**List of Technologies**

* HTML
* CSS
* Javascript
* JQuery
* Maven & Spring Framework
* Spring Boot
* Kafka

HTML: Hyper Text Markup Language

CSS: Cascading Style Sheet

Web development, there are 3 technologies are used

1. HTML: It is to display the content
2. CSS: It is to style the HTML
3. Javascript: It is to add effects dynamically and add HTML / CSS at runtime

Editors

1. VS Code
2. Eclipse

HTML

Heading tags: h1, h2, h3, h4, h5, h6

<h1>some content</h1>  
<h2>some content</h2>

Paragraphs: These contents are used to display the paragraphs, which gives a line break before & after the paragraph

bold & italic tags: <b> & <i>, these tags doesn’t have line breaks, these are used inline

<p>Some content <b> few more contents </b> again some more </p>

Self closing tags: These tags don’t have closing, they can be closed with the start tag itself

<hr />: Horizontal rule  
<br />: This is to give a line break  
<img />: This to display the image

Attributes: These are some extra information’s given to the tag, they appear in the start tag

ex: <img src=”url” width=”200” height=”200” />

Marquee: It makes content to move

Entities: These are some special characters which can be displayed using some entity names or numbers

&copy; this creates a copyright symbol  
&reg; this is to create registered mark  
t; for < symbol  
&gt; for > symbol

&#8377; creates rupee symbol

Links & Lists

anchor tag is used to create the link <a href=”url”>LinkName</a>  
<a href=”url” target=”\_blank”>LinkName</a>

Lists: There are 2 types

1. ordered <ol>
2. unordered <ul>

<li> it is the list index or list item that comes inside the ordered/unordered lists

Tables: <table> tag is used to create tables, it will have 2 child tags <thead> & <tbody> to create table headings & table contents respectively, both uses <tr> to create rows, to create columns we have <th> & <td>, <th> is used for heading purpose, <td> is used in the table body.

div: It is a container tag that can wrap other HTML tags, this is mainly to group the HTML tags & style them later

Forms: These allow to create forms that user can input, like text, checkbox, radio buttons, drop downs, buttons, password, text area, range bar, progress bar, file upload

Form comes with 2 attributes

1. action: Carries the server URL where the data needs to be submitted
2. method: This specifies the HTTP methods like GET / POST, which tells whether form data must be submitted through URL (GET) or through the body (POST)

Note: Default form submission is GET i.e., sends data through URL, hence its advisable to specify method=”post”

Usage:

<form action=”#” method=”post”>  
 <label>Enter Name</label>  
 <input type = “text” name = “username” />  
 <label>Enter Password</label>  
 <input type=”password” name=”secret” />  
 <!-- for drop down -->  
 <select name=”qualification”>   
 <option value=”btech”>BTech</option>  
 <option value=”mtech”>MTech</option>  
 </select>  
 </form>

CSS: Cascading Style Sheet, you can apply CSS in 3 ways

1. Inline CSS: applied on a particular element using style attribute
2. Internal CSS: applied to the whole HTML document, using <style> tag
3. External CSS: applied to multiple HTML documents by creating a CSS file, CSS file needs to be linked using <link> tag

Syntax: property : value; property : value;

CSS Selectors: These help in selecting the HTML elements, there are mainly 3 types of selectors

1. tag selector: to select the element using tag name

p { color : blue }  
h2 { color : red }

1. id selector: to uniquely select an element using the element id

#a { color : blue }  
#b { color : red }  
<p id = “a”> & <h2 id = “b” >

1. class selector: to select multiple elements using the class name

.i1 { color : blue }   
.i2 { color : red }  
<p class = “i1”> & <h2 class = “i1”>

3rd party CSS: These are predefined CSS provided by CSS communities that can be used anywhere in the world

ex: bootstrap, semantic-ui, angular-materials, primefaces, tailwind

Bootstrap common class-names

Button: btn btn-primary, btn-secondary, btn-success, btn-danger,

Forms: form-control, form-control-lg

div: container-fluid, container

Tables: table, table-striped

Activity: Align the Form to the center of the page

Javascript: It adds dynamic effects to your web page by accessing HTML & CSS at runtime, apart from modifying HTML & CSS it can also perform following operations

