate coding
- ▮ Download and install from here:
<https://code.visualstudio.com/>

VSCode



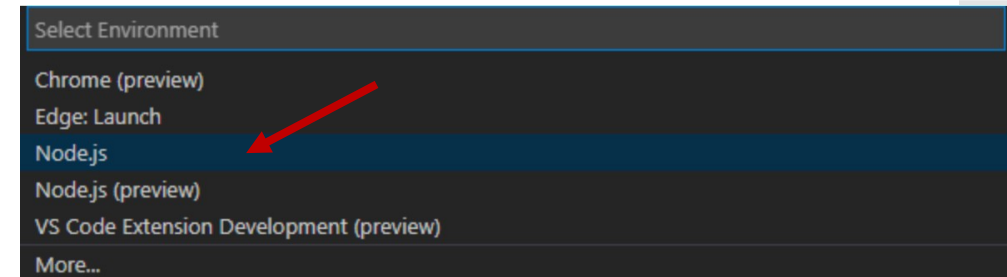
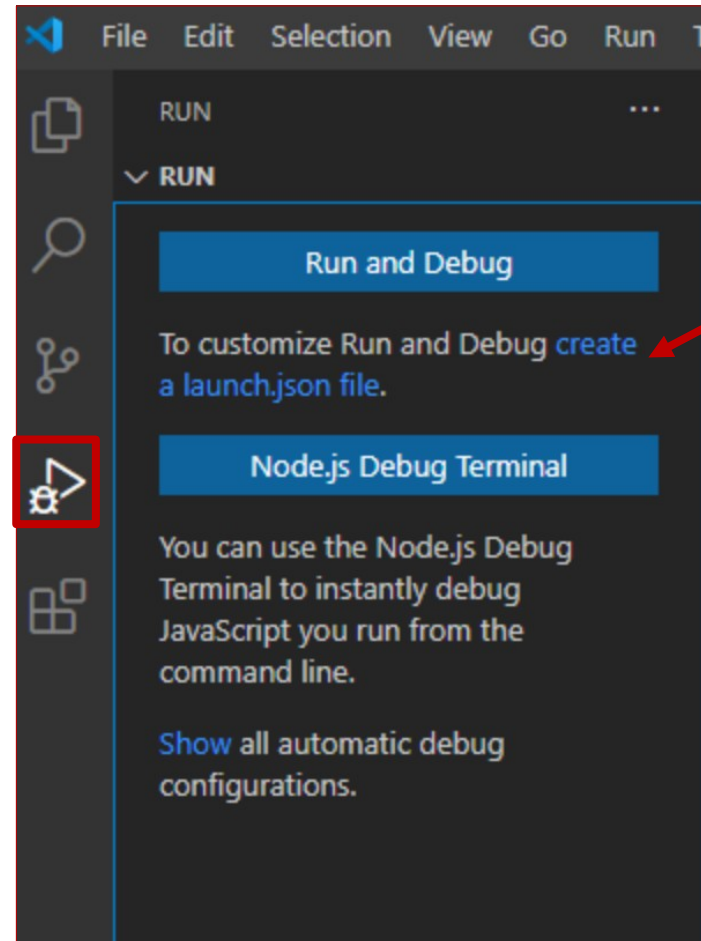
VSCode



