**Frontend**

* User interface where the users can see and interact with the application
* Client side
* Creating websites, designs, structure

**JavaScript**

1. It is a object based and object oriented programming language used to convert static page into dynamic page.
2. Static and Dynamic websites:

* Static websites remains same to everyone (FB login)
* Dynamic websites changes to everyone (Facebook data, Youtube data, Criczbuzz, Google maps)

1. Scripting language
2. Used to develop dynamic websites
3. Often used for form validations
4. Current version “ES13”
5. Java script code is written inside the script tag
6. Two ways to run :

* Using Browser
* Using Node.js

**Datatypes**

It is a value used in programming language ,two types primitive and non primitive

* Primitive datatype:

String, Number, Boolean, Undefined, Null, Symbol, and BigInt.

* Non primitive datatype:

Object, Array, Function, RegExp

**Variables**

* used to store value/data
* dynamic in nature
* no need to mention the datatype
* 3 types of variables (var, let, const)

Ctrl shift j – console in browser

Break point

**Scope**

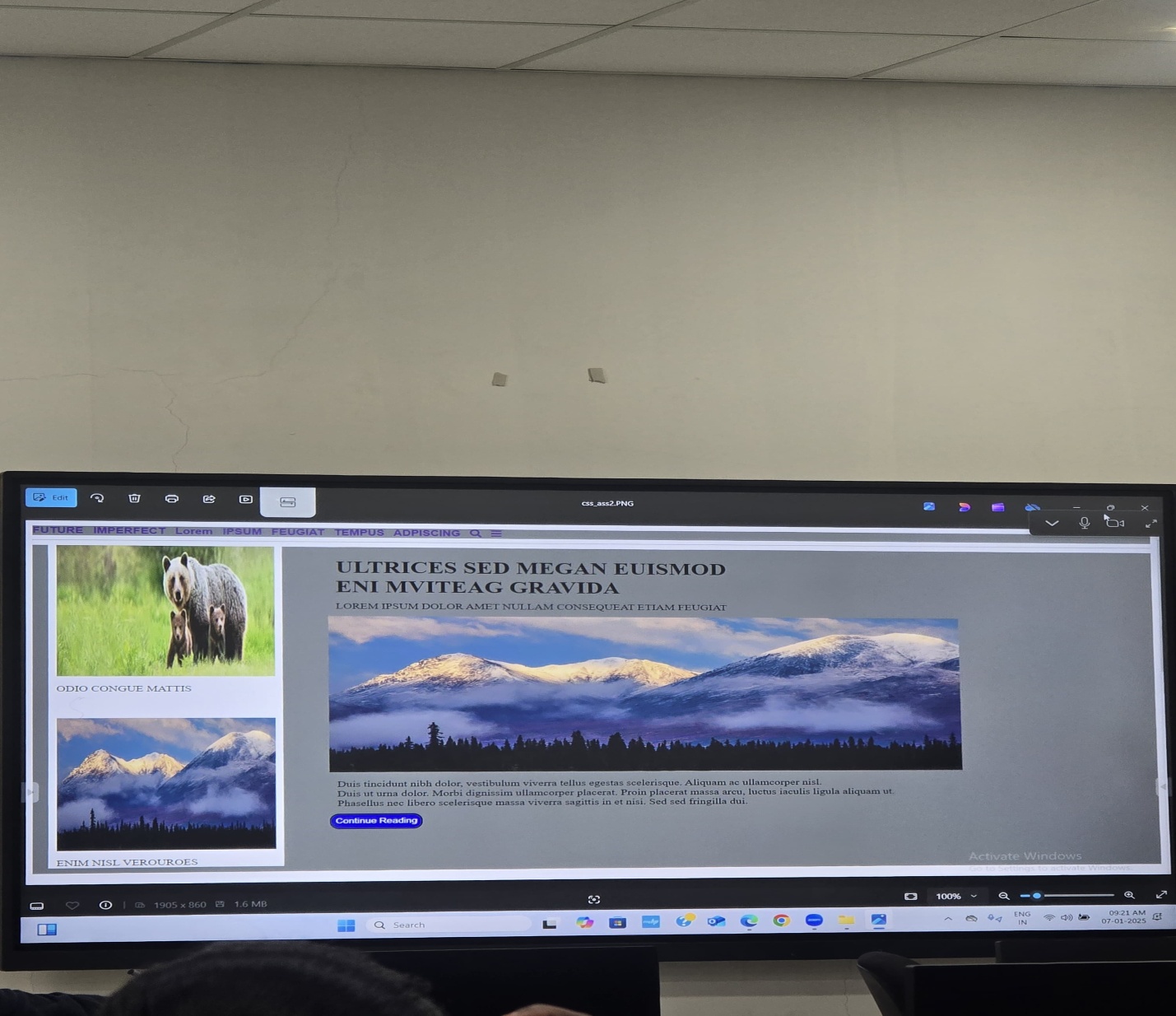
* global scope – declare the variables outside the method and access inside the method
* local scope/script scope – declare the variables inside the method but cannot access outside
* block scope – declare the variables inside {}

