Javascript Concepts

Introduction

Javascript is a single threaded language runs syncronously used to create both client-side and sever-side code.

Execution context

How does the compiler executes the code What happens behind the scence?

It has two components:

- Memory (Enviroment Variable): Stores variable, and funtion in key value pair
- Code (Thread of execution)

Lets take an example how things work in Javascript

Here is the code

```
var n= 2;
function square(n){
    var ans= n*n ;
    return ans;
}
var square1= square(n);
var square2=square(4);
```

First phase: [MEMORY ALLOCATION] Javascript allocate the memory to the variables and functions
| Memory | Code | | ------ | ---- | n:undefined| f square(n){ var ans= n*n ;return ans;} ()| square1: undefined| square2: undefined|

Secont phase: [CODE EXECUTION] Javascript assigining the value to the variables | Memory | Code | | ------ | ---- | n:2| f square(n){ var ans= n*n ;return ans;} ()| square1: square(n) f initated| This creates a new Local execution cycle which again got through the phases and return the value 4. square2: square(4) f initated| This also does the same creates a new Local execution cycle which again got through the phases and return the value 16.

Hoisting in JAVASCRIPT

Hoisting is similar to as we discused above in the execution cycle the variable and function memory has been allocated and are assigned to undefined previously or the function itself.

```
Note:- Functions and var are being hoisted undefined or the function itself but let, const are initialized during code execution.[ Also arrow f() const fx=()=>{} are also declared and during the time of code execution cycle]
```

Lets learn it through example