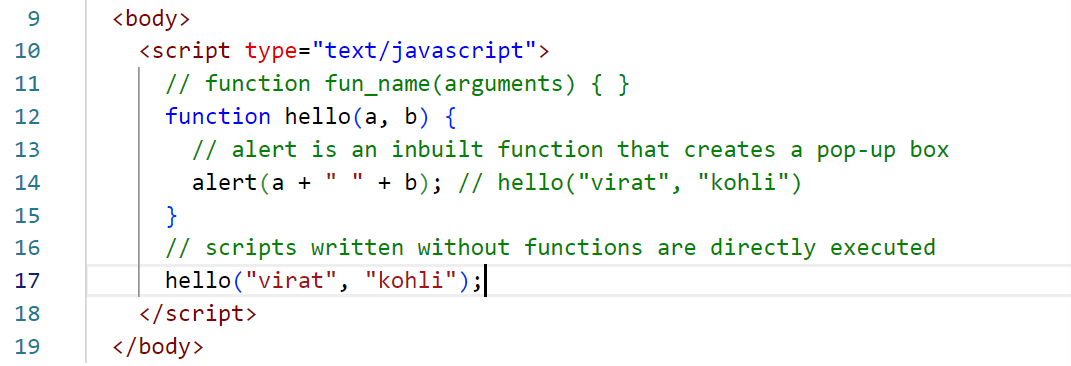
1. Validations
2. Event Handling

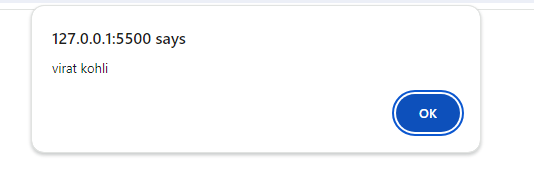
<script> tag is used to add javascript to your html

Fundamentals of Javascript

1. Variables
2. Functions
3. Operators
4. Conditional Statements
5. Arrays
6. Loops
7. Objects
8. Events

Our first JS program



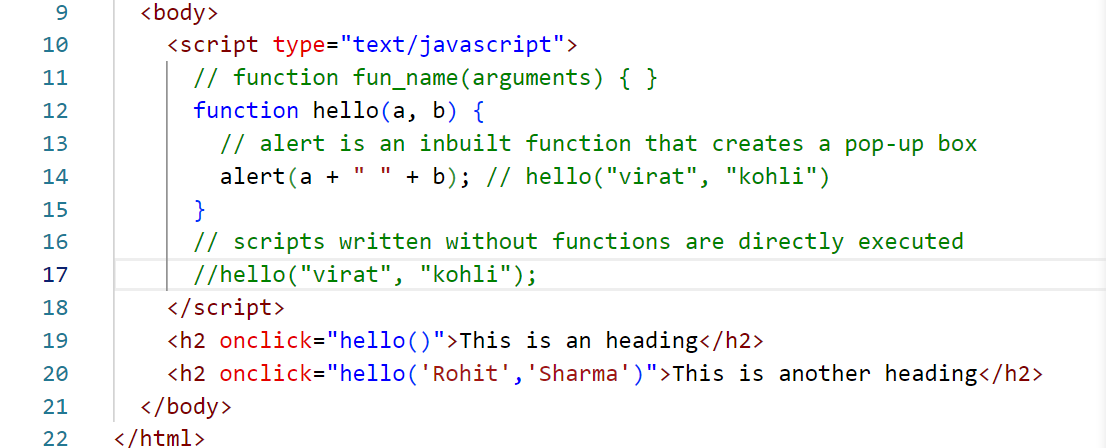
Output  


How to generate events & handle these events

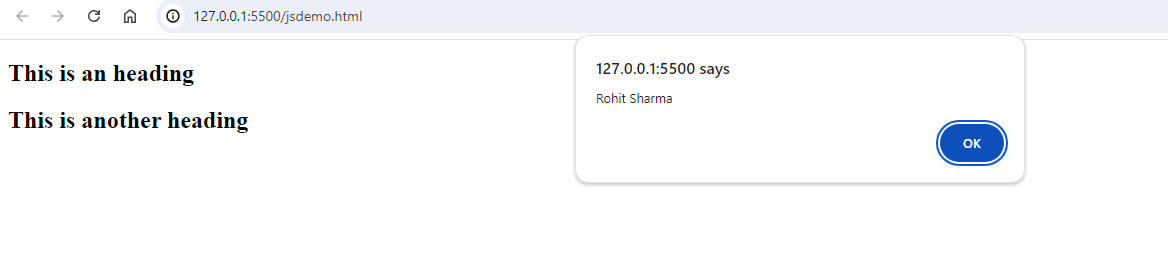
Events are anything that happens on browser, it could be click, mouse over, mouse out, form submit, input focus, input blur, all these events can occur on any HTML element

