15 December 2025 09:47

Javascript = Interpreted Language

I need an Interpreter to RUN The JS file !!!

Ex.js ----------> JS Interpreter --------> translate a line and run it

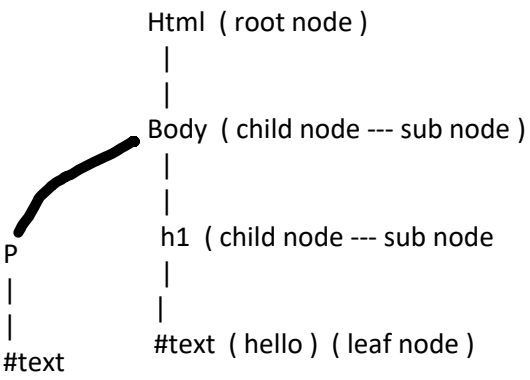
WEB Dev = **The user will interact through BROWSER**

Browser will RENDER the UI built in HTML

1. Browser gets the HTML
2. Browser Creates the DOM = Document Object Model
   DOM is a tree of html tags

```
<html>
    <body>
        <h1> Hello </h1>
        <p> This is a DOM Revision </p>
    </body>
</html>
```

DOM Tree



3. Browser RENDERS the HTML as per the DOM

Static HTML = a prewritten code
Dynamic HTML = DOM is not final after the first Render
DOM will keep changing
EVERY time the DOM changes
the browser will RERENDER the UI .

DOM changes = At User Interaction Time
The new nodes are added to the DOM
An existing node is Removed from the DOM
Modify the current node

DOM changes are ENABLED by Javascript (Client Side JS)

Client Side JS - has a special library object to represent the DOM = **document**

NODE in DOM that matches the ID = **document.getElementById("id1");**

document object in the JS - helps us TRAVERSE the DOM
Do the relevant modifications in the DOM

React ====> React Build (1 html , 2 JS files , CSS ,images)
|
|
Stored on the Web Server
|
|
GO to the Browser (Web client)
|
|
Browser will Use the JS
Generate DOM
Render UI

Angular = Write code in Typescript
|
|
Angular Build (1 htmls , few JS files, CSS , images)
|
|
Stored on Web server (Deployed)
|
|
Go to the browser
Browser will parse the JS
Generate a DOM
Render the UI

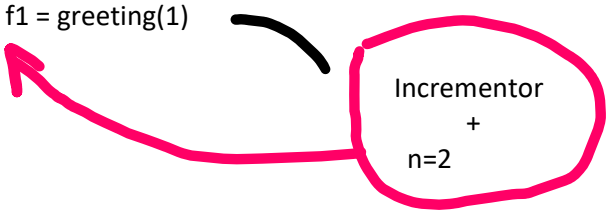
Where is the JS interpreter ?

1. Browser has inbuilt JS interpreter
2. Node interpreter = download node from node org !!!

Variables using let and var

Data types like	number , string, object ,boolean , undefined, function
In Javascript	The data type is INFERRED from the value
Functions	

Let f1 = greeting(1)



CLOSURE = function is returned alongwith its outer context

Destructuring syntax = convenience syntax

JSON objects

{ name : "sunil" , age: 23, city:"dehradun" }

JSON = can be sent as TEXT on the HTTP

Web Server	Http Server
Web Client	Http Client

Web Server <======HTTP========>Web Client
HTTP = ... TEXT.... .....

Web Server <====== Stringified JSON ====> Browser
TEXT

Spread operator = for copying objects

```
let arr = [1,2,11,22]
let arr2 = [ ...arr ]   //spread operator
```

SHALLOW COPY =
let remix = { ...album }
DEEP COPY
let remix = {...album, copyright:{...album.copyright}

Typescript = for the benefit of the Programmer
= it will ALLOW the programmer to specify data types.
= It will check that the values match the data types

= it will raise compile time errors in case of type mismatch
= support Generics , aliases for specifying types

Typescript (ex1.ts) ----->tsc (compiler) ---->ex1.js (output file)
ex1.js ------------>node(interpreter) --------> RUN

Compiler	Interperter
Translates from source code To target code	Translates from source code to target code
Generate an output file in target code That can be used many times	Does not generate any output file- everytime the code runs , it must be translated
We need an external executor to run the output file	Interpreter has an inbuilt executer to run the translated code

---

TO RUN TS on your machine

1. Create a folder - mytypescript
2. npm init }} to create a node project
   ask questions about the config file
3. npm install typescript
4. write ex1.ts
5. run the tsc compiler to translate the ts to js
6. run the js using node

OR

Typescript online compiler

```
function add(n1: number,n2:number ) : number
{
   return n1+n2
}
console.log(add(10,12))
```

```
//DEFINE A TYPE
interface MyUser
{
   name:string
   age? : number  //age is optional
}
interface Resident extends MyUser
{
   adhaar:string
}
//function greet(user:any) {
function greet(user:MyUser){
```

```
  return "Hello " + user.name +"  "+user.age //+user.city
}
const person = { name: "Prachi", age: 25 , city:"pune"};
console.log(greet(person));
let p2 ={name:"priya"}
console.log(greet(p2))
let p3 : Resident = {name:"prach"
                    ,age:12,
                   adhaar:"1234455",
                  }
```

```
class Pen
{
    private color:string
    private cost:number
    constructor(s1:string , c1:number)
    {
        this.color = s1
        this.cost = c1
    }
    getCost()
    {
        return this.cost
    }

    showDetails()
    {
        console.log(`the ${this.color} pen costs Rs ${this.cost}`)
    }
}
let arr:Pen[]=[]  //create empty array
arr.push(new Pen("red",5) , new Pen("blue",45),new
Pen("green",20))
let total:number = 0
for(let p of arr)
{
    total = total + p.getCost()
}
console.log("total=",total)
```