**DIFFERENCE BETWEEN BROWSER JS VS NODE JS CONSOLE.**

| **DEFINITION** | **NODE** | **BROWSER** |
| --- | --- | --- |
| **USAGE** | Front End runtime environment. | Back End runtime environment**.** |
| **Js Engine** | V8 engine it has. | Based on the browser V8 for chrome, Spider Monkey for Firefox. |
| **GLOBAL obj** | Global | Windows |
| **Synchronous activity(Js code)** | Single threaded | Single Threaded |
| **Module system** | As Usual common javascript. | ES 6 Module system |
| **Asynchronous Activity(Js code)** | Thread pool of 4 configured Managed by libuv. | Thread pool in case of chrome it is managed IO thread |

**TASK : 4**

let a = 1;

console.log (typeof(a));// number.

let b = 1.1;

console.log(typeof(b)); // NUMBER. Because it doesn't have any separate value like float and int(only number).

var c = '1.1';

console.log(typeof(c));// String.

var d = true;

console.log(typeof(d));// Boolean.

var e = null;

console.log(typeof(e));//null.

var f = undefined;

console.log(typeof(f));//undefined.

var g = [];

console.log(typeof(g));//object.

var h = ({})

console.log(typeof(h));// object.

var i = NaN;

console.log(typeof(i));// number.

**PROTOTYPE IN JAVASCRIPT:**

If we want to add a new property at the latest stage of function, With the help of protocol it can be done.

How does the browser actually render a website ?

Rendering is a process used in web developing that turns website code into The process is completed by a rendering engine, the software used by a web browser to render a web page. Because of its close association with web browsers, rendering engines are commonly referred to as browser engines.

How web pages got rendered:

**It first constructs DOM (Document Object Module)and CSSOM from raw code**

* While loading a web page, a web server sends a folder of files containing HTML, CSS, and JavaScript code to a user’s web browser.
* The browser engine converts this data (bytes) into characters (the HTML code).
* It parses the characters into tokens, which are further parsed into nodes.
* The browser engine links the nodes into a tree-like structure known as a Document Object Model (DOM). The DOM is the JavaScript representation of the HTML.
* Simultaneously, the browser converts the CSS code to a CSS Object Model (CSSOM) through a similar process.

**Uses Render Tree to create the end-user web page :**

* The browser engine combines the DOM and CSSOM to create a tree-like structure called a Render Tree. The Render Tree contains the style and content information browsers need to populate a webpage for viewers to see, calculate the layout for each visible element of a webpage and paint them on the screen for the end user's view.
* The layout operation is the next step. Using the Render Tree, the browser engine calculates the position of each visible element of a web page.
* Finally, the browser engine adds, or paints, the elements on the screen for the end-users view. The web page has now been rendered.

## **Dynamic Rendering**

JavaScript is a popular choice for rendering web pages because it is used to create an intuitive user experience. However, many Search engine bots struggle to process JavaScript readily. Consequently, websites that use JavaScript to house most of their content and navigation run the risk of being invisible to the web crawler. Dynamic rendering addresses this issue by rendering a web page as described above for the human user, while also rendering static HTML for a search engine bot to crawl and Index.

## **page render speed**

Page render speed refers to the time it takes to generate a page. Page render speed is measured from when the browser request is sent (when a user clicks on a link) to when the page becomes fully functional to the user. Page render speed and page load speed (the time it takes for users to see a webpage) are used interchangeably because both processes generally occur within split seconds of one another.

Low page rendering speeds increase and lower conversions. In one Google study, loading times for bounced sessions were 2.5 seconds slower than its non-bounced counterparts.

## **Importance of Rendering for SEO**

Webpage rendering affects how a page is indexed by bots and experienced by users. Understanding the impact of rendering on search engine rankings and SEO results should be a key consideration for any web development team.

In summary, for many websites (such as online retailers), building a website primarily in JavaScript may provide the most user-friendly and aesthetically engaging interface. However, search engine bots can struggle to crawl these websites, which negatively impacts search engine rankings and organic traffic. On the other hand, pages that render slowly negatively impact both user experience and search engine rankings.

These facts considered, web developers must pay careful attention to the requirements for bots and humans to maximize SEO performance.