Basics of Javascript

Software’s requirement

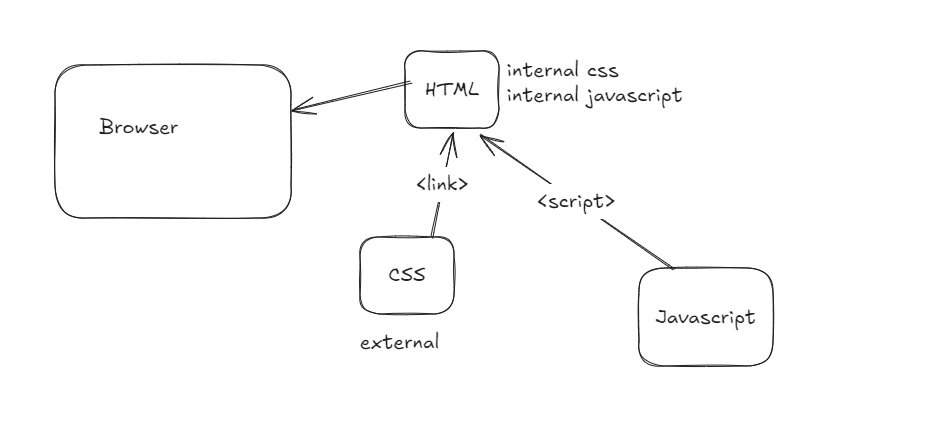
* VS Code editor
* Browser

Web technologies

* HTML: Display the content on the web page
* CSS: It is used to style the HTML elements
* Javascript: It allows you to add dynamic effects at runtime

Fundamentals of Javascript

* variables
* functions
* conditional constructs
* looping statements
* arrays
* objects
* event handling
* form validations
* DOM manipulations

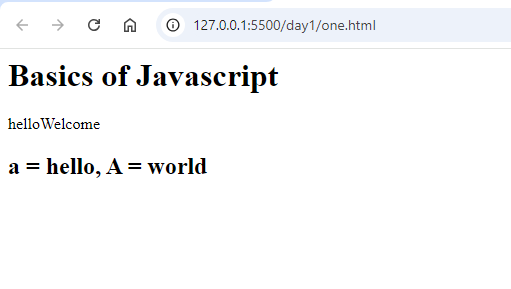


HTML is the file the browser loads, where you can have Javascript & CSS linked to the HTML or internally also you can have HTML & CSS

one.html



Output:



In Javascript you don’t have types for variables unlike other programming languages like Java, C, C++, C#, which means a variable can store any kind of value

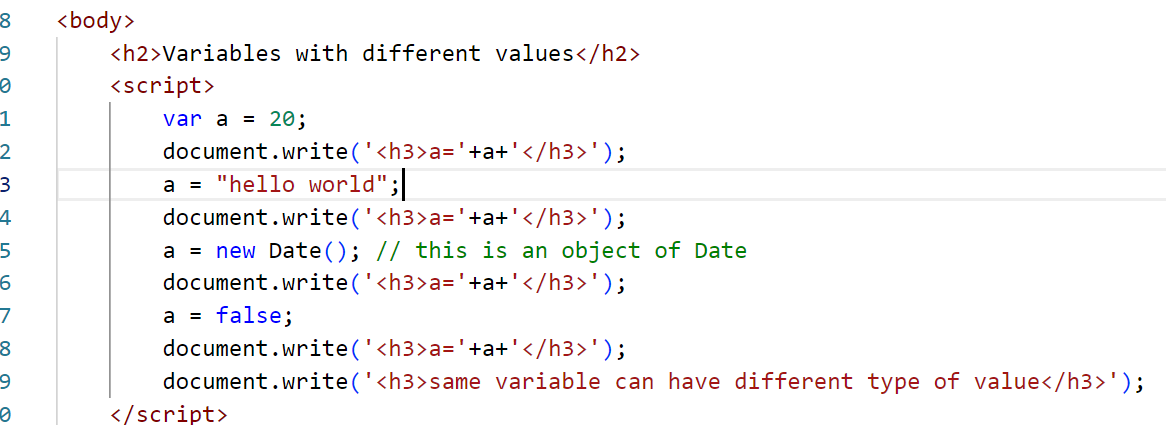
a = 10;  
a = “Hello”;  
a = true;