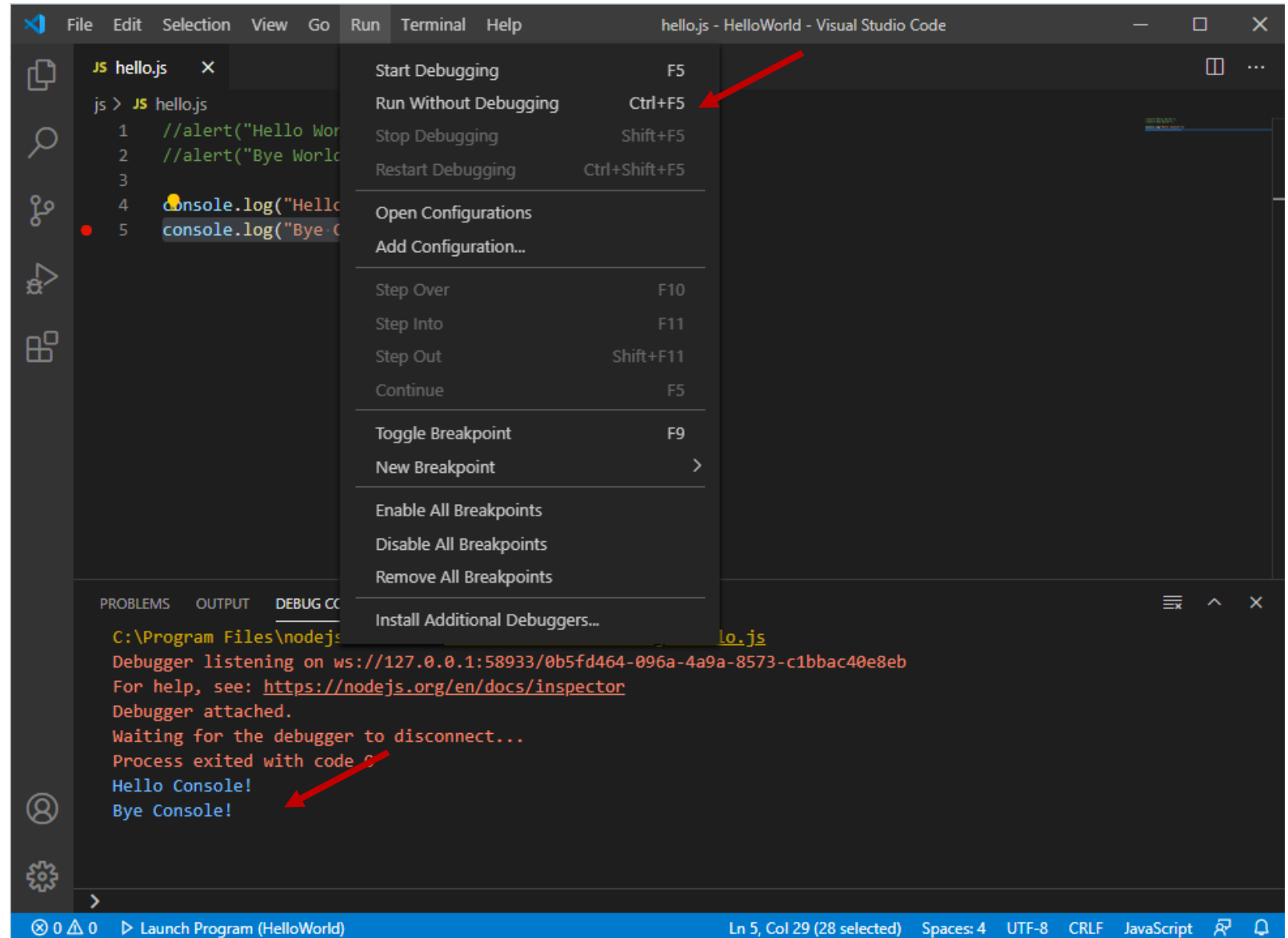


Running Node.js within VSCode

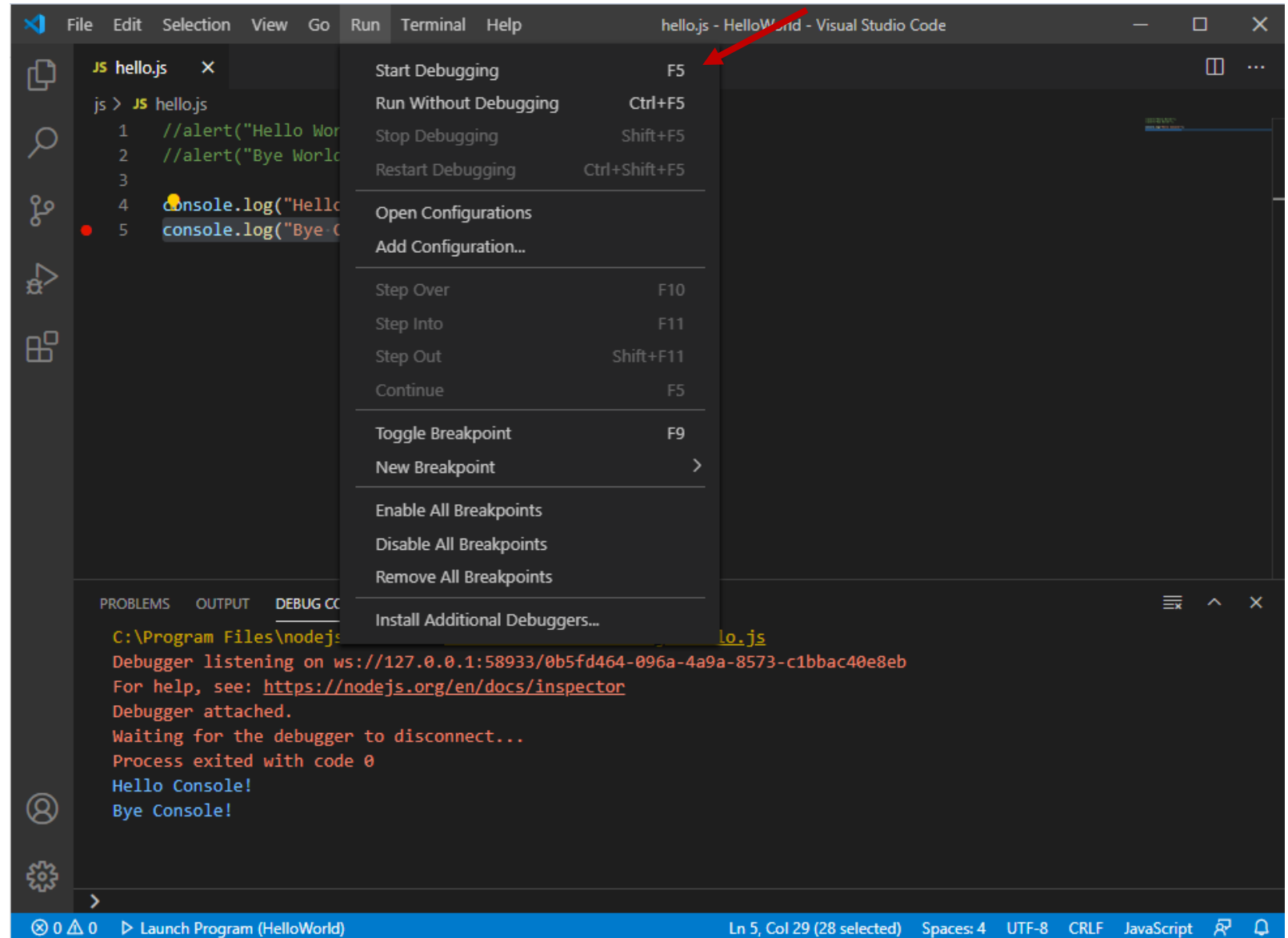
Running Node.js within VSCode



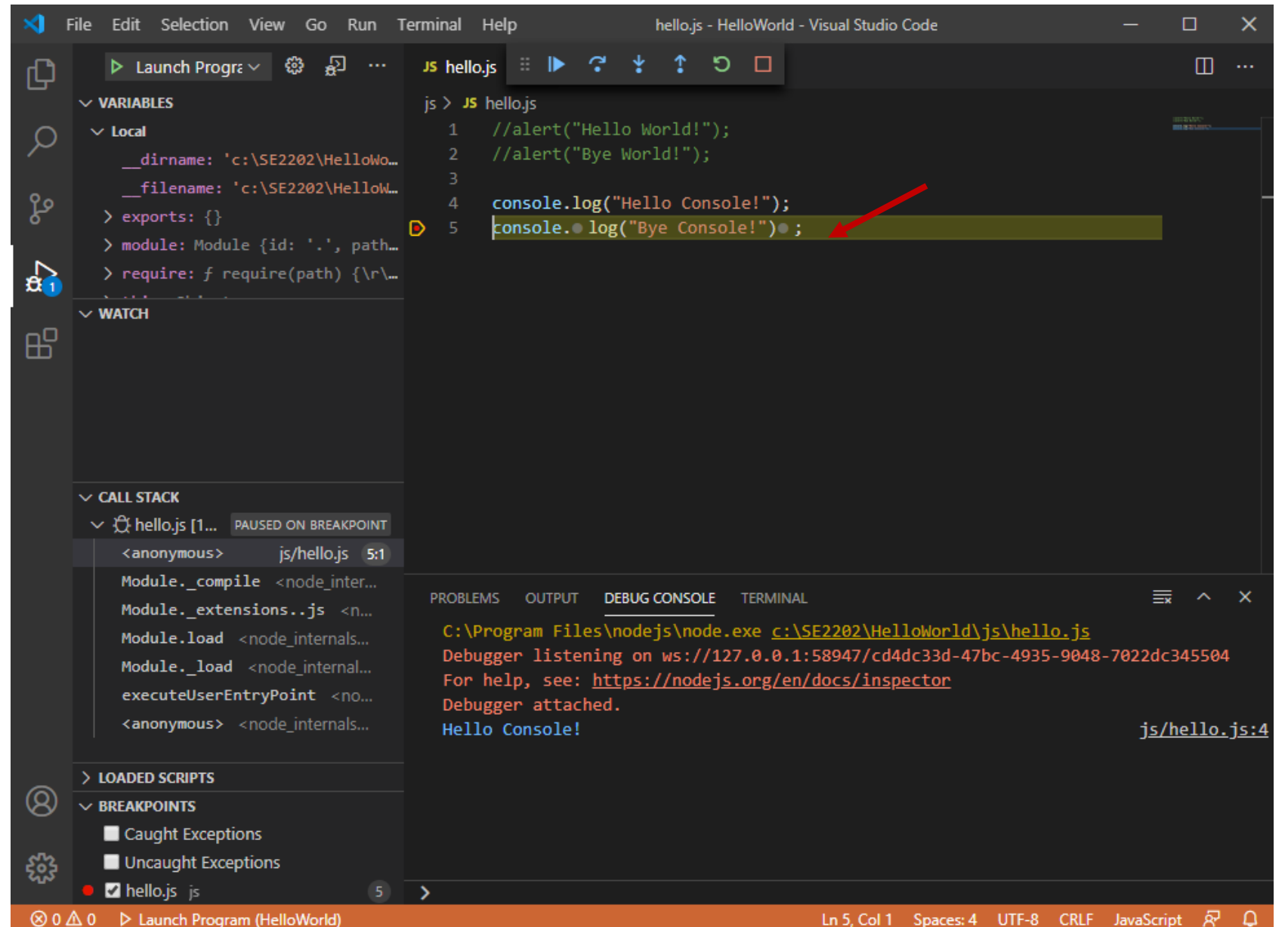
Running Node.js within VSCode



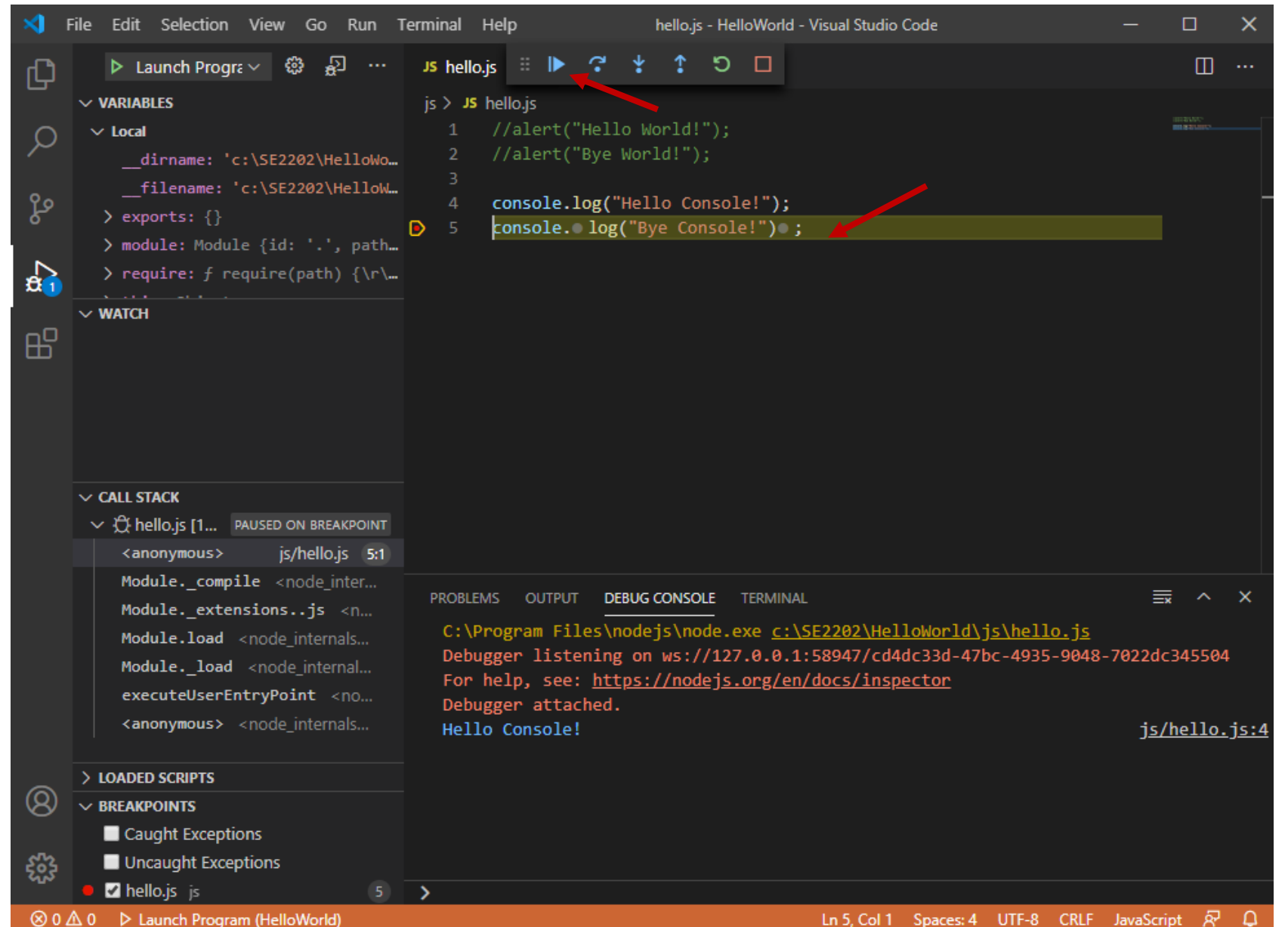
Running Node.js within VSCode



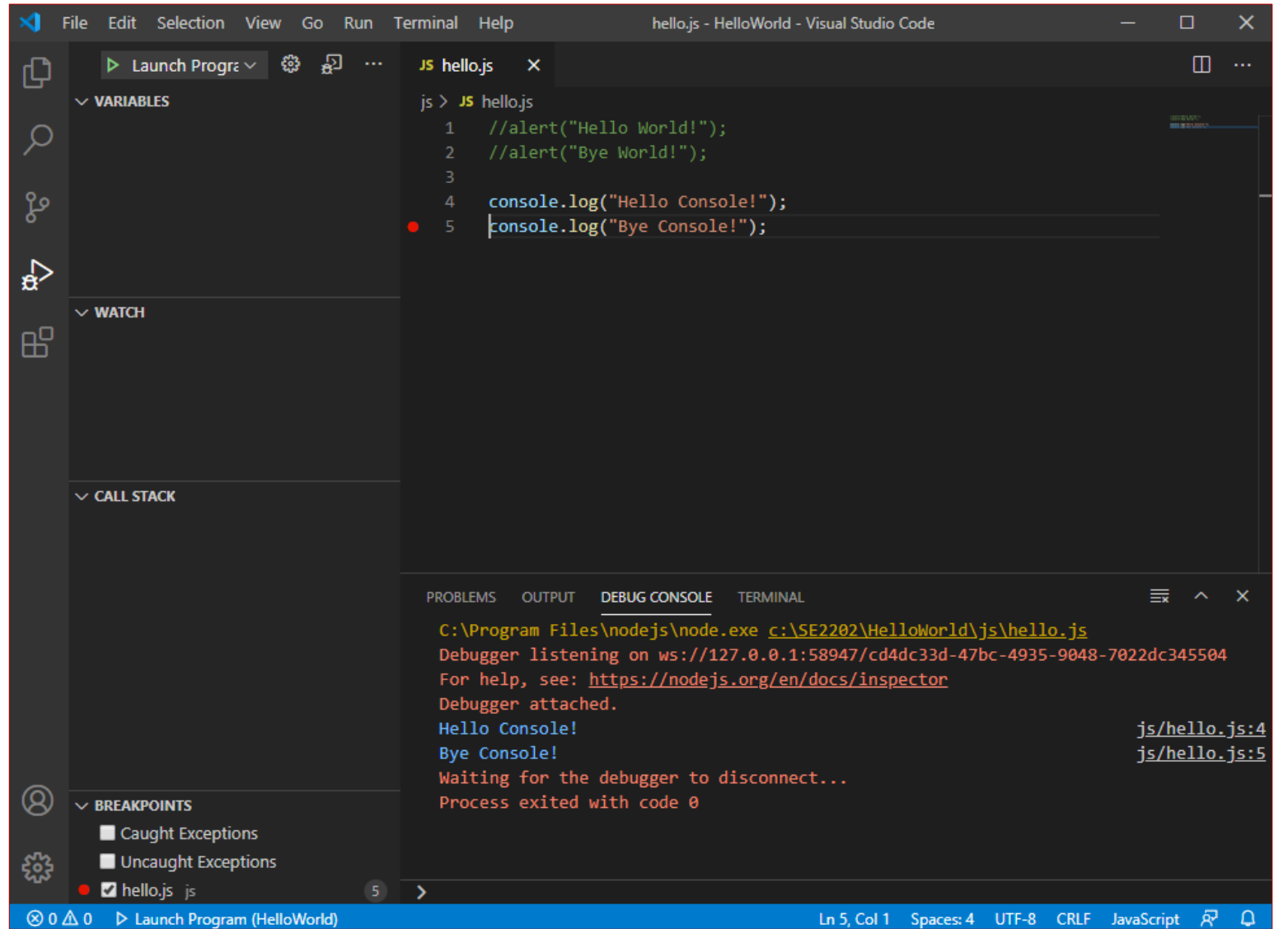
Running Node.js within VSCode



Running Node.js within VSCode



Running Node.js within VSCode



