HELLO WORLD IN JAVASCRIPT

As said by Kernighan and Ritchie at the beginning of The C Programming Language, the only way to learn programming in a language is to write programs in it.

They also said that the writing of ‘Hello World’ is the first program in every language. So we will start there.

There are a couple of choices to make here. The first is knowing what environment the code is written for. Will it run in **Node.js** or work on a **webpage**?

The HTML page has the advantage of being readable by any device with a browser. The Node method requires a desktop or laptop computer that has Node installed into its operating system. So it has to be running on Windows, Mac OS, or Linux.

The second choice will be, which method to output do I want to use? For there are several. In fact, there are several types.

1. methods using the console: These can include:

- console.log(), console.err(), console.warn(), console.info()
- console.table(), console.dir()
- console.group()/groupend()
- console.count()

2. browser dialogue methods:

- alert()
- confirm()
- prompt()

3. methods that manipulate the Document Object Model (DOM):

document.write() - GENERALLY UNTRUSTED AS A SECURITY RISK

innerHTML – *commonly used*

innerText

textContent

So, the simplest method is simply a single-line console.log entry.

```
console.log(‘Hello World’);
```

An alert is also quite simple to use.

```
alert(‘Hello World’);
```

What about just using **Node**?

The console.log() method still works, and you can use alerts.

There is also a way to use **stdout** to do this, with before and after newlines.

Note that this is **only** in Node.js. It doesn’t exist outside of it.

```
process.stdout.write("\nHello World\n\n");
```

The general method is process.stdout(“CONTENT GOES HERE!”);

\n can be used inside double quotes for newlines.

For webpages only: Using the inner HTML is still simple, but does require a HTML environment to use it. So this method is more advisable for active webpages.

(in <script> tags or a .js file)

```
document.getElementById("demo").innerHTML = "Hello World";
```

(in a HTML file)

```
<!DOCTYPE html>
<html>
<body>

<div id="demo"></div>

<script>
document.getElementById("demo").innerHTML = "Hello World";
```

```
</script>


</body>
</html>
```

OK. So now we have three solid methods for both environments for programmers. We have methods for webpages only, and for using in Node. So let’s break them down.

- `console.log(“”)` and alert(‘’) methods work in both.
- `document.getElementById("DIV-NAME”).innerHTML` for webpages.
- `process.stdout.write` for Node.

So let’s present our solutions.

For both webpages and Node, the following methods work equally.

- console.log(“Hello World!”);
- alert(‘Hello World’);

For Node only:

```
process.stdout.write("\nHello World\n\n");
```

For webpages only:

```
document.getElementById("demo").innerHTML = "Hello World";
```

And that is is. The goal of a Hello World output in Javascript is achieved.