a = ‘x’; // it is still a string

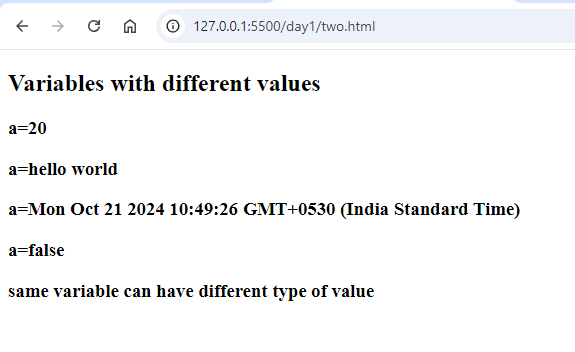
a = ‘hello world’;

a = new Date(); // a is storing a Date object

two.html



Output

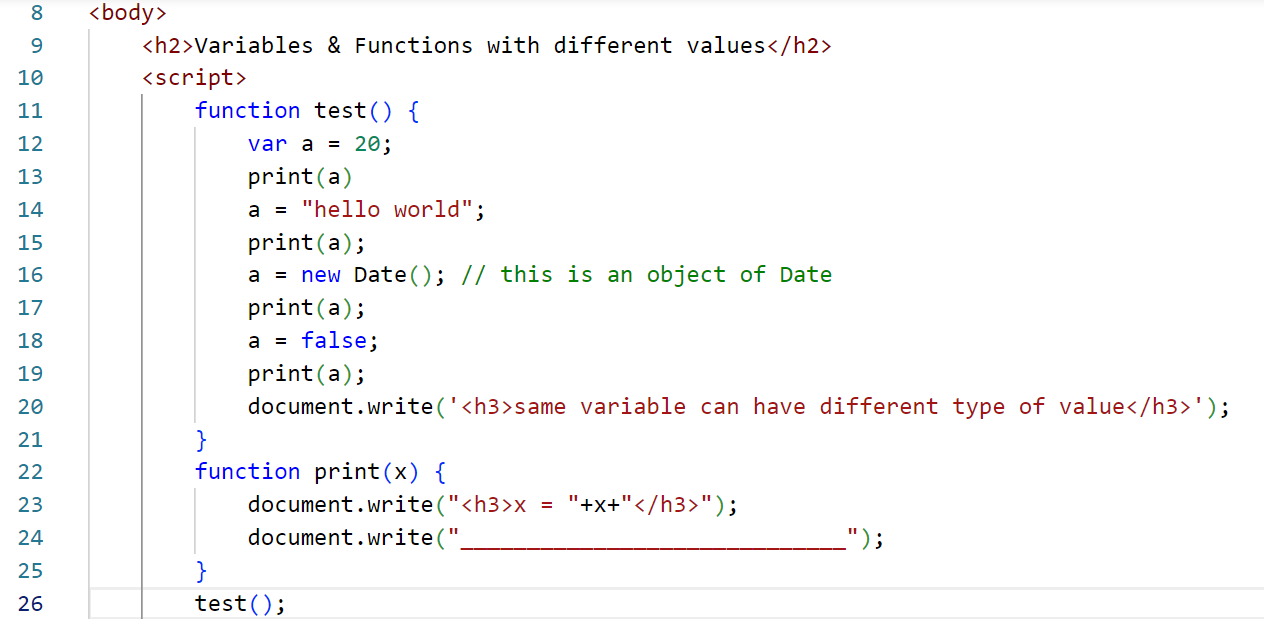


Functions: These are named blocks which are invoked by its name, they are used to create a reusable statements that can be called later

