



Intro to JavaScript Fundamentals

Learning objective: By the end of this lesson, students will be able to utilize comments for code documentation and print to the console for debugging.

Making Comments

In JavaScript and many other programming languages, we can make our more readable and maintainable for ourselves and others by writing **comments**. Comments are pieces of text that are ignored when code is evaluated.

There are several reasons why comments in code are valuable to us:

- Help ourselves and others understand our code when reading it later.
- Help us document why a decision was made - what the purpose of code is or provide other context.
- They help document our code's functionality, aiding in troubleshooting and debugging.
- Documenting any known issues with your code. Perhaps it doesn't account for specific edge cases or isn't as performant as it could be, and you want to return to it later.

As you're just learning, don't be afraid to add comments to help you understand what that code is doing. However, outside of a learning environment, comments explaining why decisions were made are typically more valuable than comments explaining what code does.

There are two types of comments in JavaScript: **single-line** and **multi-line**.

Single-line comments

Single-line comments start with two forward slashes `//` and continue to the end of the line. Everything on that line after the forward slashes is ignored when the code is evaluated.

```
// This is a single-line comment.
```

Copy

You can write single-line comments on their own line or on the same line as code (but this can get messy). The forward slashes will ignore the code written before them.

```
console.log('Comments are great'); // I only ignore words written after me!
```

Copy



VS Code can automatically comment line(s) of code for us by pressing `⌘ Command` + `/` in macOS or `Ctrl` + `/` in Linux and Windows.

Multi-line comments

Multi-line comments in JavaScript are opened with a forward slash and an asterisk `/*` and closed with an asterisk and a forward slash `*/`. When the code is evaluated, anything written between the start and end tags is ignored.

```
/*  
  This is a multi-line comment.  
  It can span multiple lines.  
*/
```

Copy

Printing to the console

We can print output to the terminal app or console in the browser running our code!

To print to the console in JavaScript, we can use a `console.log()`. Whatever is in the parenthesis will be printed.

```
console.log('Hello, world!');  
// Prints: Hello, world!
```

Copy

Printing to the console, or console logging, is one of the most powerful tools we have for understanding and debugging our code. Trying things out and console logging often is one of the best ways to learn JavaScript.