

The JavaScript Console

When we create front-end websites and applications, we work in the browser environment. We can use JavaScript to manipulate the DOM (Document Object Model) and display different elements and values, but many times, we just need a quick way to show what a value is. We also need a place to see any errors or warnings in our script or any script that runs. This is where the JavaScript console comes in.

The console is part of the **Developer Tools** in the browser. In most browsers, you can open the dev tools with **F12** on a Windows PC and **CMD+OPT+I** on a Mac. From there you have a bunch of tabs and one (usually the second) is the console. You can also use **CMD+OPT+J** or **CTRL+ALT+J** to go right to the console.

From here you can actually run JavaScript. Try typing in the following directly into the browser console and hit **enter**.

```
alert('Hello from the console')
```

It should show a browser alert with the text.

We can write just about any single line JavaScript expression directly in the console, however, you probably won't do this very much. We usually use the console to output information and values from our script/code.

Console methods

The **global object** in the browser environment gives us access to a **console** object that has a bunch of useful methods that we can use from our JavaScript file to interact with the JavaScript console. When I say a method, I just mean a function that is attached to an object. In this case, the "console" object.

console.log()

This is the most common console method. We pass whatever we want to log to the console as an argument to the **log()** method. This can be a string, a number, a boolean, an object, an array, or even a function. It will output the value to the console.

Log a number

Different data types will have different colors in the console

```
console.log(123);
```

Log a string

```
console.log('Hello World');
```

Log multiple values

You can separate values with a comma

```
console.log(123, 'Hello', true);
```

Log a variable

For the most part, we use the console to debug and log out variables or the result of a function or network request.

```
x = 100;  
console.log(x);
```

console.error()

using console.error() will make the text red

```
console.error('This is an error');
```

console.warn()

Using console.warn() will make the text yellow

```
console.warn('This is a warning');
```

console.clear()

```
console.clear()
```

console.table()

If you want to log an object, you can format it as a table

```
console.table({ name: 'Brad', city: 'Boston' });
```

console.group()

```
console.group('simple');  
console.warn('warning!');  
console.error('error here');  
console.log('Hello World');  
console.groupEnd('simple');  
console.log('new section');
```

Log with style

```
const styles = 'padding: 10px; background-color: white; color: green;';  
console.log('%cHello', styles);
```

There are even more methods. See all of them [here](#)