1. **JS (Console) Verses Node.js**

**JS (Console):**

1. Initially released on December 4, 1995
2. **JavaScript** is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior.
3. JavaScript can run in any browser engine as like JS core in safari and Spider monkey in Firefox.
4. It is the upgraded version of ECMA script that uses Chrome’s V8 engine written in C++.
5. Some of the JavaScript frameworks are RamdaJS, TypedJS, etc.

**Node.js:**

1. Initially released on May 27,2009
2. **Node.js** is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.
3. V8 is the JavaScript engine inside of node.js that parses and runs JavaScript.
4. Node.js is written in C, C++ and JavaScript.
5. Some of the Node.js modules are Lodash, express etc. These modules are to be imported from npm.
6. **Summarization of Ryan Seddon: So how does the browser actually render a website:**

<https://www.youtube.com/watch?v=SmE4OwHztCc&ab_channel=JSConf>

1. Parsing HTML: There are few things about parsing HTML that are a bit different to most languages, you can make a lot of mistakes and it will just work, which means parsing isn’t straightforward.
2. Tokenizer:

Start Tag “< div >” End Tag”</ div >”. In Parsing, as each character comes in through the token, it will create a taken called ‘Start Tag’ then on the other end it will look for close tag and create token for the same

1. <script /> tag will stop the parser from going through all the HTML, constructor and DOM tree. By putting <script /> tag at the bottom, parse can go uninterrupted and page renders faster.
2. DOM (Document Object Model) and CSS Object model will go into and create what’s called render or the frame tree. DOM + CSSOM combines object models ands style resolution. This is the actual representation of what will be show on the screen.
3. A rendering engine calculates all the visual properties and it has to combine all the styles. There’s a lot of algorithms around matching rules for each element and do the style computation around CSS.

A cool Clip on Mozilla laying out a web page

<https://youtube.com/clip/UgkxdbmbXnXVxRecD-LhW1y0KA58v3WT0yE4>

1. **Execute the below code and write your description in txt file**
   1. typeof(1) number
   2. typeof(1.1) number
   3. typeof('1.1') string
   4. typeof(true) boolean
   5. typeof(null) object
   6. typeof(undefined) undefined
   7. typeof([]) object
   8. typeof({}) object
   9. typeof(NaN) number