The screenshot shows the Visual Studio Code interface with a JavaScript file named `hello.js` open. The code contains two `console.log` statements. The interface is configured for debugging, with the `DEBUG CONSOLE` tab selected at the bottom. The `DEBUG CONSOLE` shows the execution of `node.js` and the output of the `console.log` statements. The `CALL STACK` tab is also visible, showing the current execution context.

```
js > JS hello.js
1 //alert("Hello World!");
2 //alert("Bye World!");
3
4 console.log("Hello Console!");
5 console.log("Bye Console!");
```

DEBUG CONSOLE

```
C:\Program Files\nodejs\node.exe c:\SE2202\HelloWorld\js\hello.js
Debugger listening on ws://127.0.0.1:58947/cd4dc33d-47bc-4935-9048-7022dc345504
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Hello Console!
Bye Console!
Waiting for the debugger to disconnect...
Process exited with code 0
```

CALL STACK

```
js/hello.js:4
js/hello.js:5
```

Launch Program (HelloWorld) Ln 5, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

Just-In-Time (JIT) Compiler

Just-In-Time (JIT) Compilation

- ▮ Also known as **dynamic translation** or **run-time compilations** (Wikipedia)
- ▮ Compilation of code during run time
- ▮ Instead of translating then executing one line at a time, it translates the whole script and cache it for execution.

JIT Compiler vs. Simple Interpreter

Source Code

```
cons getName = (person) => person.lastname;  
  
cons han = {firstname: "Han", lastname: "Solo"};  
cons luke = {firstname: "Luke", lastname: "Skywalker"};  
cons leia = {firstname: "Leia", lastname: "Organa"};  
cons obi = {firstname: "Obi-Wan", lastname: "Kenobi"};  
  
cons persons = [han, luke, leia, obi];  
  
for(var i = 0; i < 1000 * 1000 * 1000; i++) {  
  getName(persons[i & 3]);  
}
```

JavaScript Engine

Interpreter

Compiler (JIT)

© Rainer Hahnekamp

Source:

<https://www.freecodecamp.org/news/javascript-essentials-why-you-should-know-how-the-engine-works-c2cc0d321553/>