Ctrl shift j – console in browser

Break point

Debugging javascript

Website using html and css



**Operators**

* It is a pre-defined symbol used to perform specific operation
* Arithmetic Operators:

|  |  |
| --- | --- |
| Operator | Description |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| \*\* | Exponentiation ([ES2016](https://www.w3schools.com/js/js_2016.asp)) |
| / | Division |
| % | Modulus (Division Remainder) |
| ++ | Increment |
| -- | Decrement |

**Increment:**

* Pre increment
* Post increment

**Assignment operator(short hand)**

* +=, -=, \*=, /=, %=
* Eg: var c=10

console.log(c) //10

c=c+10 🡪c+=10

**Logical operator**

* AND(&&)
* OR(||)
* NOT(!)

**Relational operator**

* Gives output in Boolean format
* <, <=, <, >=, ==, ===, !=
* Check values not the datatype 🡪 ==
* Checks both datatype and value 🡪 === (strictly equals to)
* Console.log(5==5) //true
* Console.log(5==’5’)//true
* Console.log(5===5) //true
* Console.log(5===”5”)//false

**Ternary operator**

* It will accept three operands
* Syntax: (condition) ? true-statement : false-statement
* Similar to if-else

**Conditional statements**

* They are used to make some decision based on some condition
* **Simple if** (only have true statement)

If(s<=10)

{

Console.log(“hello world”)

}

* **If-else** (have true and false statement)

If(s<=10)

{

Console.log(“hello world”)

}else{

Console.log(“bye”)

}

* Else-if-else / else-if ladder

Let a=20

Let b=30

Let c=40

If(a>=b && a>=c)

* Nested if

Let user=”admin”

Let password=12345

If(user===”admin”)

{

Console.log(“uservalid”)

If (password===12345)

{

Console.log(“password valid”)

Console.log(“login successful”)

}else{

Console.log(“password invalid”)

}

}else{

Console.log(“login unsucessfull”)

}

* Switch case in js

let n=1

switch(n)

{

case 1:

{

console.log("print i")

}

break;

case 2:

{

console.log("print 2")

}

break;

default:

{

console.log("invalid")

}

}

* If the case is monday and Tuesday print football
* If the case is Wednesday and tusday print cricket
* If the case is Friday print volleyball
* If the case is Saturday print basketball
* If the case is Sunday print holiday
* The input should be given by the end user

let day = prompt("Enter the day of the week:").toLowerCase();

switch (day) {

case "monday":

case "tuesday":

console.log("Football");

break;

case "wednesday":

case "thursday":

console.log("Cricket");

break;

case "friday":

console.log("Volleyball");

break;

case "saturday":

console.log("Basketball");

break;

case "sunday":

console.log("Holiday");

break;

default:

console.log("Invalid input. Please enter a valid day of the week.");

}

**looping statement :**

* used to execute/iterate set of instructions based on conditions

**while loop**

* it will iterate set of instructions until the condition become false
* by default 0 iteration

**syntax**

while(condition)

{

   set of instruction

}

**Example**

let i=1;

while(i <= 5){

    console.log(i);

    i++  //output --> 1 2 3 4 5

}

**do while loop**

* it will execute first then it will check the condition
* by default it will takes one iteration

**syntax**

do(

    set of instruction

)while(condition)

**Example**

let a=5;

do{

    console.log(a);

    a--;  //output --> 5 4 3 2 1

}while(a >= 1)

**for loop**

three arguments

1)initialization

2)condition

3)increment/decrement

**syntax**

for(initialization; Condition; increment/decrement){

    set of instruction

}

**Example**

for(let i=1; i<=5; i++){

    console.log(i);   //output --> 1 2 3 4 5

}

**Functions in javaScript :**

* Functions are first citizens in java Script
* Functions are object in javaScript

Types of functions

1. Function declaration statement/Named function/Pure function
2. Function expression
3. Function programming

* Higher order function
* Call back function

1. Arrow function
2. Nested function
3. IIFE( Immediate invoking function expression)

**Function declaration statement:**

* It is a set of instruction or block of code used to perform specific task

**Syntax:**

function fun-name(parameters,......){

    set of instructions

}

fun-name(arguments,......)

|  |
| --- |
| **Return keyword**  It will stop execution of the function  **Hoisting**  We can call function before the function declaration |

function – keyword

fun-name – identifier

parameters – identifier

block{} – local scope(cannot access outside the function

arguments – data/values(data types)

Main Advantages

* Code Reuseability(write once call multiple times)

Create a function of two numbers and the input should be taken from end user