**TRAINING**

**03/01/25**

**Local repo :** Divided into 2

* Un-tracked files : 1. Create

2. Update

3. Delete

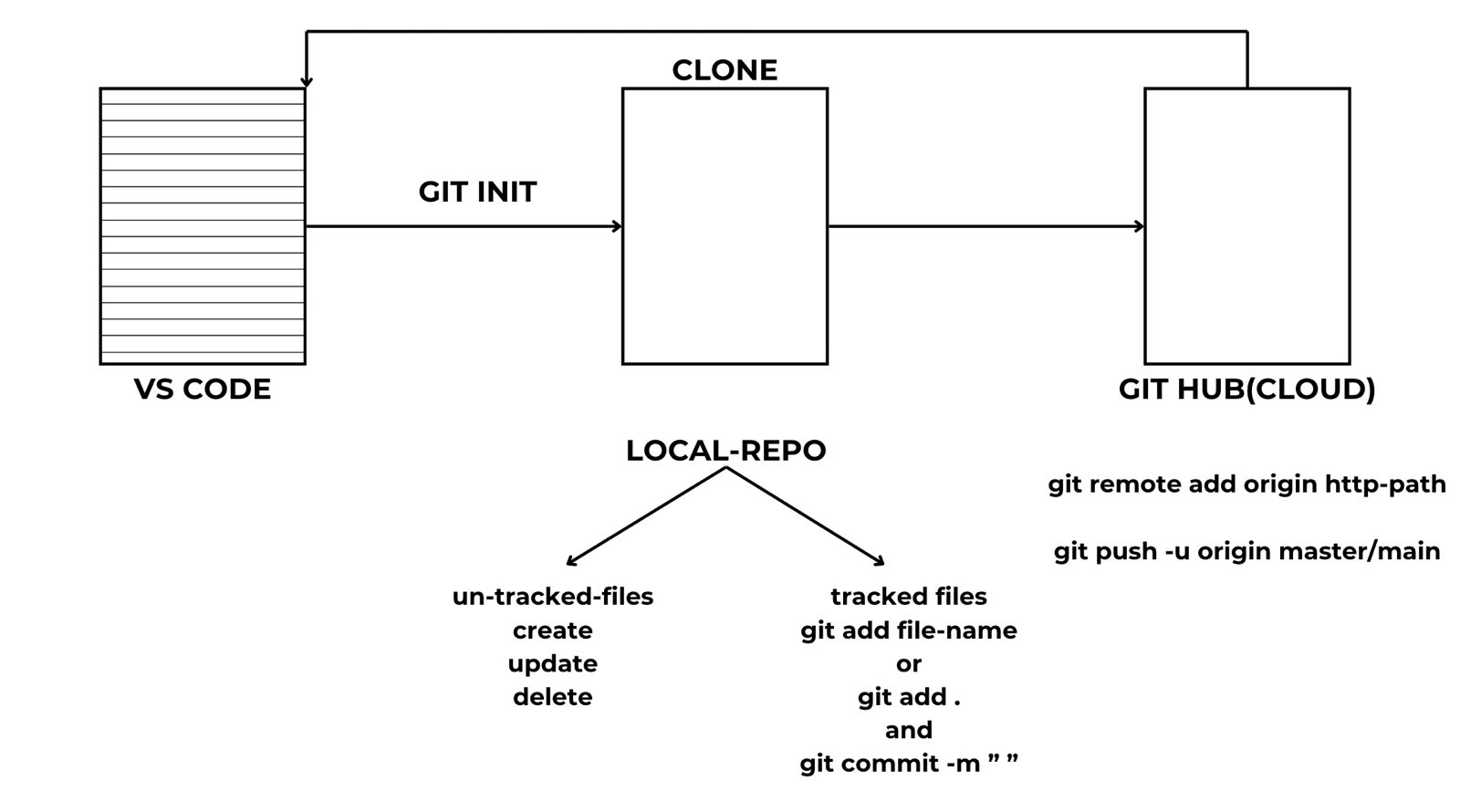
* Tracked files : 1. git add file-name - this command is used to add individual files.

(or) git add . - this command is used to add multiple files.

2. git commit -m ""

**Github (cloud) :**

* **Git remote add origin http-path**
* **Git push –u origin master/main**

****

**04/01/25**

* React native is used to develop Mobile application.
* Electron JS is used to develop Desktop application.
* React JS is used to develop Web application.