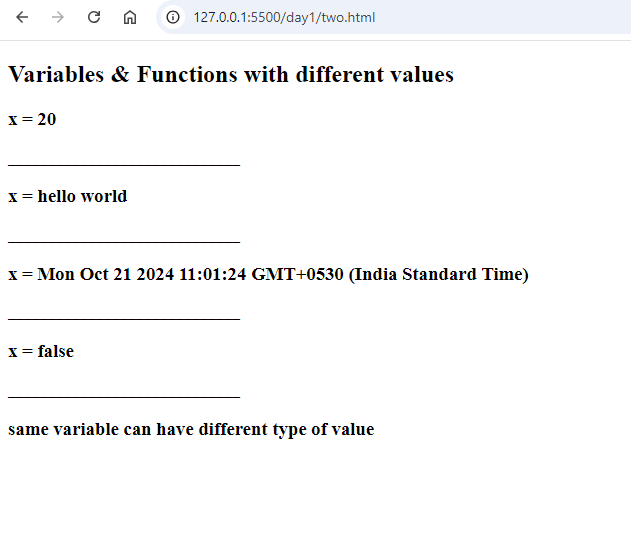
Syntax:

function function\_name(parameters, parameters) {   
 statements; // one or more lines of statements  
}  
Invoking function:

function\_name(arguments);



Output:



Predefined functions in Javascript

In Javascript there are some inbuilt objects like

* window
* document
* console
* Math
* Date
* String

window is the top level object whose function/properties you can access without using its name.

Object: It is a real world entity, it will have 2 things

* properties: variables
* functions: actions

window.alert(“creates an alert message);  
or  
alert(“creates an alert”)

window.parseInt(“”) or parseInt(“”)

window.parseFloat(“”) or parseFloat(“”)

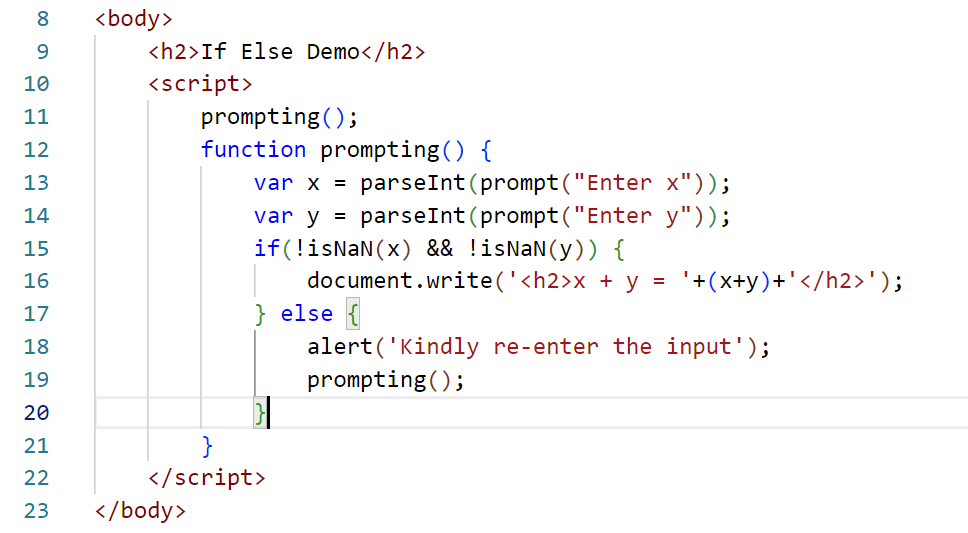
Conditional Statements

Javascript supports following conditional statements

1. if
2. if – else: if (….) { stmt; } else { stmt; }
3. if – else if else if … else
4. switch

Activity:

Using if & else, try to take 2 inputs that needs to add those inputs only if both the input’s are numbers, else prompt again to ask the input



There are two parse functions in Javascript

* parseInt
* parseFloat

parseInt(“12345.45”): It returns 12345  
parseFloat(“12345.45”): It returns 12345.45

Complex types

1. Arrays
2. Objects

Array: These are the variables that can store more than one value

Object: These are the variables that will have properties & functions

array is created using [ ] in Javascript

var items = [1, 2, 3, 4, 5];  
var names = [“Alex”, “Virat”, “Rohit”]  
object is created using { } in Javascript

var user = { userId: 134, name : “Virat”, age:38 };

Object Array is created using [{ }, { }, { } ]   
var users = [ {userId:123, name: “Virat”, age:38}, {userId: 456, name:”Rohit”, age:37}]

