WEB TECHNOLOGY WORKSHOP 2
DAY 9

JAVASCRIPT

- **Lightweight**: JavaScript is a lightweight programming language, which means that it can run on almost any device without requiring a lot of memory or processing power
- **Cross – platform**
- **Object oriented**
- **Dynamic**
- **Interactive**

JAVASCRIPT IS SCRIPTING LANGUAGE

- All scripting languages are programming languages.
- The scripting language is basically a language where instructions are written for a run time environment. They do not require the compilation step and are rather interpreted.
- NODE JS , PYTHON, RUBY,PEARL
- Scripting languages are used in web applications.
- It is used in server side as well as client side.
- Server side scripting languages are: JavaScript, PHP, Perl etc.
- Client side scripting languages are: JavaScript, AJAX, jQuery etc.

JAVASCRIPT VERSIONS

- JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997. (a standard for scripting languages)
- ECMAScript is the official name of the language.
- ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, and ES6(MODERN JS).
- Since 2016, versions are named by year (ECMAScript 2016, 2017, 2018, 2019, 2020)
- VanillaJS is a name to refer to using plain JavaScript without any additional libraries like jQuery back in the days.

BASIC BUILDING BLOCK

- VARIABLE
- DATA TYPES
- STATEMENTS
- FUNCTIONS

VARIABLE

- **Variables are identifiers used to store the data and information**
- **Variables can be defined using the below keywords:**
- **let**
- **Const**
- **Var**

let: This is the recommended way of defining variables that remain valid in the block in which it is defined, and its value can be changed and reassigned.

let name;

let age;

let num1,num2,num3;

const: This is used whenever the containing value is constant and will not undergo change. This is also block scoped. const pi=3.14;

var: This is the traditional way of defining variables in JS, which is not block scoped but function scoped.

var dummy;

SCOPES

- **Scope in JavaScript tells us which variables will be accessible at a given point.**
- **There are two kinds of scope – global scope and local scope.**
- **Global scope**
- **Any variable which is declared outside any function is accessible anywhere in the code, even in the functions and is in the global scope.**
- **const global = 'Hi! I am Global';**

Local scope

- **Function scope: When a variable is declared within a function, it is accessible only within the function. You can't access this variable once you are out of the function.**

```
function HelloWorld () {
const hello = 'Welcome to JS!';//this is function scoped
console.log(hello);}
HelloWorld (); //'Welcome to JS!'
console.log(hello);// This will give error hello is not defined
```

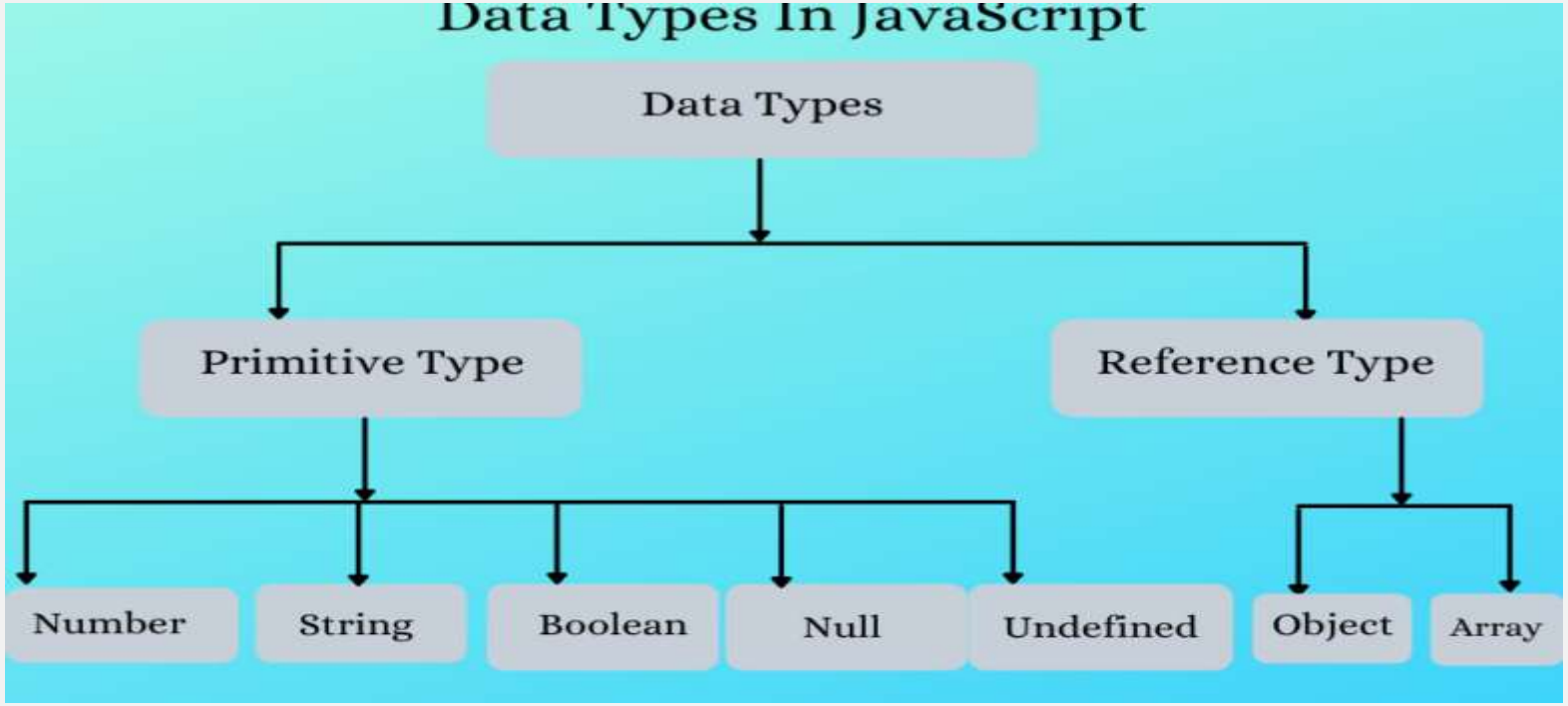
- **Block scope:** When a variable is declared using the const or let keyword, within a block of curly brace ({}), it is accessible only within that curly brace.

```
{

const hello = 'Welcome to JS!';//this is block scoped

console.log(hello);

}

console.log(hello);// This will give error hello is not defined
```

QSN

```
Var a=28;
Var b=john;        // let
{
Var b =“this”      //let
c.l(b);
}
c.l(b)
//output
```

DATA TYPES


Data Types In JavaScript
Data Types
Primitive Type
Reference Type
Number
String
Boolean
Null
Undefined
Object
Array

QSN

1. Create a variable of type string and try to add a number to it.
2. Use typeof operator to find the datatype of the string in last question.
3. Create a const object in js , can you change it to hold a no. later?
4. Try to add a new key to the const object in qsn 3 , were you able to do it?
5. Write a js program to create a word meaning dictionary of 5 words.
6. Use logical operators to find whether the age of a person lies between 10 and 20?
7. Demonstrate the use of switch case statements in js.
8. Write a js program to find whether a number is divisible by 2 and 3.
9. Write a js program to find whether a number is divisible by 2 or 3.
10. Print “you can drive “ or “you can’t drive” based on age being greater than 18 using ternary operator.s