Important Concept in Javascript.

Javascript Execution Context

```
┌──────────┐
│   Js     │     Converted
│  code    │  ───────────────▶     Global Execution Context
│  file.   │                            ▲
└──────────┘                            │        Stored in "this"
                                  (this)          Variable.
```
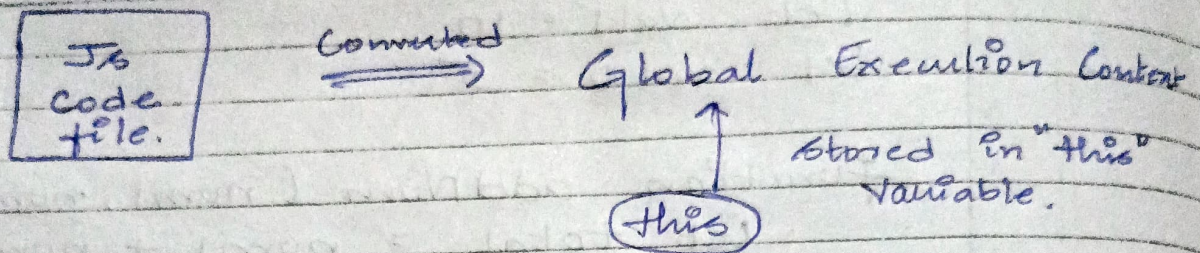
NOTE :- Different Environment has different Global Execution Context, but in browser the global execution Context is referred to variable "this" which is a <u>window</u> <u>Object</u>
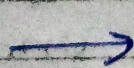
There are usually two phases Context

↳ Global Execution Context

↳ Function Execution Context

phases are :-

```
┌──────────┐
│   Code   │  ──────────▶   Memory Creation phase
└──────────┘

              ──────────▶   Execution phase.
```

Let's understand with small Example.

```
Let  val1 = 10.
let  val2 = 5

function addNum (num1, num2) {
    let total  = num1 + num2
    return total
}

Let  result_1 =  addNum (val1, val2)
let  result_2 =  addNum (6, 5)
```

Execution Steps :

① Global Execution.
          ↓
        (this)

② Memory Creation phase.

```
Val1  →  undefined
Val2  →  undefined
addNum  →  defination      ⎫  cycle OT
result1  →  undefined       ⎬
result2  →  undefined.      ⎭
```
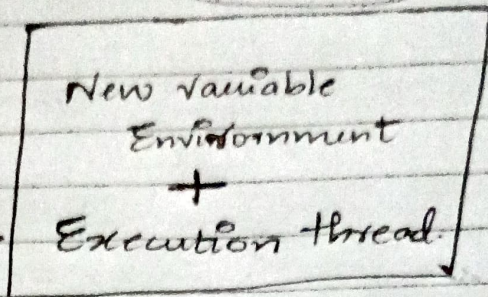
Execution phase

    Val1 ← 10

    Val2 ← 5

    addNum ──────→

(Sandbox)

New variable Environment
+
Execution thread

The memory phase of Execution phase takes place for addNum function

──→ Memory phase

    Val 1 ──→ undefined

    Val 2 ──→ undefined

    total ──→ undefined

① Global Execution

returns

──→ Execution phase.

    num1 ──→ 10

    num2 ──→ 5

    (total) ──→ 15

⇒ After the Execution of function the New Sandbox is deleted automatically