<tagName onclick=”fn()”>Some content</tag>

Program that generates click event on an HTML element



Output:



Arrays: Collection of elements, in Javascript arrays can have elements with different datatypes

var items = [ 20, 10, 30, 50, 40];

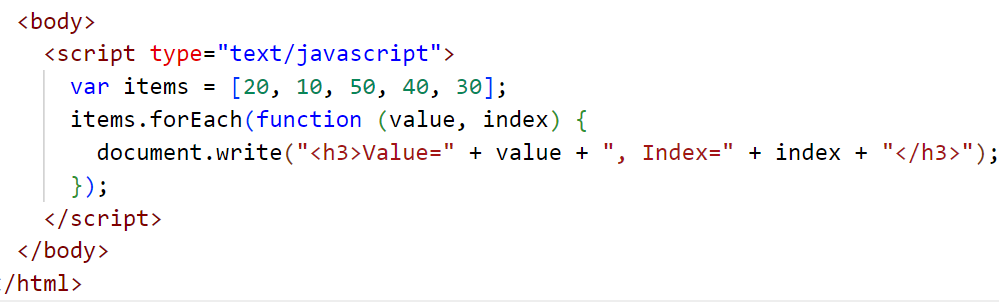
for(var index=0; index < items.length; index++) {   
 document.write(items[index]); // print the content in the browser  
}

Another way to iterate is using forEach() function of an array

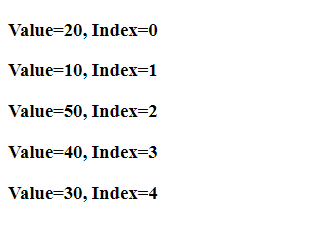
items.forEach( callbackFn );

items.forEach( function(value, index) {  
 document.write(value+ “ “ +index);  
 } );

Arrays iteration with forEach



Output:



Keywords used to declare variables in javascript

1. var (avoid using var): Because it doesn’t have any scope
2. let: It has scope & its value can be modified
3. const: It has scope & its value can’t be modified

Limitations of var

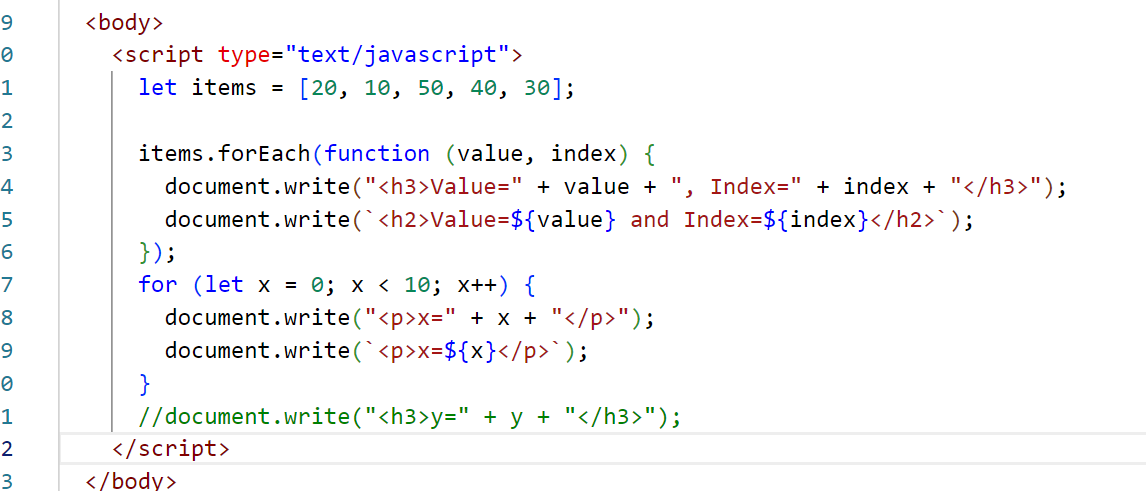
1. It doesn’t have a scope
2. It can be re-declared which is not correct

