Date
Page

Javascript Execution Code Context.

- An execution context is an abstract environment where JS code runs.
- Each time a script runs, a context is created.

* Three types are :-

1. Global Execution Code (GEC) :-
- Created when 1st page loads.
- Creates window obj. (in browsers) or global object (in Node.js).
- Sets up this keyword to point to that global object.

2. Function Execution Code (FEC) :-
- Created when function is called.
- Has its own scope & this binding.

3. Eval Execution Context
-

Execution Context Life cycle :-

1. Creation Phase (Memory Phase)
- variable, functions, arguments are hoisted.
- this is set

2. Execution Phase
- Code is executed like by line
- Variable values are assigned.

Date
Page

Steps

1 ⇒ Global Execution Context created
↳ this is set.

2. Memory Phase
- val1 - undefined
- val2 - undefined
- addnum - defination of function
- result 1 - undefined
- result2 - undefined.

3. Execution Phase
val 1 - 10
val 2 - 5
add new
result 1 = 15.

Program

```
let val1 = 10
let val 2 = 5
function addNum1
    raadn1, num2) {
let total = num1+num 2
return total.
}
let result1 = addNum
    (val1, val2)
let result 2 = addNum
    (10,2).
```

new variable environment + execution thread

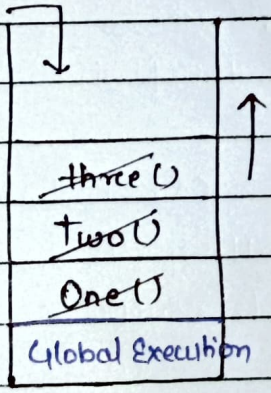
→ Memory
num 1 - undefined
num2 - undefined
total - Undefined

→ Execution.
num1 = 10
num 2 = 5
result = 15
this will be returned in Global Execution Context to variable

- new variable environment is deleted

result 2 = 12 → new variable environment + execution thread → Memory:
num1 = undefined
num2 = undefined
total = undefined
→ Execution
num1 = 10
num2 = 2
returned ← total = 12
→ Deleted ~~exit~~ environ-~~ment~~ment.

Call Stack :-



Program
One ( ) {
two () {
three { }
}
}