| **S.No** | **Javascript** | **NodeJS** |
| --- | --- | --- |
| 1 | Javascript is a programming language that is used for writing scripts on the website | NodeJS is a Javascript runtime environment. |
| 2. | Javascript can only be run in the browsers. | We can run Javascript outside the browser with the help of NodeJS. |
| 3. | It is basically used on the client-side. | It is mostly used on the server-side. |
| 4. | Javascript is capable enough to add HTML and play with the DOM. | Nodejs does not have capability to add HTML tags. |
| 5. | Javascript can run in any browser engine as like JS core in safari and Spidermonkey in Firefox. | V8 is the Javascript engine inside of node.js that parses and runs Javascript. |
| 6. | Javascript is used in frontend development. | Nodejs is used in server-side development. |
| 7. | Some of the javascript frameworks are RamdaJS, TypedJS, etc. | Some of the Nodejs modules are Lodash, express etc. These modules are to be imported from npm. |
| 8. | It is the upgraded version of ECMA script that uses Chrome’s V8 engine written in C++. | Nodejs is written in C, C++ and Javascript. |

**2. HOW DOES THE BROWSER ACTUALLY RENDER A WEBSITE:**

* They are three main steps are involved rendering process High level view, In-depth view and performance insight.
* Speculative Parsing which is used for look ahead,external images, script and CSS.
* Reentrant means the parsing process can be interrupt.
* They are 4 types of tree are there Render Object, Render Style, Render Layer and live box
* Painting computes bitmaps and composites to screen

**4.Execute the below code and write your description in txt file**

let a=1

console.log(typeof(a))

Output:

number

let a=1.1

console.log(typeof(a))

Output:

number

let a="1.1"

console.log(typeof(a))

Output:

String

let a=true

console.log(typeof(a))

Output:

boolean

let a=null

console.log(typeof(a))

Output:

object

let a=undefined

console.log(typeof(a))

Output:

Undefined

let a=[]

console.log(typeof(a))

Output:

object

let a={}

console.log(typeof(a))

Output:

object

let a=NaN

console.log(typeof(a))

Output:

Number

**5. PROTOTYPE:**

* That is, instead of creating a new object, some prototype is used which allows creating new objects by copying from this prototype.
* This could be useful when you want to copy an object which has a complicated state, which means that just initialising a new object will not be enough, you also need to reach that particular state of the object to consider it as a valid copy.