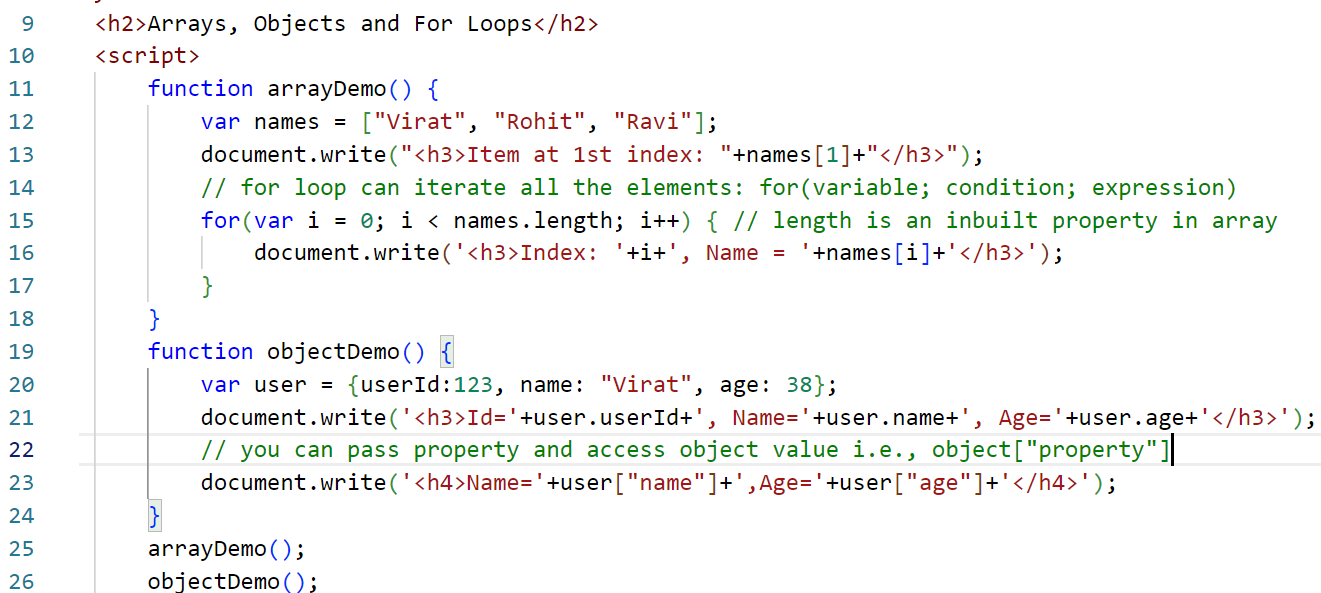
How to access an element in an array

items[0], items[1], items[2] or you can use for loop to iterate all the elements

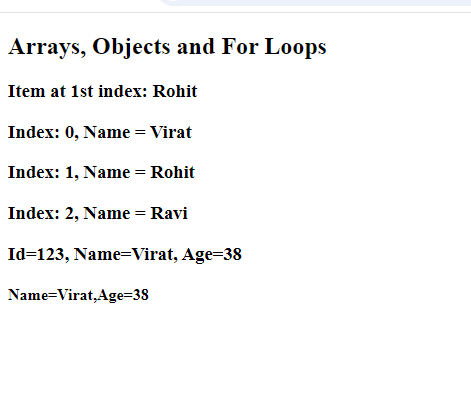
How to access object property

user.userId, user.name, user.age

users[0].userId, users[1].userId, users[2].userId ….



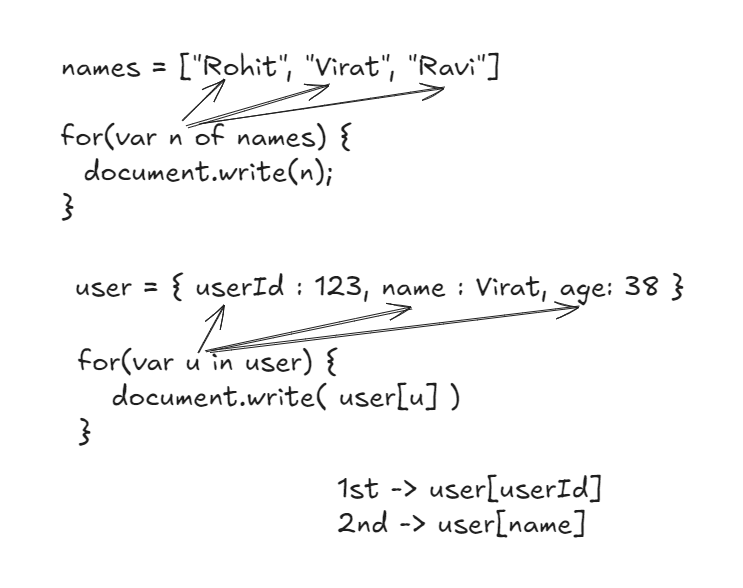
Output:



Javascript has three types of for loops

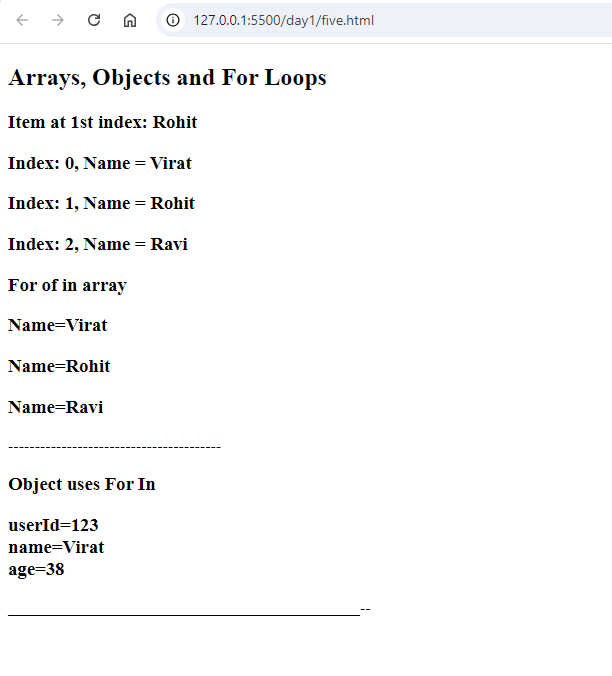
1. traditional for loop which uses index to access the array  
   i.e., for(init; condition; expr) { … }
2. for .. of : to iterate an array, but it doesn’t need index to access the element, it directly iterates the value  
   i.e., for(var x of names) { print x }
3. for.. in: to iterate an object with properties, it doesn’t need you to know what properties an object have

i.e., for(var u in user) { print user[u] }





Output:



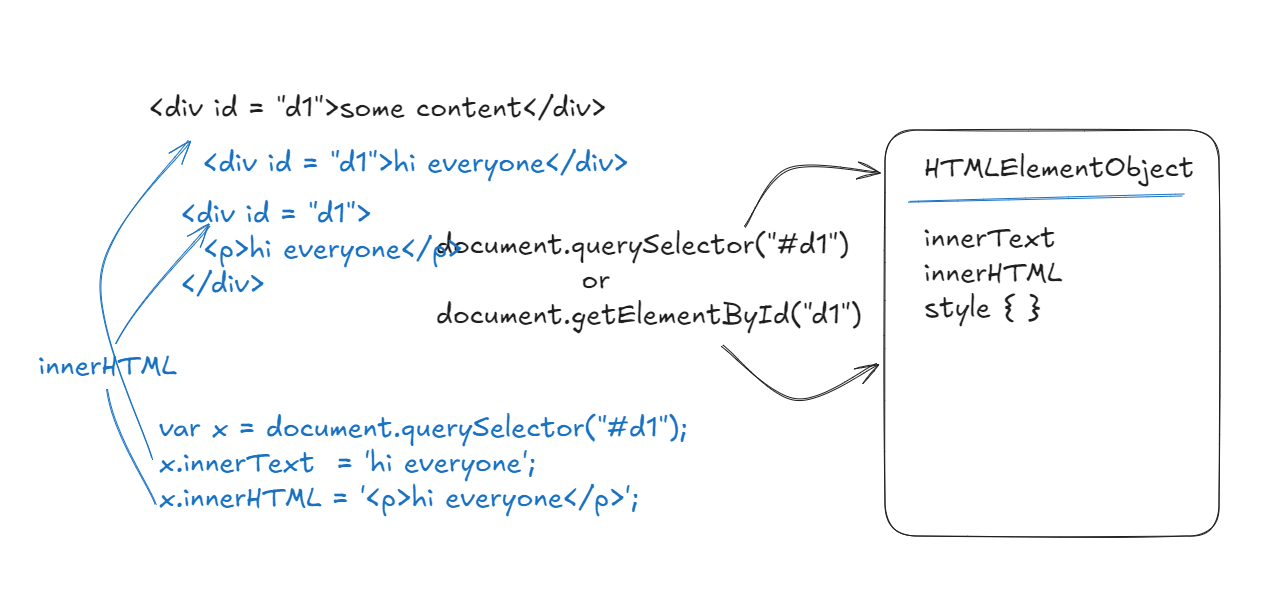
DOM manipulation:

Using Javascript you can access HTML and manipulate it like adding/removing contents/attributes, you can use some of the inbuilt functions provided by document object which are:-

* getElementById(“id”)
* getElementsByTagName(“tag”)
* getElementsByClassName(“class”)
* querySelector(“selector”): It can take tag, class and id
* querySelectorAll(“selector”): it can take tag, class & id

<div class = “c1”>hello</div>   
<div class = “c2”>thankyou</div>  
<p id = “p1” class = “c2”>hello p1</p>   
<p id = “p2” class = “c1”>hello p2</p>

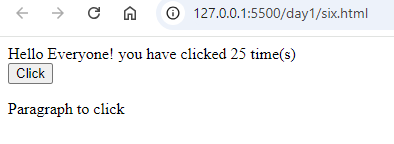
document.getElementById(“p1”) ; // <p id = “p1” class = “c2”>hello p1</p>  
document.getElementsByTagName(“p”); // [ <p>, <p> ]  
document.getElementsByClassName(“c1”); // [ <div.c1>, <p.c1> ]  
document.querySelector(“.c1”); // first element using the class c1 i.e., <div class=”c1”>  
document.querySelector(“#p1”); // <p id = “p1”>  
document.querySelector(“p”); // first occurrence of <p> i..e, <p id = “p1” class=”c2”>hello p1</p>  
document.querySelectorAll(“.c1”) or document.querySelectorAll(“div”)



x.style.color = ‘red’; >> <div id = “d1” style=”color:red”>



Output:





Above code at 28th line it is not calling handleClick, it uses handleClick as an handler that is executed when you place mouse over the element #btn2

event object:

This is an object that is supplied as a parameter to the handlers

btn2.addEventListener(“click”, function(ev) {   
 // ev is an event object that generated the click event  
});

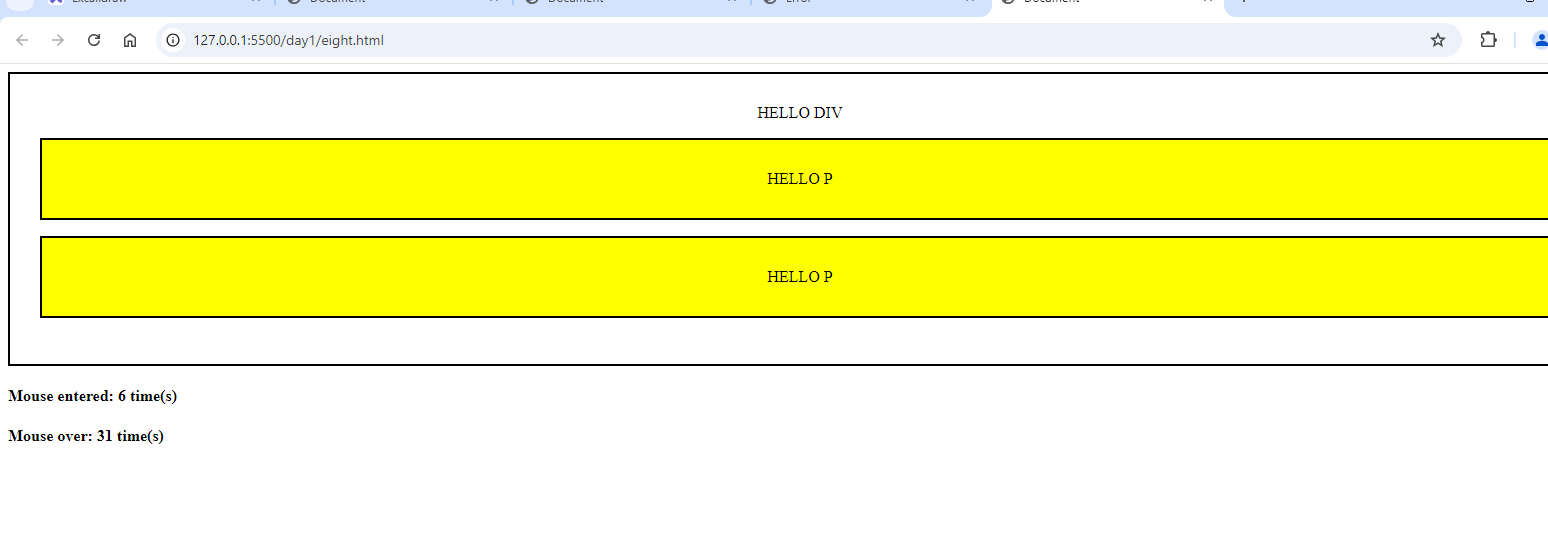
mouseover vs mouseenter: mouseover occurs even for the child elements, whereas mouseenter occurs only to the parent element using mouseenter

