1. Dfference between Browser JS(console) v Nodejs

1. JavaScript is a language.  
   Node.js is *not*a language or a special dialect of javascript.

it's just a thingamabob that runs normal JavaScript.

1. All browsers have javascript engines  that run the JavaScript of web pages.

Node.js is simply the V8 engine bundled with some libraries to do I/O and networking, so that you can use javascript outside of the browser, to create shell scripts, backend services or run on hardware .

1. JavaScript = script being run on  client-side.  
    Node.js = script being run on  server -side***.***
2. Javascript is a high-level programming language that uses the concept of Oops but it is based on prototype inheritance.

Nodejs allows Javascript code to run outside the browser. Nodejs comes with a lot of modules and mostly used in web development.

1. Mainly using for any client-side activity for a web application, like possible attribute validation or refreshing the page in a specific interval or provide some dynamic changes in web pages without refreshing the page.

It mainly used for accessing or performing any non-blocking operation of any operating system, like creating or executing a shell script or accessing any hardware-specific information or running any backend job.

2 SUMMERY POINTS

PARSING

RENDER / FRAME TREE

LAYOUT

POINT

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

typeof(1)

console.log(typeof(1)); - Number

typeof(1.1)

console.log(typeof(1.1)); -Number

typeof('1.1')

console.log(typeof('1.1')); -String

typeof(true)

console.log(typeof(true)); -Boolean

typeof(null)

console.log(typeof(null)); - Object

typeof(undefined)

console.log(typeof(undefined)); -Undefined

typeof([])

console.log(typeof([])); -Object

typeof({})

console.log(typeof({})); -Object

typeof(NaN)

console.log(typeof(NaN)); - Number

4. Prototype

A **prototype** is an early sample, model, or release of a product built to test a concept or process.

 It is a term used in a variety of contexts, including  semantics, design, electronics and software programming.

A prototype is generally used to evaluate a new design to enhance precision by system analysts and users.

 Prototyping serves to provide specifications for a real, working system rather than a theoretical one.

 In some design workflow models, creating a prototype

(a process sometimes called **materialization**) is the step between the formalization and the evolution of an idea.