

Javascript - Day -2 : Request & Response cycle

1)Five Differences Between Node JS and Browser JS

Browser JS	Node JS
We cannot access File System	Flexibility to access File System
Document, windows and other objects are present in Browser JS	We don't have Document and Windows objects in Node JS
Browser JS do not upgrade quickly, so there is no flexibility to use latest release ECMAScript	Flexibility to use latest version ECMAScript.
Uses ES modules standard	Uses CommonJS
JS runs on JavaScript Engine in Browser	Node JS is an Environment to run JavaScript on OS

2)watch & summary 5 points -

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

1. The browser actually renders a website by the parsing of the HTML and CSS, by the javascript then they will render a tree after that do the layout and painting.
2. Parsing is nothing but analyzing and converting a program into an format that a runtime environment actually run, for example javascript engine inside browsers.
3. The browser parses HTML into a DOM tree. HTML parsing involves tokenization and tree construction. HTML tokens include start and end tags, as well as attribute names and values. If the document is well-formed, parsing it is straightforward and faster. The parser parses tokenized input into the document, building up the document tree.
4. The parse tree will be having html, head, body, paragraph, text, division, spam, test similarly applying to DOM tree elements.

5. When the HTML parser finds non-blocking resources, such as an image, the browser will request those resources and continue parsing. Parsing can continue when a CSS file is encountered, but <script> tag particularly those without an async or defer attribute blocks rendering, and pauses parsing of HTML.
6. When the browser encounters CSS styles, it parses the text into the CSS Object Model i.e CSSOM a data structure i.e it is having selectors and declarations it then uses for styling layouts and paintings.
7. The DOM and CSSOM compiles the two objects models, style resolution and this is the actual representation of what will show on screen and it is not 1-to-1 mapping of our HTML.
8. The render tree is not having the non visual elements like head, script, title .etc., and nodes are hidden via display none.
9. The browser creates the layout of each individual render tree node and the layout consists of size of each node in pixels and where it will be printed in the screen. So the process we can call as recursive process or reflow process.
10. We are having list of geometries that need to be printed in the screen. So the elements in the render tree can overlap each other and they can have CSS properties that make them frequently change the look, position, or geometry (such as animations) the browser creates a layer for it.
11. Creating layers helps the browser efficiently perform painting operations throughout the lifecycle of a web page such as while scrolling or resizing the browser window. Having layers also help the browser correctly draw elements in the stacking order (along the z-axis) as they were intended by the developer.

4) Please check the script file for the code

Observed points after executing the code

- In JavaScript, integer number, float number and NaN(Not a Number) are considered as one data type i.e., ***number***

- Anything between single or double quotes are considered as data type ***string***
- True and false are considered as ***boolean*** data type
- Undefined variable are of type ***undefined***
- Remaining all are considered as type ***Objects***
Examples for object data type are [], { } , null etc.,