mouseout vs mouseleave: mouseout occurs event for the child elements, whereas mouseleave occurs only to the parent element using mouseleave

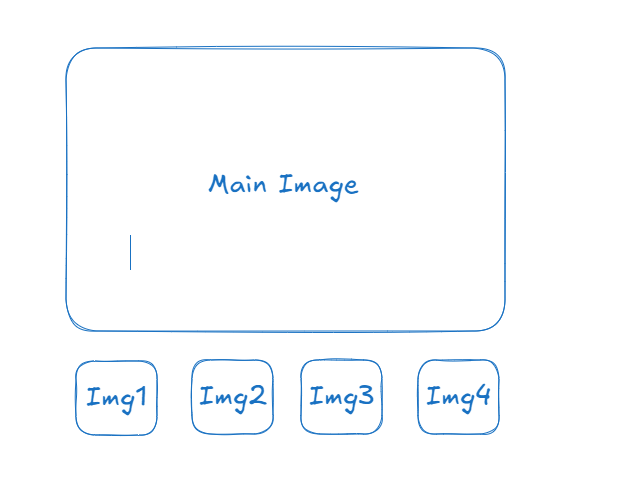
<div onMouseOver = “fn1()” onMousenter=”fn2()”>  
 <h2>..</h2>  
 <p>..</p>  
</div>



Output:



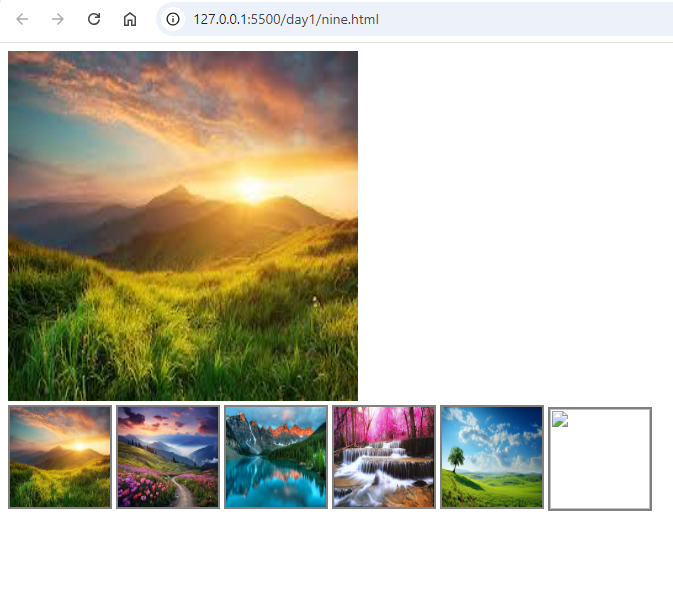
Using event handling to display the image that you click in the thumbnail, when you click on the thumbnail image, the main image will show the image you clicked on the thumbnail



