Day-3 (Additional):

1. Hoisting
2. Scope
3. loops

Recap:

Hoisting :

→ declaration

→ definition/initialization

} Variable

Var x ; (declare)

x = 10; (initialize/define)

index.js

```
Var x = 10;
let name = 'Mani';
Var y;
  y = 20;
Const z = 30;
```

Js Engine (Node.js)

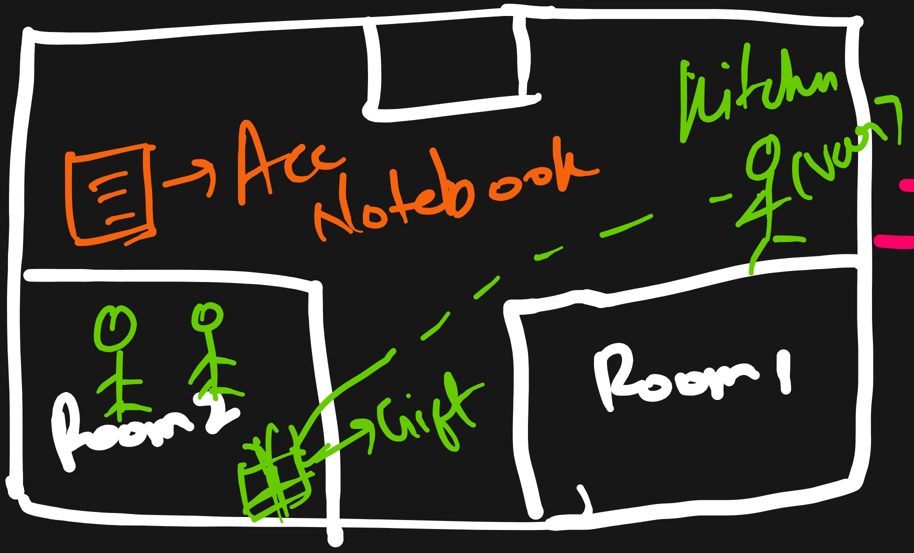
```
x = undefined
y = undefined
```

"node index.js"

Hoisting:

"Hoisting is JavaScript's default behaviour of moving declarations to the top of the current Scope (Script, function, file)".

My house : (Address)



⇒ 6 Numbers ✓
(Address)

⇒ Neighbours × ✓

⇒ Postman × ✓

⇒ delivery boy × ✓

⇒ Milk man × ✓

```
let address;
let accNotebook;
let gift;
```

Global Varible : Address

Local/function Scope : Acc Notebook

Block Scope : Gift (let & const)

is not applicable to "var"

var

↓

═══

letkconst

↑

Loops:

→ For loop

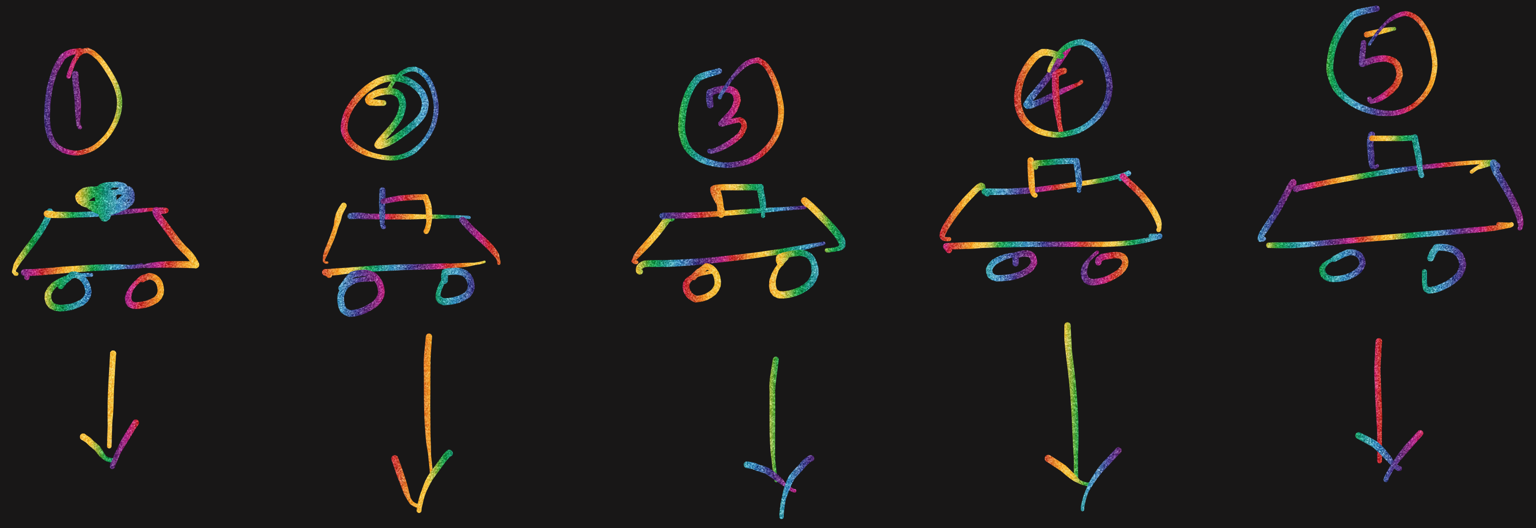
→ While loop

→ Do-while loop

→ For-in loop

→ For-of loop

For loop:



" loops through a block of code a number of times "

```
function  functionName() {
      // block of code
  }
```

Syntax for function!!

for loop Syntax :-

```
for (statement 1 ; statement 2; statement 3)
  {
     =
     // code block to be executed!!
     =
  }
```

Statement 1:
"is executed (one time) before the execution of the code block"

Statement 2: defines the condition for executing the code block

statement 3: "is executed (every time) after the code block has been executed".