**What is JavaScript ?**

* It is used to convert static page into dynamic page.
* It is used to develop dynamic websites.
* It is a dynamically typed language.

**Variables in JS :**

* Set of blocks used to store data/values (data types). (any kind of data)
* Dynamic in nature (because there is no need of declaring any data types).
* They are case sensitive.
* Can start with letters, under score (\_), dollars($).
* Cannot start with numbers.
* Reserved words (keywords) are not allowed.
* In variables, we have 3 types of scopes :

1. **Global scope:** we can declare variables outside the function, and can access inside the function.
2. **Local scope/script scope :** if we declare a variable inside a function.
3. **Block scope** : let is possible, var is not possible.

* Variables can be declared using 3 keywords :

i. var (1995-2015) (used only in old browsers)

ii. let (2015)

iii. const

* There are 2 types of variables :

1. Static type
2. Dynamic type

**06/01/25**

* **Static websites :** Remains same to everyone.

Ex : fb login.

* **Dynamic websites :** Changes to everyone (data).

Ex : Youtube, Cricbuzz, Google maps #time-time#, Instagram.

**Reassigning variables :**

* Reassigning variables is possible by ‘var’ and ‘let’.

Ex 1: var a=1;

      a=5;

      document.write(a+a);

      console.log(a+a);

Output : 10

Ex 2 : var fav\_std = "ramya";

        fav\_std = "kavya";

        fav\_std = "sam";

        document.write(fav\_std);

        console.log(fav\_std);

Output : sam

Ex 3 : let fav\_std = "ramya";

        fav\_std = "kavya";

        fav\_std = "sam";

        document.write(fav\_std);

        console.log(fav\_std);

Output : sam

* Reassigning variables is not possible by ‘const’.
* The latest value assigned to the variable will be the output.

**Redeclaring variables :**

* Redeclaring with ‘var’ is accepted.
* Redeclaring with ‘let’ and ‘const’ is not accepted.

**07/01/25**

**Data types in JS :**

To check the data type - **typeof**

2 types:

1. Primitive Data types:

* Pre-defined.
* We can store single values.
* Ex:

1. Number (integers, floats)
2. Boolean (true or false)
3. String (Stream of characters enclosed in quotes : single, double or backtick quotes)

* Single and double quotes works the same.
* Backtick quotes were introduced in ES6 version.
* It provides extra functionality in which we can insert the variables.
* Ex (1):

let a = 1

console.log("hi +a+a")

document.write('hi + a+a' , "<br>")

document.write(`hi+ ${a+a}`)

console.log(`hi +${a+a}`)

**Output:** hi+a+a  
 hi+ 2

* Ex(2):

var a = "sai"

var b = "kumar"

var age = 30

let c = `I am ${a+” ”+b}, my age is ${age} years old`

console.log(c);

document.write(c);

**Output :** I am saikumar, my age is 30 years old

1. Undefined (declared variable with no value).
2. Null (type of null in JS is object) – empty value or null value

* Ex:

let a = null

document.write(a , "<br")

document.write(typeof(a))

**Output:** null

1. Big int.
2. Non Primitive data types:

* Ex:

1. Class
2. Objects
3. Arrays
4. Functions
5. Maps
6. Sets

**08/01/25**

**JavaScript Operators:**

1. **Arithmetic Operators :**

* Multiplication - \*
* Division - / : gives quotient as the result
* Modulus Operator - % : gives remainder as the result
* Exponential (raised to the power) - \*\*
* ++ : increments by +1
* -- : decrements by -1
* Ex:

let a=4

let b=5

a++

b—

document.write(a, “<br>”)

document.write(a, “<br>”)

document.write(a\*b , "<br>")

document.write(a\*\*b, "<br>")

document.write(a/b, "<br>")

document.write(a%b, "<br>")

Output :

5  
4  
20  
625  
1.25  
1

1. **Assignment Operators:**

* Ex:

let a=5

let b=2

a+=40 //a=a+40 a=5+40=45

b-=5 //b=b-5

document.write(a , "<br>")

document.write(b , "<br>")

Output:

45  
-3

1. **Logical Operators:**

* Ex:

let a=5

let b=2

document.write(a>b && a>2 , "<br>")

document.write(a>b && a>12 , "<br>")

document.write(a>b || a>12 , "<br>")

document.write(!(a>12) , "<br>")

Output:

true  
false  
true  
true

1. **Relational Operators:**

* They are also known as Comparison Operators.
* “==” operator doesn’t check for the data type of both the values.
* “===” operator checks the data type of both the values and gives the output accordingly.
* Ex:

let a=5

let b="5"

document.write(a==b, "<br>")

document.write(a!=b, "<br>")

document.write(a===b, "<br>")

**document.write(a!==b, "<br>")**

Output:

true  
false  
false  
true

1. **Conditional Operators:**

* Ex:

let t = 40

let r = (t>40)?'Ac-off':'Ac-on'

document.write(r, "<br>")

Output:

Ac-on

**Window Methods:**

1. Prompt – by default it is considered as string.
2. Alert

Ex:

//alert("Zoro")

//document.write(alert)

//a=prompt("Enter Your Name")

//alert(a)

let aa = prompt("Enter your name")

alert(`My name is : ${aa}`)

**09/01/25**

**Conditional Statements:**

* To make the set of instructions (block of code)(conditional block) execute only when the given condition is True.
* Used to decide whether the code has to be executed or skip based on the given condition.
* Line-Line(sequence)
* **Block of code:** set of instructions.
* It will execute only when that specific condition is True.
* **Condition:** An expression that evaluates a result(True or False).
* **Ex:** console.log(5>6)
* **Methods:**

1. **If:**

* Only one possible condition.
* If the condition is true, it will execute otherwise it skips.
* The outer block doesn’t look at the condition, it will always execute.
* Ex:

let a=5

let b=10

if(a<b){

document.write("Hi Luffy","<br>")

}

document.write("Welcome to INDIA") //outer block

Output:

Hi Luffy  
Welcome to INDIA

1. **If-else:**

* Ex:

let a=5

let b=10

if(a>b){

document.write("Hi luffy","<br>")

}else{

document.write("Zoro","<br>")

}

document.write("Welcome to INDIA")

1. **Else if:**

* Ex(1):

age = prompt("Enter your age")

name = prompt("Enter your name")

if(age<18){

alert(name+ ", You are too young to marry")

}else if(age>35){

alert(name+ ", You are too old to marry")

}else{

alert(name+ " Perfect age to marry")

}

alert("Welcome to India")

* Ex(2):

let a = prompt("Enter the A score")

let b = prompt("Enter the B score")

if(a>300 || b>300){

alert("You can team up")

}

else if(a+b<500){

alert("You can team up")

}

else{

alert("You can't team up")

}

1. **Switch:**
2. **Ternary operator**

**20/01/25**

**LOOPS:**

* For loop
* For in
* For of
* While
* Do while

1. **For loop:**

Syntax:

For (initialization; condition; update-expression){

---block of code

}

**Ex:**

<script>

        //ex1

        for(let i = 0; i<7; i=i+2){

            document.write(i+2)

        }

        //ex2

        for (let i=0; i<7; i=i+2){

            document.write(i+2)

        }

        //i=0 0<7 True, 0+2 = 2

        //i=2 2<7 True, 2+2 = 4

        //i=4 4<7 True, 4+2 = 6

        //i=6 6<7 True, 6+2 = 8

        //i=8 8<7 False, loop ends

    </script> //OUTPUT : 2468

//ex3

        //for(let i = 10; i<=15; i++){

            //document.write(i, "<br>")

        //}

        //ex4

        //for(let i=1; i<=5; i++){

            //document.write("dakshuu", "<br>")

        //}

        //ex5

        for(let i=1; i<=10; i++){

            document.write(i\*\*2)

        }

        //ex6

        document.write("<br>")

        for(let i=1; i<=10; i++){

            document.write("9 x "+i+"= "+ i\*9, "<br>")

        }

//ex7 adding the numbers in array

        const numbers = [1,2,3,4,5]

        let sum = 0

        for (let i=0; i<numbers.length; i++){

            sum = sum + numbers[i]

        }

        document.write(sum)

        //0+1=1 #0 = 1

        //1+2=3 #1 = 2

        //3+3=6 #2 = 3

        //6+1=4 #3 = 4

        //10+5=15 #4 = 5

**21/01/25**

1. **For in loop:**

Ex:

//ex1

        const a = "dakshita"

        for (let i in a){

            console.log(a[i])

            document.write(a[i])

