01-05-2024

1.Execution Context :

In Javascript everything happens in **execution context**. It is an environment that handles execution of code. It has two components

- **Memory Allocation** : Memory component is also known as a variable environment which contains variables and functions as the key-value pairs. It allocates memory for variables as undefined and function references. Example of memory allocation is

Memory allocation	Code Execution
a=undefined demo(){…..} b=undefined	

- **Code Execution :** In this phase, the JavaScript code is executed one line at a time inside the Code Component (also known as the Thread of execution or single thread synchronous) of Execution Context.

Memory allocation	Code Execution
a=undefined demo(){…..} b=undefined	Var a=20 Console.log(a)//20

Example for Execution Context:

```
var n = 5;
function square(n) {
  var ans = n * n;
  return ans;
}
var square1 = square(n);
var square2 = square(8);
console.log(square1)
console.log(square2)
```

i. Creation Phase:

Memory allocation	Code Execution
n : undefined square() {.....} square1 : undefined square2 : undefined	

ii. Execution Phase :

Memory allocation	Code Execution
n : 5 square(){.....} square1 : undefined square2 : undefined	<table><tr><th>Memory</th><th>Code</th></tr><tr><td>n : undefined ans : undefined</td><td></td></tr></table>

Once the calculation is done, it assigns the value of square in the 'ans' variable that was undefined before. The function will return the value, and the function execution context will be destroyed.

The returned value from square() will be assigned on square1. This happens for square2 also. Once the entire code execution is done completely, the global context will look like this and it will be destroyed also.

Memory allocation	Code Execution
n : 5 square() {.....} square1 : 25 square2 : 64	