1.List 5 difference between Nodejs V Browser JS (console) .

Node js :

In Node everything is a module. You must keep your code inside a module.

* Node doesn’t have a predefined “window” object cause it doesn’t have a window to draw anything.
* “Location” object is related to a particular URL ; that means it is for page specific. So, node doesn’t require that.
* Of course Node doesn’t have “document” object also, cause it never have to render anything in a page.
* Node has “global”, which is a predefined global object. It contains several functions that are not available in browsers, cause they are needed for server side works only.
* “Require” object is predefined in Node which is used to include modules in the app.

Browser JS(Console) :

* “Window” is a predefined global object which has functions and attributes, that have to deal with window that has been drawn.
* “Location” is another predefined object in browsers, that has all the information about the URL we have loaded.
* “Document”, which is also another predefined global variable in browsers, has the html which is rendered.
* Browsers may have an object named “global”, but it will be the exact one as “window”.
* Browsers don’t have “require” predefined. You may include it in your app for asynchronous file loading.
* Modelling is not mandatory in client side JavaScript, i.e. in browsers.

2. Watch & Summary 5 points - <https://www.youtube.com/watch?v=SmE4OwHztCc&ab_channel=JSConf>

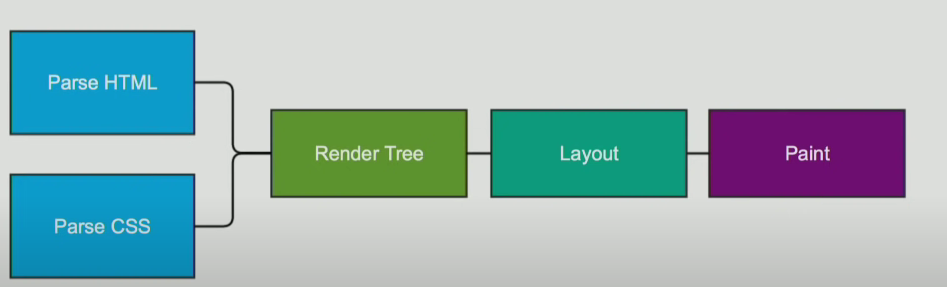
BROWSER

A computer program that lets you look at words and pictures from other computer systems by receiving information through telephone wires.

Components of Browser are

* Bindings
* Rendering : Parsing , Layout , Painting
* Platform
* Javascript Virtual Machine

High Level View Of How Browser Work

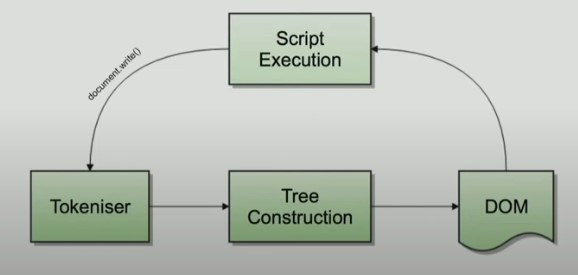


We now discuss about Rendering : Parsing , Layout , Painting.

Parsing

Parsing HTML is forgiving in nature. It is not straight forward , can be halted and if not at least it will do speculative parsing. Speculative parsing can look ahead and it can insert image and its links. This parsing process can be interrupted.

Parsing Flow



Parsing to Render / Frame Tree

Render Tree

It combines two object models ,styles resolution. This is the actual representation of what will show on screen. There are multiple trees namely render objects , render styles , render layers , line boxes. It does not contain non-visual elements head , script , title etc.

**Render tree to Layout**

Layout

It is a recursive process.

The process are

* Traverse render tree
* Nodes position and size
* Layout its children

It will batch layouts and it is Incremental layouts .The browser will intelligently batch change. Render trees items will flag themselves if wrong and the batch will traverse the tree and find all wrong trees. It is asynchronous process. And There is Immediate layouts where doing a font size change will relay out the entire document.

**Layout to painting**

Painting

Painting setup will take the layered out render trees. It creates layers and it is also a incremental process and it can build up and over 12 phases.

It produces a bitmap from each layer. Bitmap is uploaded to the GPU as a texture and composites the textures into a final image to render to the screen.

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

1. typeof(1)

It is a number

1. typeof(1.1)

It is a number

1. typeof('1.1')

It is a string

1. typeof(true)

It is a boolean

1. typeof(null)

It is an Object

1. typeof(undefined)

It is undefined

1. typeof([])

It is an object

1. typeof({})

It is an object

1. typeof(NaN)

It is number

Code :

