**1.Difference between Browser JS and Node JS**

**Browser JS:**

* JavaScript can only be run in the browsers.
* JavaScript can be run on any engine, including Spider Monkey, V8, and JavaScript Core.
* JavaScript is capable to add HTML tags.
* It is generally used on the client-side server.
* It is designed to build network-centric applications.

**Node JS:**

* Node JS can run JavaScript outside of the browser.
* Node JS is only supported by the V8 engine, which Google Chrome mostly uses. Any JavaScript program written with Node JS will always be run in the V8 engine.
* Node JS does not have the capability to add HTML tags.
* It is generally used on the server-side.
* It is designed for data-intensive real-time application that

run on several platforms.

**2.Typeof**

* typeof (1) - Returns “number”
* typeof (1.1) - Returns “number”
* typeof ('1.1’) - Returns “String”
* typeof(true) - Returns “Boolean”
* typeof(null) - Returns “Object”
* typeof(undefined) - Returns “undefined”
* typeof ([]) - Returns “Object”
* typeof ({}) - Returns “Object”
* typeof (NaN) - Returns “Number”

**3.Summary points:**

* Introduction to browser and its rendering, platforms and JavaScript virtual machines.
* **High level flow** which shows the combine of parse HTML and parse CSS to render tree then to layout for positioning the stuffs and then the paint to style the document as visual result.
* Parsing means **analysing and converting a program into an internal format that a runtime environment can actually run**, for example the JavaScript engine inside browsers
* Then the parsing flow which describes the flow of compiling the element from a starting to closing tag called tokenizer.
* Then CSSOM is a set of API for reading and modifying CSS, then the DOM and CSSOM combines to represent the document on web pages.