Use <img> tag with src, width & height attributed i.e.,   
<img src = “ImageURL:GoogleImage” width = “300” height = “300” id = “main”>



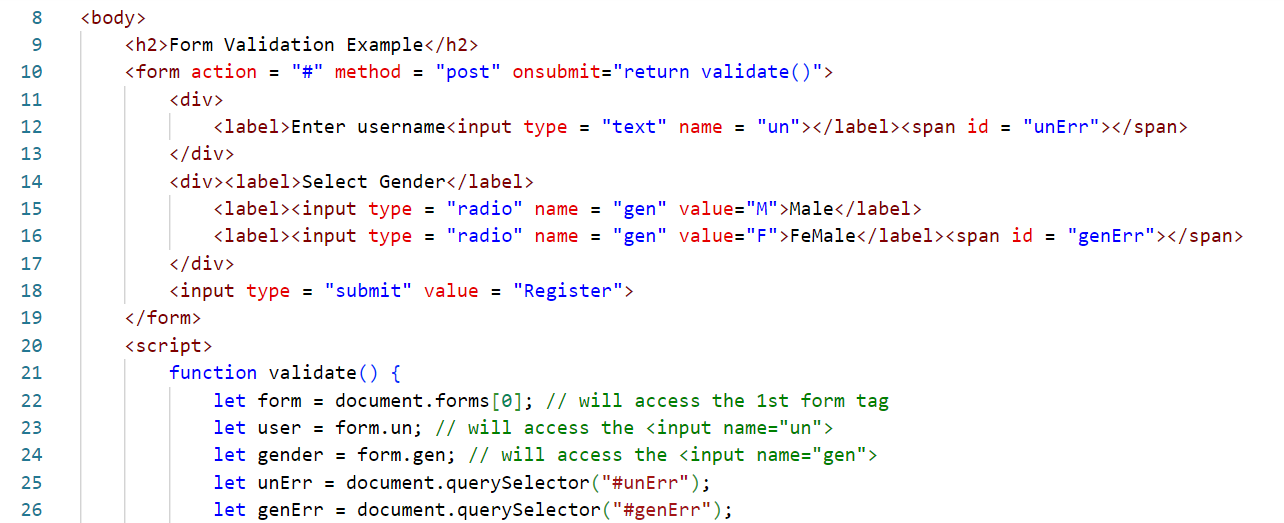
Output:

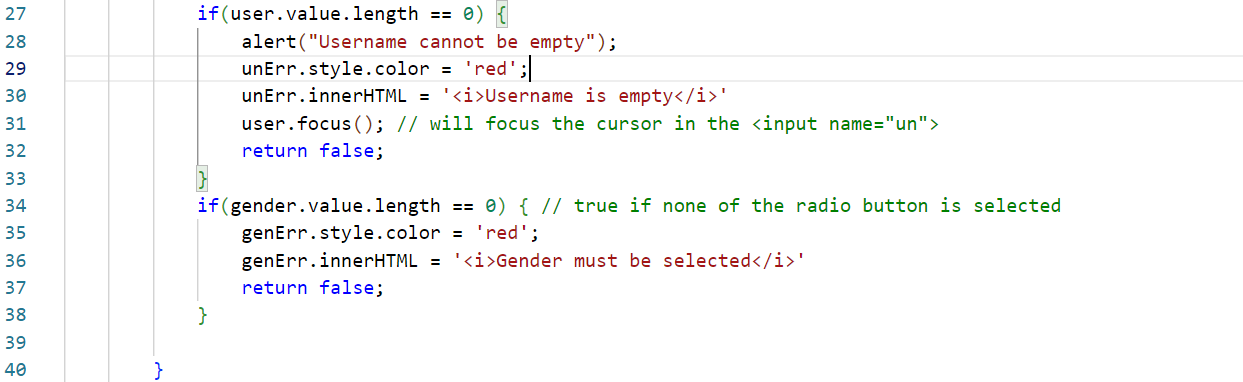


Form Validations in Javascript

These are client side validations that can restrict client to submit invalid inputs like

password & confirm password must be same, an account number / amount must not allow alphabet



Output:

