



# Tutorial 4: JavaScript

CS 104

Spring, 2024-25

TA: Abhi Jain

Credits: TA-2023-2: Guramrit  
Singh



# Topics

- JavaScript (Basics)
- Fun Activities

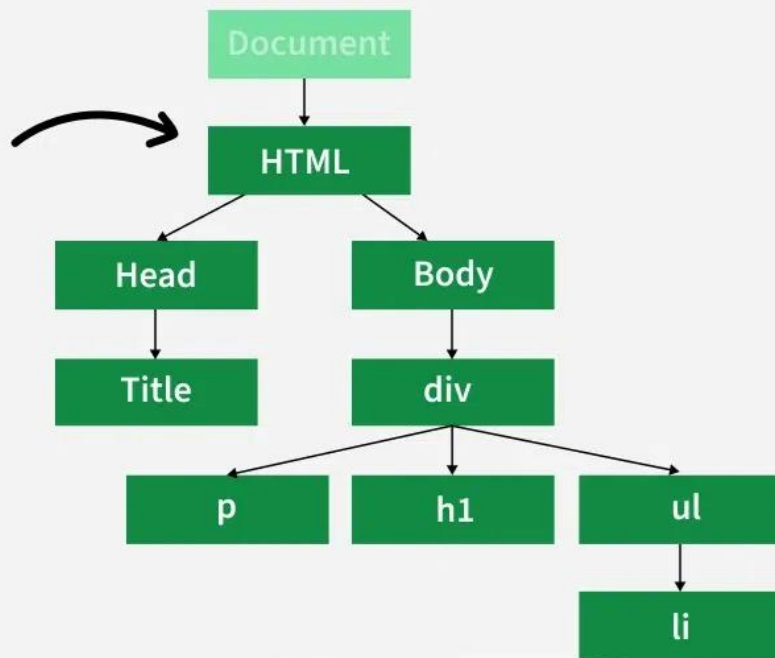


# JavaScript (Basics)

- Functions
- Iterables
- Events

```
<html>
<head>
  <title> Webpage Title</title>
</head>
<body>
  <div>
    <h1>This is Heading Tag</h1>
    <p>Some Text Content</p>
    <ul>
      <li> List Item</li>
    </ul>
  </div>
</body>
</html>
```

## The “DOM Tree”



# Functions

slides > functions.html > html > body > br

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Function</title>
7 </head>
8 <body>
9   <h1>Square Root Calculator</h1>
10
11   <label for="number">Enter a positive number: </label>
12   <input type="number" name="number" id="number">
13   <br>
14   <button onclick="calculateSquareRoot()">Calculate Square Root</button>
15   <p id="result"></p>
16
17   <script>
18     function calculateSquareRoot() {
19       var number = document.getElementById("number").value;
20       var squareRoot = Math.sqrt(number);
21       document.getElementById("result").innerHTML =
22         "The square root of " + number + " is " + squareRoot;
23     }
24   </script>
25 </body>
26 </html>
```

← → ↺ http://127.0.0.1:3000/slides/functions.html

## Square Root Calculator

Enter a positive number:

The square root of 25 is 5

- ❖ The `getElementById()` method is the DOM method that returns the element with the specified ID.
  - ❖ The `innerHTML` property is used to change the content of an HTML element.
  - ❖ The `Math.sqrt()` method is used to calculate the square root of a number.
  - ❖ The `value` property is used to get the value of an input element.
  - ❖ The `document` object represents the HTML document that is displayed in the browser.
- 
- ❖ The syntax of function is very similar to C++, except that we don't need to specify the return type of the function. The `function` keyword is used to define a function.
  - ❖ the `onclick` attribute is used to call the function when the button is clicked..

# External Javascript

slides > Functions.html > html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Function</title>
7
8 </head>
9 <body>
10   <h1>Square Root Calculator</h1>
11
12   <label for="number">Enter a positive number: </label>
13   <input type="number" name="number" id="number">
14   <br>
15   <button onclick="calculateSquareRoot()">Calculate Square Root</button>
16   <p id="result"></p>
17
18   <script src="functions.js"></script>
19 </body>
20 </html>
```

slides > JS Functions.js > calculateSquareRoot

```
1 function calculateSquareRoot() {
2   var number = document.getElementById("number").value;
3   var squareRoot = Math.sqrt(number);
4   document.getElementById("result").innerHTML =
5     "The square root of " + number + " is " + squareRoot;
6 }
```

- ❖ We do not use the `link` tag to link to external javascript files. We use the self-closing `script` tag instead.
- ❖ It may be placed in the `head` or the `body` tag. The `src` attribute is used to specify the location of the external javascript file.

# Arrays and Loops

```
slides > objects.html > html > body > script
12  <table>
13    <tr>
14      <th>Name</th>
15      <th>Grade</th>
16    </tr>
17    <tr>
18      <th class="name">John</th>
19      <th class="grade">A</th>
20    </tr>
21    <tr>
22      <th class="name">Jane</th>
23      <th class="grade">B</th>
24    </tr>
25    <tr>
26      <th class="name">Joe</th>
27      <th class="grade">C</th>
28    </tr>
29    <tr>
30      <th class="name">Jack</th>
31      <th class="grade">D</th>
32    </tr>
33  </table>
34
35  <p id="demo"></p>
36
37  <script>
38    let names = document.getElementsByClassName("name");
39    let students = "";
40    for(let i=0; i<names.length-1; i++){
41      students += names[i].innerHTML + ", ";
42    }
43    students += names[names.length-1].innerHTML;
44    students = "The students are: " + students + ".";
45    document.getElementById("demo").innerHTML = students;
46  </script>
47 </body>
48 </html>
```

← → ↺ http://127.0.0.1:3000/slides/objects.html

## Arrays and For Loops

### Table of Students and Grade

#### Name Grade

John	A
Jane	B
Joe	C
Jack	D

The students are: John, Jane, Joe, Jack.

- ❖ `getElementsByClassName()` returns an array of objects with the same class name.
- ❖ The `for` loop is used to iterate on an array of objects.

# UI Events

```
<style>
  #box1, #box2, #box3{
    width: 50px;
    height: 50px;
    background-color: blue;
    position: relative;
    left: 0;
  }
</style>
</head>
<body>
  <h1>UI Events</h1>
  <div id="box1"></div>
  <br> <br>
  <div id="box2"></div>
  <br> <br>
  <div id="box3"></div>
  <br> <br>
  <p id="demo"></p>
```

❖ Demo shown in tutorial session.

```
<script>
  let box1 = document.getElementById("box1");
  let box2 = document.getElementById("box2");
  let box3 = document.getElementById("box3");
  box1.addEventListener("mouseover", function(){
    box1.style.backgroundColor = "red";
  });

  box1.addEventListener("mouseout", function(){
    box1.style.backgroundColor = "blue";
  });

  box2.addEventListener("mousedown", function(){
    box2.style.backgroundColor = "green";
  });

  box2.addEventListener("mouseup", function(){
    box2.style.backgroundColor = "blue";
  });

  box3.addEventListener("dblclick", function(){
    box3.style.backgroundColor = "yellow";
  });

  box3.addEventListener("click", function(){
    box3.style.backgroundColor = "orange";
  });

  document.addEventListener("keydown", function(event){
    document.getElementById("demo").innerHTML = "You pressed the " + event.key + " key.";
  });

  document.addEventListener("keyup", function(event){
    document.getElementById("demo").innerHTML = "You released the " + event.key + " key.";
  });
</script>
```



# Window Events

```
slides > > windowevents.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Document</title>
7    <!-- We will discuss event handling for window/document events -->
8  </head>
9  <body>
10
11
12    <h1>Window Events</h1>
13    <p>Scroll down to see the effect of the scroll event.</p>
14    <p>Resize the browser window to see the effect of the resize event.</p>
15    <p>Click on the browser window to see the effect of the click event.</p>
16    <p>Close the browser window to see the effect of the beforeunload event.</p>
17    <p>Refresh the browser window to see the effect of the unload event.</p>
18
19    <script>
20      window.addEventListener("load", function(){
21        alert("The page has loaded.");
22      });
23
24      window.addEventListener("scroll", function(){
25        document.body.style.backgroundColor = "red";
26      });
27
28      window.addEventListener("resize", function(){
29        document.body.style.backgroundColor = "blue";
30      });
31
32      window.addEventListener("click", function(){
33        document.body.style.backgroundColor = "green";
34      });
35    </script>
36
37  </body>
38 </html>
```

← → ↺ http://127.0.0.1:3000/slides/windowevents.html

## Window Events

Scroll down to see the effect of the scroll event.

Resize the browser window to see the effect of the resize event.

Click on the browser window to see the effect of the click event.

Close the browser window to see the effect of the beforeunload event.

Refresh the browser window to see the effect of the unload event.

Window events occur on the global window object and help track user interactions with the browser window.

### Common Window Events:

- ❖ **- Load** – Fires when the entire page has fully loaded.
- ❖ **- Scroll** – Fires when the user scrolls the page.
- ❖ **- Resize** – Fires when the browser window is resized.
- ❖ **- Click** – Fires when the user clicks anywhere on the window.

### Form Events:

- ❖ Form events will be discussed in the Fun Activities section.

# DOM Manipulation

```
<body>
  <h1 id="title">DOM Manipulation Example</h1>
  <button id="add">Add Paragraph</button>
  <button id="remove">Remove Paragraph</button>
  <button id="change">Change Title</button>
  <button id="highlight">Highlight Title</button>

  <div id="container"></div>

  <script>
    document.getElementById("add").addEventListener("click", function () {
      let newPara = document.createElement("p");
      newPara.textContent = "This is a dynamically added paragraph.";
      document.getElementById("container").appendChild(newPara);
    });

    document.getElementById("remove").addEventListener("click", function () {
      let container = document.getElementById("container");
      if (container.lastChild) {
        container.removeChild(container.lastChild);
      }
    });

    document.getElementById("change").addEventListener("click", function () {
      document.getElementById("title").textContent = "Title Changed!";
    });

    document.getElementById("highlight").addEventListener("click", function () {
      document.getElementById("title").classList.toggle("highlight");
    });
  </script>
</body>
```

## document.createElement(tagName)

- Creates a new HTML element.

## appendChild(element)

- Appends a child element to a parent.

## removeChild(element)

- Removes a child element from its parent.

## element.remove() (Modern approach)

- Removes the selected element itself.

## setAttribute(name, value)

- Sets an attribute to an element.

## classList.add() and classList.remove()

- Adds or removes CSS classes dynamically.



# Fun Activities

- Form Validation
- Submission Handling
- Your Own Carousel

# Activity 1 - Form Validation

- ❖ In this activity, we will validate the entries in the form to be in a specific format(or in other words, obey a pattern or a regular expression)
- ❖ When the form is submitted, the verifier is invoked which checks for correctness of the entries.
- ❖ Remove all the required attributes we added to make the section compulsory. We instead do this verification in the `submit_verifier` itself.
- ❖ The following functions may be of help:
  - `.reset()` for form resetting
  - `alert()` for popping up an alert window.

Solution: Provided as [index\\_validate.html](#)

Honorable Students of Computer Science

S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rathi	2026	Graph Theory, Combinatorics

## Add a new student

Roll Number:

Name:

Graduation Year:

Website:

Research Interests:

# Activity 1 - Hints



- ❖ We shall follow the following simple checking rules:
  - Roll Number should be of Form YYBXXXX - where Y and X are digits
  - Name should contain letters and spaces. Ensure no two consecutive spaces, and no space at the end.
  - Research Interests and GradYear fields should not be empty.
  
- ❖ You would have to write your own regular expression to verify/test the input. Refer [here](#) for ECMAScript(JavaScript) RegExp References.
- ❖ To build and test Regular Expressions, check out [this](#).

# Activity 1 - Demo

Honorable Students of Computer Science				
S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rath	2026	Graph Theory, Combinatorics

Add a new student

Roll Number: 123

Name: Enter Name

2024

2025

2026

2027

Graduation Year: 2027

Website: Enter Website

Research Interests:

Enter Research Interests

Submit

Honorable Students of Computer Science				
S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rath	2026	Graph Theory, Combinatorics

🌐 127.0.0.1:5500

Invalid Roll Number

OK

Name: Enter Name

2024

2025

2026

2027

Graduation Year: 2027

Website: Enter Website

Research Interests:

Enter Research Interests

Submit

# Activity 1 - Demo

Honorable Students of Computer Science				
S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rath	2026	Graph Theory, Combinatorics

**Add a new student**

Roll Number:

Name:

Graduation Year:

Website:

Research Interests:

Honorable Students of Computer Science				
S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rath	2026	Graph Theory, Combinatorics

**Add a new student**

Roll Number:

Name:

Graduation Year:

Website:

Research Interests:

Honorable Students of Computer Science				
S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rath	2026	Graph Theory, Combinatorics

**Add a new student**

Roll Number:

Name:

Graduation Year:

Website:

Research Interests:

1. Alert box shows that all information was correct
2. Resets the form

# Activity 2 - Submission Handling

- ❖ In this activity, we will be handling the submission of the **add-student** form created in the activity last week.
- ❖ When the form is submitted, append the student information to the end of the table and reset the form after submitting.
- ❖ Hints:
  - Disable the default form submission behavior.
  - The website input field is not mandatory.
  - Backtick character ` creates template literals which allow for multi-line strings and string interpolation.

Solution: Provided as [index\\_submit.html](#)

Honorable Students of Computer Science

S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rathi	2026	Graph Theory, Combinatorics

Add a new student

Roll Number:

Name:

Graduation Year:

Website:

Research Interests:



# Activity 2 - Demo

Honorable Students of Computer Science

S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rathi	2026	Graph Theory, Combinatorics

Add a new student

Roll Number: 210050051

Name: Krishna Sai Kusal

Graduation Year: 2024  
2025  
2026  
2027

Website: cse.iitb.ac.in/~krishnasaikusal

Research Interests:

Scalable and Secure Database Systems

Submit

Submit

Honorable Students of Computer Science

S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rathi	2026	Graph Theory, Combinatorics
5	210050051	Krishna Sai Kusal	2025	Scalable and Secure Database Systems

Add a new student

Roll Number: Enter Roll Number

Name: Enter Name

Graduation Year: 2024  
2025  
2026  
2027

Website: Enter Website

Research Interests:

Enter Research Interests

# Activity 3 - A Simple Carousel

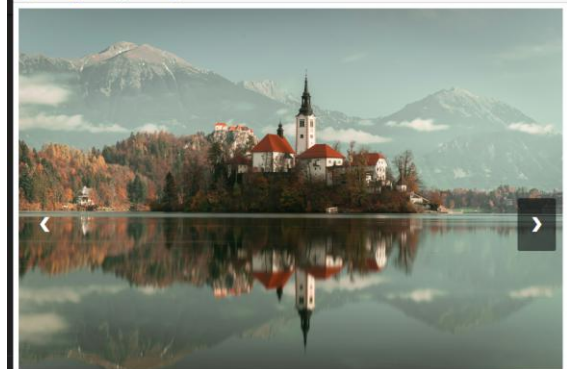
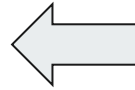
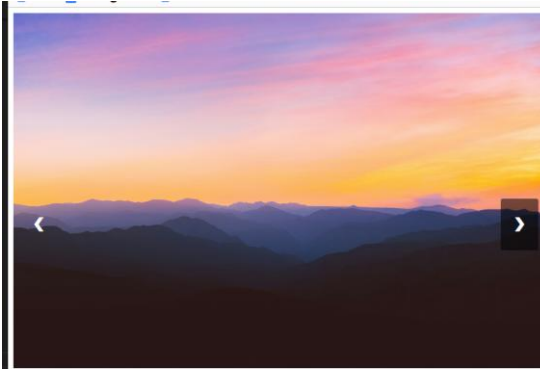
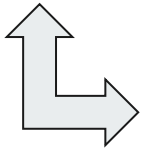
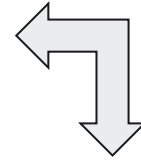
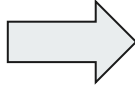
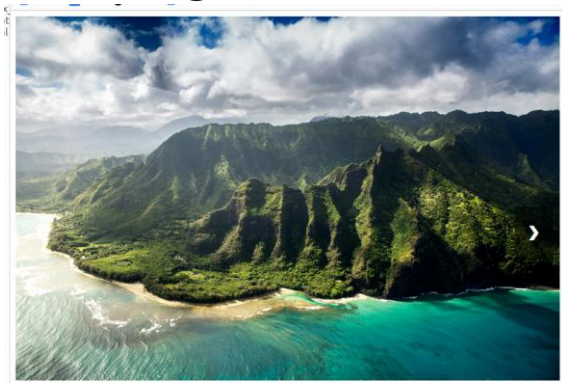


## ❖ Description:

- The carousel will have 4 images and 2 buttons to navigate through the images.
- The images will be displayed one at a time.
- The buttons will be used to navigate to the next and previous images.
- You shall ensure that there is a cyclic navigation i.e.
  - Next on the last image displays the first image
  - Previous on the first image displays the last image

- ❖ In this activity, we will build a simple carousel/slideshow.
- ❖ You can use the images provided in the resources section or use your own images.
- ❖ You can use the Unicode Decimal Code for arrows to display the next and previous buttons.
- ❖ Check out [CSS Transitions](#).
- ❖ HTML Code for Left Angle is `&#10094;` and for Right Angle is `&#10095;`
- ❖ Solution: Provided as [carousel.html](#)

# Activity 3 - Demo





**Thank You !!!**