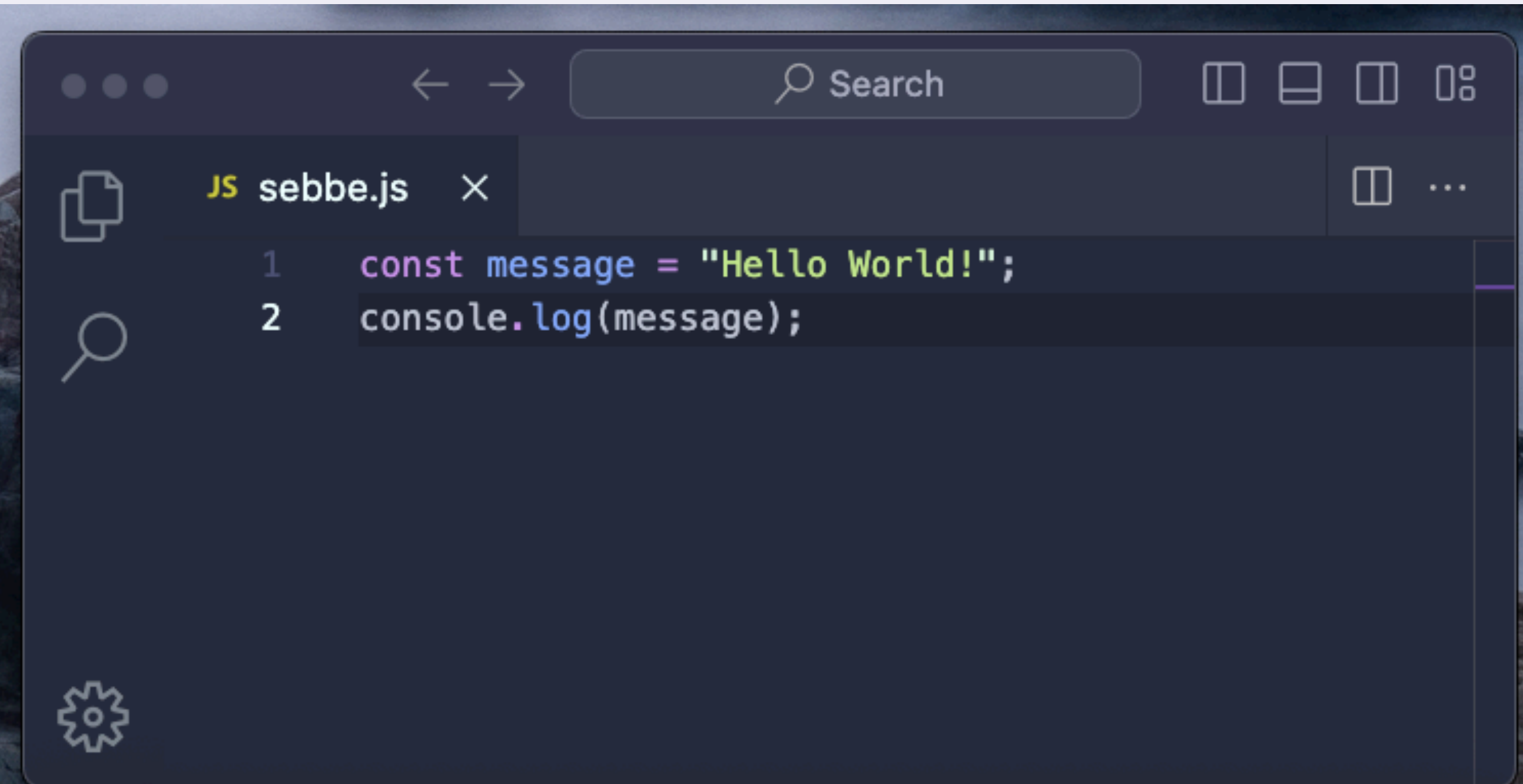




- Installera Deno
 - <https://docs.deno.com/runtime/#install-deno>
 - På sidan finner ni ett kommando ni behöver köra i er terminal
 - **OBS!** På Windows behöver ni använda "PowerShell"
- Skapa en JavaScript fil och kör den (i terminalen!)
 - `$ deno sebbe.js`



A screenshot of a code editor window. The title bar shows three dots, navigation arrows, a search bar with the text "Search", and icons for window management. The editor has a tab labeled "JS sebbe.js" with a close button. The code is as follows:

```
1  const message = "Hello World!";  
2  console.log(message);
```

On the left side of the editor, there are icons for a file, a search, and a settings gear.



A screenshot of a terminal window titled "Terminal". The prompt is "sebbe:MCC44YN792N6 ~/Desktop". The command "deno sebbe.js" has been entered and executed, resulting in the output "Hello World!". The prompt is now "sebbe:MCC44YN792N6 ~/Desktop" followed by a cursor.

```
sebbe:MCC44YN792N6 ~/Desktop λ deno sebbe.js  
Hello World!  
sebbe:MCC44YN792N6 ~/Desktop λ
```

<https://docs.deno.com/examples/>

Docs

ManualAPI referenceExamplesDeploySubhosting

deno.com

Search

Examples

A collection of walkthrough tutorials, examples, videos and guides to teach you about the Deno runtime and how to use it with your favorite tools.

Filter by type:

</> Examples: ☒

☒ Tutorials: ☒

☒ Videos: ☒

Basics

</> What is Deno?

☒ Run a script

</> Hello World

</> Built in TypeScript support

</> Your Deno Dev Environment

☒ Initialize a project

☒ Executable scripts

</> All-in-one tooling

</> Tasks and configuration with deno.json

☒ Update from CommonJS to ESM

</> Import and export functions

</> Interoperability with Node.js

</> Introduction to Deno APIs

☒ Simple file server

Deploying Deno projects

☒ AWS Lambda

</> Deploy Deno to AWS Lambda

☒ AWS Lightsail

☒ Cloudflare workers

☒ Digital Ocean

☒ Google Cloud Run

☒ Kinsta

</> Deploying Deno with Docker

Connecting to Databases

☒ Connecting to databases

☒ Use MySQL2 with Deno

☒ Use PlanetScale with Deno

☒ Use Redis with Deno

System

</> Handling OS signals

</> Benchmarking

☒ Create a subprocess

</> Subprocess Spawning

</> Collecting output from subprocesses

</> Reading system metrics

</> Process information

</> Environment variables

</> Subprocesses: Spawning

☒ Handle OS signals

FileSystem

</> Path operations

</> Reading files