Javascript is derived from EcmaScript (ES), in ES6 they have made lot of changes that improved the Javascript syntax, those changes are:-

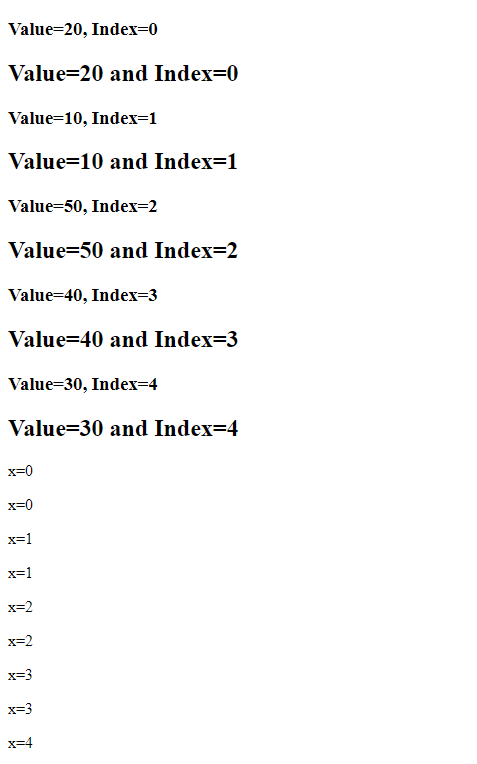
1. let, const keywords
2. class, super, extends keywords
3. template strings
4. arrow functions

Template String: This avoids lot of string concatenation you do with +, it uses back tick(`) character to include strings & javascript expressions without breaking the string.

let x = ‘<p>Hello ’+username+’</p>’;  
let y = `<p>Hello ${username}</p>`



Output:

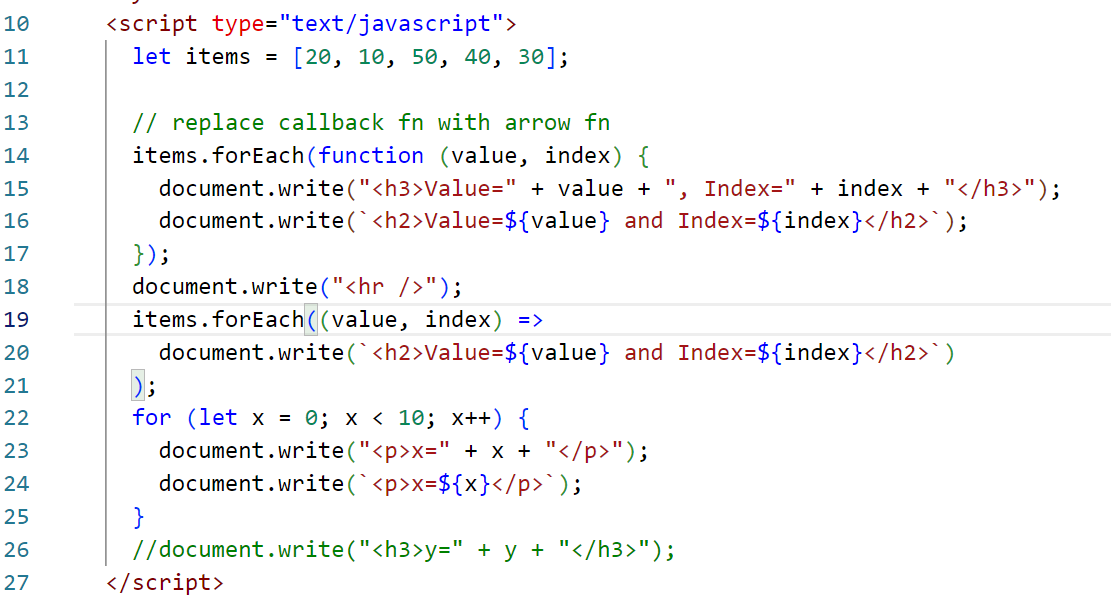


Arrow function:

It is a replacement for the callback functions, to simplify the syntax

|  |  |
| --- | --- |
| Callback Function | Arrow Function |
| function(x, y) {   console.log(x, y); } | (x, y) => console.log(x, y)  Note: If the body has only one line code, then { } is optional |
| function(x, y) {  return (x + y); } | (x, y) => x + y;  Note: By default arrow function returns the expression without return keyword |
| function(x, y) {   console.log(x, y);  return (x + y); } | Note: If arrow function uses { }, then it must use return keyword if necessary  (x, y) => {  console.log(x, y);  return x + y; } |
| items.forEach(function(x, y) { } ); | items.forEach((x, y) => { .. } ); |

Arrow function demo



Accessing & Manipulating the DOM

DOM stands for Document Object Model, it is a tree like structure of the HTML elements loaded in the browser, it can be accessed using document object which provides methods like

* getElementById(“id”): Returns the element having a matching id
* getElementsByTagName(“tag”): Returns an array of elements matching the name
* querySelector(“selector”): Returns the first element matching to the selector, here selector can be id, class or tag name
* querySelectorAll(“selector”)

<p id = “p1”>  
<p id = “p2”>  
<p class=”c1”>  
<p class=”c1”>

document.getElementById(“p1”): returns <p id=”p1”>  
document.querySelector(“#p1”): returns <p id = “p1”>  
document.querySelector(“.c1”): returns <p class=”c1”> first p.c1 is returned  
document.querySelectorAll(“.c1”): returns an array of [p.c1, p.c1]  
document.querySelectorAll(“p”): returns all the <p> elements  
document.getElementsByTagName(“p”): returns all the <p> elements

How to add Styles to the element in Javascript  
You need to add style attribute to the element at runtime

|  |  |
| --- | --- |
| HTML/CSS | Javascript |
| <p style=”color:red;background-color:yellow”> | let element = document.querySelector(“p”);  element.style.color=’red’; element.style.backgroundColor=’yellow’ |

Accessing the elements using event object

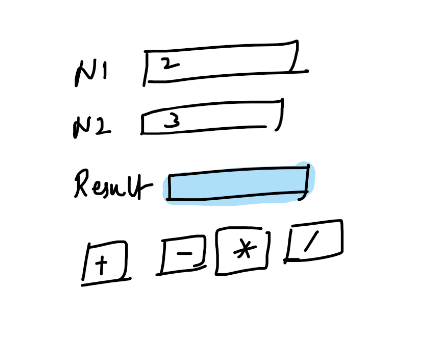
event: It is an object that is generated by an element, when an element generates the event, you can use event.target to know which element generated the event.

<p onclick=”hello(event)”>Hello World</p>  
hello( event ) { let ele = event.target }

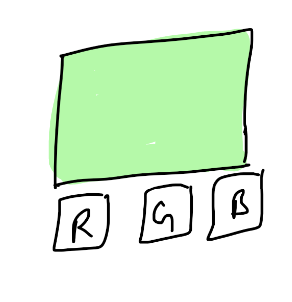
<div onclick=”hello(event)”>  
 <p>hello</p>  
 <h2>Welcome</h2>  
 <img src=”…”>  
</div>

Activity:

1. Create 3 textbox where you can enter numbers on 2 text box and the 3rd text box will display the result, create 4 buttons like add, subtract, multiply & division



1. Create 3 buttons with names like RED, GREEN, BLUE, on clicking these buttons add a background-color



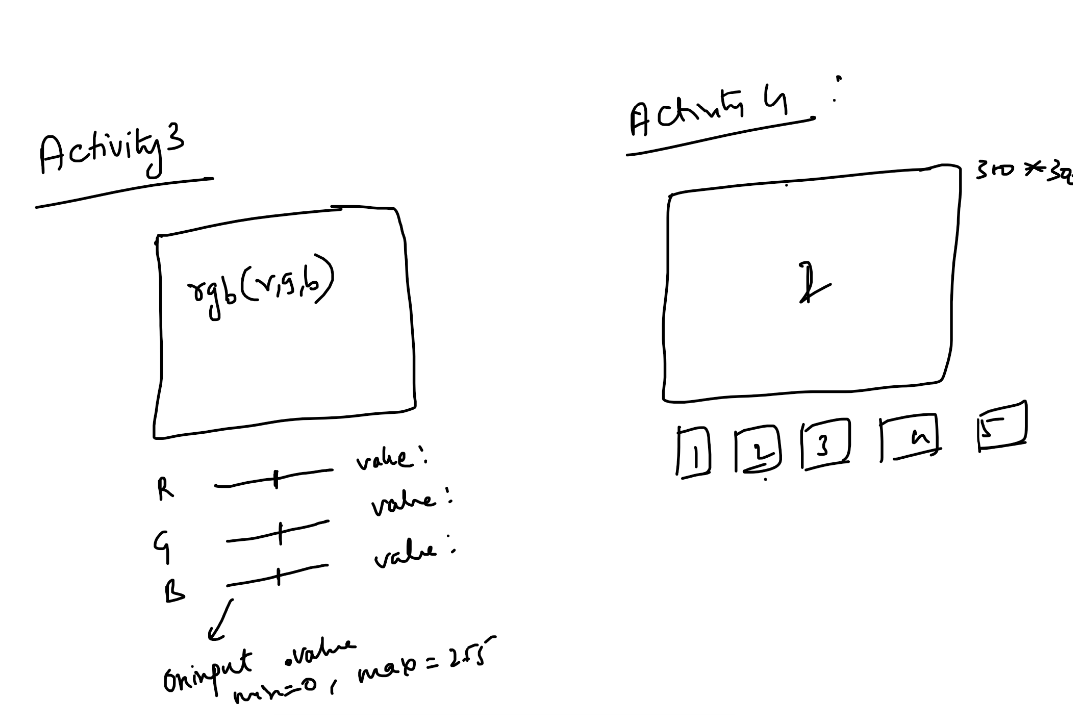
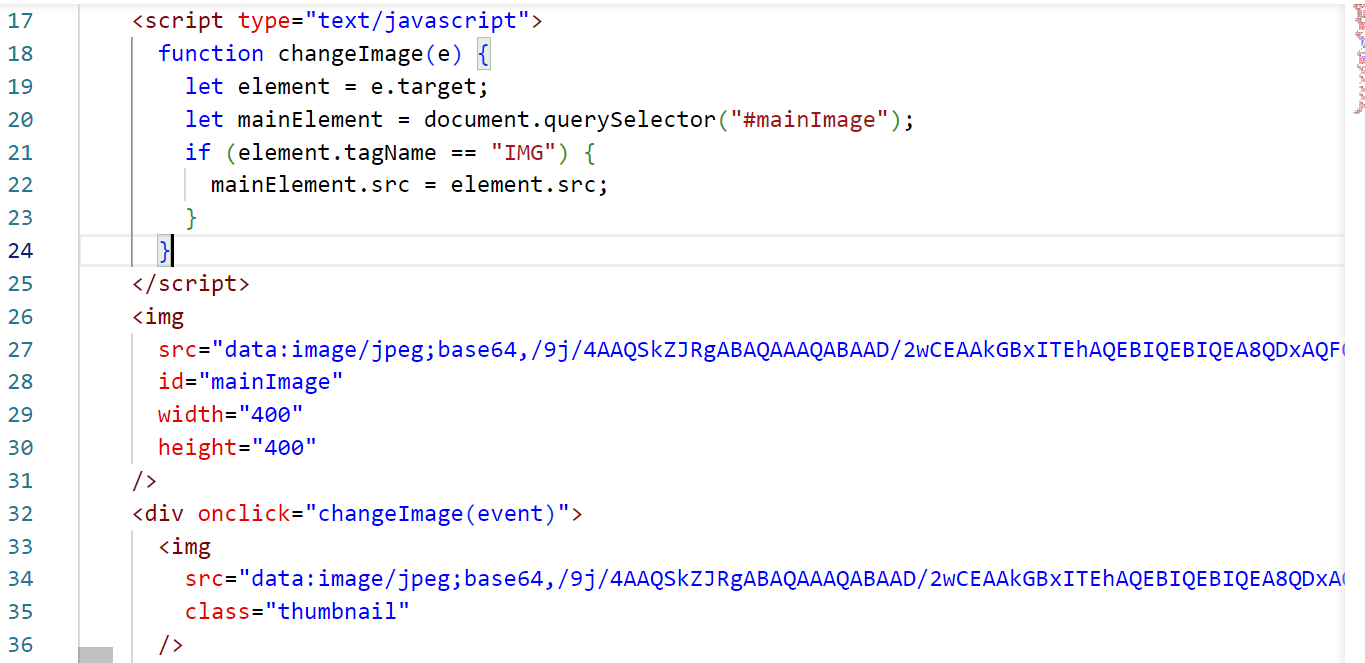


Image Changing Activity Solution





JQuery: It is a javascript library used to quickly manipulate DOM, it provides many inbuilt functions which are written in javascript

Javascript:

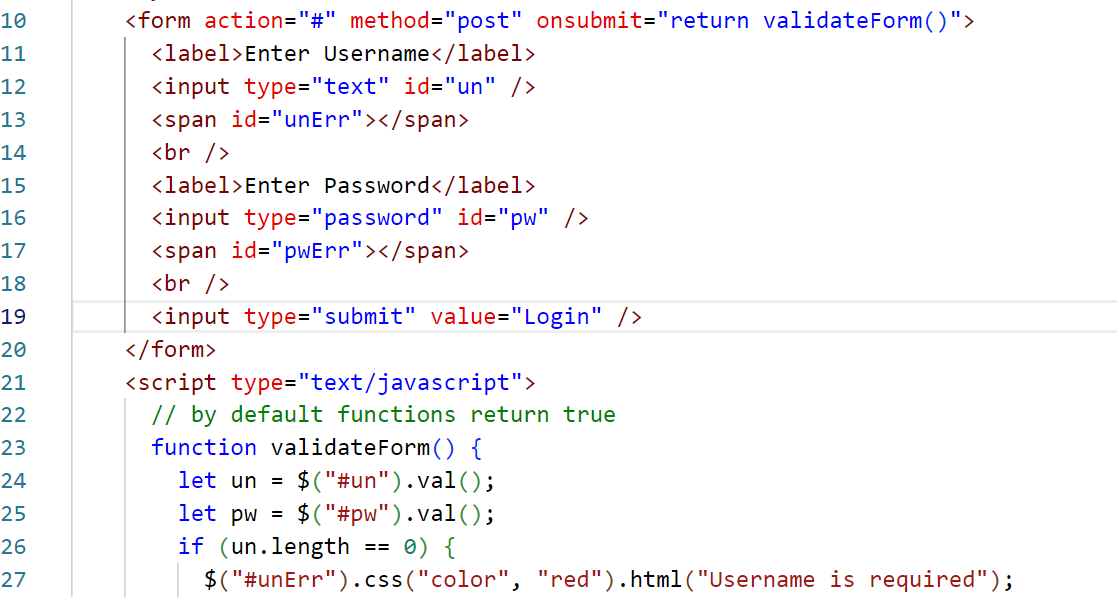
let element = document.querySelector(“#i1”);  
element.innerHTML = ‘<p>Some Content</p>’;  
element.style.color=’red’;  
element.style.backgroundColor=’yellow’;  
<p onclick=”test()”>…</p>  
<p onclick=”test()”>…</p>

JQuery:

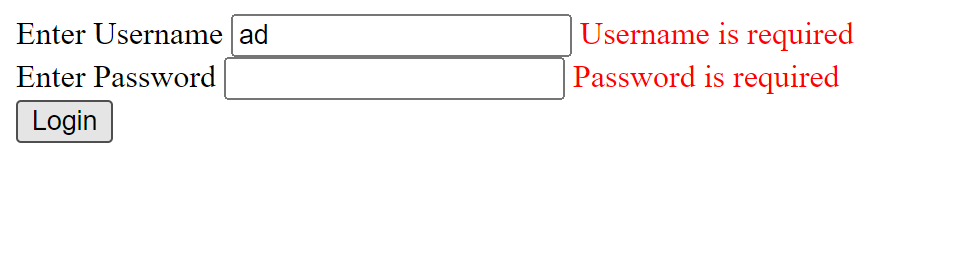
let element = $(“#i1”);  
element.html(‘<p>some content</p>’);  
element.css(‘color’, ‘red’);  
element.css(‘background-color’,’yellow’);  
$(“p”).click(function(){…});

Form Validations

Whenever user inputs invalid data we need to stop submitting the form, this can be done through validation.



Output:



Activity:

In the above form the error message is not disappearing when the username or password has value, that needs to be implemented.