        }

        //ex2

        const s = "itachi"

        for(let i in s){

            console.log(s[i])

        }

1. **For of loop:**

Ex:

const arr = ["itachi","obito","shisui"]

        for (let e of arr){

            console.log(e)

        }

1. **While loop:**

* We use while loop when we don’t know the number of iterations.
* Syntax:

While(condition){

//block of code

}

* Ex:

//while loop

let i=2, n=8

while(i<=n){

console.log(i+1)

i=i+2

}

i++;

1. **Do while loop:**

* Syntax:

Do{

//block of code

}

While(condition){}

* Even when the condition is false, the while loop will execute atleast once.
* Ex:

//do while loop

let c=0

do{

console.log("count is:" + c)

c++

while(c>5)

}

//do while loop

        //let c=0

        //do{

            //console.log("count is:" + c)

            //c++

            //while(c>5)

        //}

        for(let i=1; i<=8; i=i+2){

            if(i==4){

                break;

            }

            console.log(i)

        }

        for(let i=1; i<=8; i=i+2){

            if(i>=4){

                break;

            }

            console.log(i)

        }

//ex4

for(let i=1; i<=8; i=i+1){

            if(i>=4){

                continue;

            }

            console.log(i)

        }

**22/01/25**

Ex:

//ex1

for(let i=1; i<=8; i=i+1){

if(i==5){

break;

       }

       console.log(i)

}

//ex2

for(let i=1; i<=8; i=i+2){

            if(i>5){

                continue;

            }

            console.log(i)

        }

        //Output: 1 3 5

**Nested Loops:**

* A loop inside another loop is called as Nested Loops.
* The inner loop will execute one time for each iteration of outer loop.
* An inner loop within the repeating block of outer loop.
* Ex:

//outer loop

for(let i=1; i<=3; i++){

document.write(i, "<br>")

//inner loop

for (let j=1; j<=4; j++){

document.write(j, " ")

}

document.write("<br>")

}

Ex 2: Printing patterns

//right angle pattern

        for(let i=1;i<=5;i++){

            for(let j=1;j<=i;j++){

                document.write("\*")

            }

            document.write("<br>")

        }

//triangle pattern

        let n=5;

        for(let i=1; i<=n; i++){

            for(let j=1; j<=n-i; j++){

                document.write("&nbsp;&nbsp;")

            }

            for(let j=1;j<=2\*i-1;j++){

                document.write("\*")

            }

            document.write("<br>")

        }

**23/01/25**

Topics to be covered:

* Functions
* DOM
* Events
* Mini project
* Major project
* HTML, CSS, JS, React JS
* JAVA, Spring Boot, My SQL
* Web applications
* Front-end
* Database, JAVA

**Functions:**

* A function is a reusable block of code.
* It can be called anywhere in the program.
* **Reusable code:** using an existing code without writing it ever we need.
* You can use the same code with different arguments to get the results.
* Syntax for writing the function in JS:

Keyword = function

()

Parameters = value

(Parameters are the values which are declared inside the parenthesis while defining a function)

* **Arguments** are the values which are passed through the parameters.

Ex:

        //ex1 without parameters

        function greet(){

            console.log("Hello Dakshita")

        }

//output: Hello Dakshita

        //ex2

        greet()

        console.log("Hello Dakshuu")

        greet()

//output: Hello Dakshita

Hello Dakshuu

Hello Dakshita

//ex3 with parameters

        function g(name){ //name is a parameter here

            console.log("Hello " + name)

        }

        let name = prompt("Enter your name")

        g(name)

//EX4 adding 2 numbers using functions

        function sum(number1, number2){ //name is a parameter here

            console.log(number1+number2)

        }

        sum(10,20)

**Function Return:**

* The return statement can be used to return the value to function call.
* Ex:

Function a(aa,bb){

return aa+bb;

}

//ex5

        function sumtwo(a,b){

            return a+b

        }

        let n1 = parseFloat(prompt("Enter number1"))

        let n2 = parseFloat(prompt("Enter number2"))

        let result = sumtwo(n1,n2)

        console.log("The sum is: " + result)

**Anonymous Function:**

* A function without name, after we create a function without a name and we assign it to a variable.
* Ex:

//ex6

sum = function(n1, n2){

let r = n1+n2

return r

}

console.log(sum(5,10))