# 2024-03-21

# Basic introduction:

What is javascript ?

- It is a programming language

VS code :

- It is free and popular editor by Microsoft

Our first js code :

console.log(“hi”);

Variables :

- They are the containers for data

# Arrays:

An array is a collection of items of same data type stored at contiguous memory locations

let selectedcolors = [‘red’ , ‘blue’];

console.log(selectedcolors);

# js is a dynamic language , so the type of the variable can change at run time , the same is with the arrays and their objects

selectdcolors[2] = ‘green’;  -à this makes the third element in the array to be green

# we can store heterogeneous data types in an array , i.e. diffn data types

#arrays are known as objects in JavaScript

Typeof selectedcolors àshows the type of the array , i.e objects

console.log(selectedcolors.length);àlength of the array

# Functions:

A set of steps to do a certain thing

Block of code that performs various tasks

```
function greet(){

        console.log(‘hello’);

}

greet(); à calling the function

function greet(name){  àparameter

console.log(‘hello’ + name);

}

greet(‘john’); àargument
```

A parameter is a variable in a method definition.

When a method is called, the arguments are the data you pass into the method's parameters.

# Strings:

String is a primitive data type

# Comparison:

In JavaScript, the == and === operators are used for equality comparison, but they behave differently:

== (Abstract Equality Operator):

The == operator performs type coercion (also known as type conversion). It converts the operands to the same type before comparing them.

If the operands have different types, JavaScript tries to convert them to a common type.

For example:

0 == false evaluates to true because false is converted to a numeric value (0).

1 == "1" evaluates to true because JavaScript automatically converts the string to a number for comparison.

null == undefined evaluates to true because both are considered equal in this context.

'0' == false evaluates to true due to type coercion.

However, be cautious with == as it can lead to unexpected results.

=== (Strict Equality Operator):

The === operator performs strict comparison without type coercion.

It checks whether the operands are of the same type and have the same value.

For example:

0 === false evaluates to false because they are of different types (number vs. boolean).

1 === "1" evaluates to false because they are different types (number vs. string).

null === undefined evaluates to false because they are distinct types.

'0' === false evaluates to false because no type coercion occurs.

Use === when you want to compare values without any type conversion.

Remember:

== is coercive equality (with type conversion).

=== is strict equality (without type conversion)

## IF else:

```
if (condition) {

        statement

}

else if (condition) {

        Statement

}

else

        statement

let hour = 10;

if (hour >= 6 && hour <12){

        console.log (‘good morning’);
}

else if (hour >= 18 && hour < 18){

        console.log(‘good afternoon’);

}

else

        console.log(‘good evening’);
```