**TRAINING**

**03/01/25**

**Local repo :** Divided into 2

* Un-tracked files : 1. Create

2. Update

3. Delete

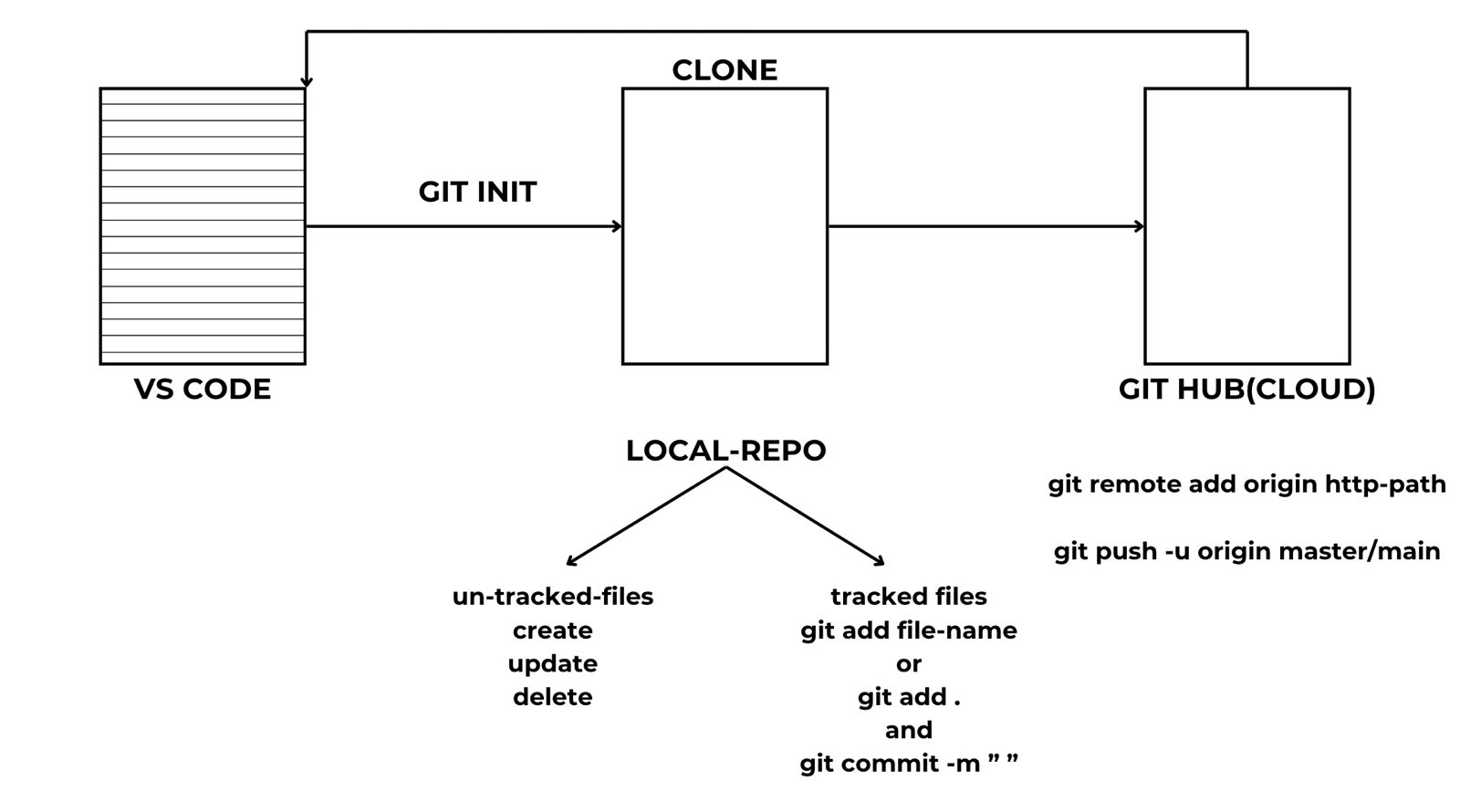
* Tracked files : 1. git add file-name - this command is used to add individual files.

(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

**Conditional Statements:**

* **Condition:** An expression that evaluates the result (true or false).

Console.log(5>4): true

* To make the set of instructions execute only when the condition is true.
* **Block of code(condition Block):** Set of instructions
* When this block of code will execute it will execute when the given specific condition is true.

**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}`)