1.Difference between Node.JS and Javascript

Node.js

- NodeJS is a cross-platform and opensource Javascript runtime environment that allows the javascript to be run on the server-side. Nodejs allows Javascript code to run outside the browser. Nodejs comes with a lot of modules and mostly used in web development.
- NodeJS is a Javascript runtime environment.
- We can run Javascript outside the browser with the help of NodeJS.
- It is mostly used on the server-side. Nodejs does not have capability to add HTML tags.
- Some of the Nodejs modules are Lodash, express etc. These modules are to be imported from npm.

JS(console)

- Javascript is a Scripting language. It is mostly abbreviated as JS. It can be said that Javascript is the updated version of the ECMA script. Javascript is a high-level programming language that uses the concept of Oops but it is based on prototype inheritance.
- Javascript is a programming language that is used for writing scripts on the website.
- Javascript can only be run in the browsers.
- It is basically used on the client-side. Javascript is capable enough to add HTML and play with the DOM.
- Some of the javascript frameworks are RamdaJS, TypedJS, etc.

2. watch & summary

How dose the browser actually render a website

- **High level flow**

  Parse HTML Parse CSS → Render Tree → Layout →paint

- **Parsing HTML**
  Parsing means analyzing and converting a program into an internal format that a runtime environment can actually run, for example the JavaScript engine inside browsers
- **DOM**
  The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects; that way, programming languages can interact with the page

- **Parsing Process**

  Tokeniser → Tree construction → Dom → Script Execution

- **Render Tree**
  The render tree captures both the content and the styles: the DOM and CSSOM trees are combined into the render tree. To construct the render tree, the browser checks every node, starting from root of the DOM tree, and determine which CSS rules are attached

- **Layout**
  Once the render tree is built, layout becomes possible. Layout is dependent on the size of screen. The layout step determines where and how the elements are positioned on the page, determining the width and height of each element, and where they are in relation to each other

- **Paint**
  The last step is painting the pixels to the screen. Once the render tree is created and layout occurs, the pixels can be painted to the screen. Onload, the entire screen is painted. After that, only impacted areas of the screen will be repainted, as browsers are optimized to repaint the minimum area required

3. Execute the below code and write description

- The typeof operator returns a string indicating the type of the unevaluated operand.

a. typeof (1)
   Output : number

b. typeof (1.1)
   Output : number

c. typeof ('1.1')
   Output : string

d. typeof (true)
   Output : boolean

e. typeof (null)
   Output : object

f. typeof (undefined)
   Output : undefined

g. typeof ([])
   Output : object

h. typeof ({})
   Output : object

i. typeof (NaN)
Output : number