

Module 25-Event,addEventListener, Event bubble

25-1 What is Event, different types of event in web

DOM ইভেন্ট হল একটি সংকেত যার মাধ্যমে আমরা বুঝতে পারি যে কিছু ঘটেছে, বা ঘটছে এবং এই সংকেতটি user interactions বা browser দ্বারা সংঘটিত হতে পারে।

Event সাধারণত কোন function এর মাধ্যমে কার্যকর করা হয় এবং event ঘটার আগে ফাংশনটি কার্যকর করা হবে না।(যেমন,যখন কোন user একটি button এ ক্লিক করে)।

```
<h1 onclick="console.log('Hamid')">Event niya Clickbaji</h1>
<button onclick="alert(7)">
  click Me
</button>
<button onclick="console.log(6)">
  click Me
</button>
```

25-2 Add onclick handler directly or via javascript

```
<!-- system no-2 -->
<button onclick="eventRed()">Click me red</button>

<!-- system no-3 -->
<button id="make-blue-button">Click me:Blue</button>

<script>
  // <!-- system no-2 -->

  // function eventRed()
  // {
  //   document.body.style.backgroundColor='red';
  // }

  // <!-- system no-3 -->

  // const bluebutton=document.getElementById('make-blue-
button');
  // // console.log(bluebutton);
```

```

        // // just set the name of the function
        // bluebutton.onclick=makeBlue;
        // function makeBlue()
        // {
        //     document.body.style.backgroundColor='blue';
        // }
    </script>
<script src="js/event.js"></script>

```

25-3 Different ways to use addEventListener on a button

Anonymous Function is **a function that does not have any name associated with it**. Normally we use the function keyword before the function name to define a function in JavaScript, however, in anonymous functions in JavaScript, we use only the function keyword without the function name.

```

<!-- system no-3 -->
<button id="make-blue-button">Click me:Blue</button>

<!-- system no-4 -    addEventListener-->
<button id="make-green-button">Click me:Green</button>

<!-- system no-5 -    addEventListener-->
<button id="make-orange-button">Click me:Orange</button>

<!-- system no-6 -    addEventListener-->
<button id="make-hotpink-button">Click me:hot pink</button>

<!-- system no-7 -    addEventListener-->
<button id="make-light-blue-button">Click me:Light Blue</button>

<script>
    // <!-- system no-2 -->

    // function eventRed()
    // {
    //     document.body.style.backgroundColor='red';
    // }

    // <!-- system no-3 -->

```

```

        // const bluebutton=document.getElementById('make-blue-
button');
        // // console.log(bluebutton);
        // // just set the name of the function
        // bluebutton.onclick=makeBlue;
        // function makeBlue()
        // {
        //     document.body.style.backgroundColor='blue';
        // }

        // // <!-- system no-4 -    addEventListener-->

        // const greenButton=document.getElementById('make-green-
button');
        // // anonymous function
        // greenButton.onclick=function()
        // {
        //     document.body.style.backgroundColor='green';
        // }
        // // greenButton.onclick=function makeGreen()
        // // {
        // //     document.body.style.backgroundColor='green';
        // // }

        // // <!-- system no-5 -    addEventListener-->

        // const orangeButton=document.getElementById('make-orange-
button');
        // orangeButton.addEventListener('click',makeOrange);
        // function makeOrange()
        // {
        //     document.body.style.backgroundColor='orange';
        // }

        // // <!-- system no-6 -    addEventListener-->
        // const hotPink=document.getElementById('make-hotpink-
button');
        // hotPink.addEventListener('click',function()
        // {
        //     document.body.style.backgroundColor="hotpink";
        // }

```

```

        // );

        // // <!-- system no-7 - addEventListener-->
        // document.getElementById("make-light-blue-
button").addEventListener('click',function(){
        //     document.body.style.backgroundColor='lightblue';
        // });
    </script>

```

25-4 Most common ways to set Event Handlers

Input-> `console.log(nameField.value);`

Other-> `console.log(nameField.innerText);`

<body>

<p id="clicking-method"></p>

<input id="name-field" type="text">

<button id="update-name">Update</button>

<button onclick="handleJustClick()">Just Function</button>

<button id="handle-event">direct add Event Listener</button>

<script>

// using Function

function handleJustClick()

{

const clickMethod=document.getElementById('clicking-
method');

clickMethod.innerText='Lorem ipsum dolor sit amet
consectetur adipisicing elit. Quia eligendi porro odit quo blanditiis
commodi, vel magnam neque iure voluptate saepe accusantium molestias
ipsum magni deleniti. Minima veniam cupiditate quasi.';

document.body.style.color='white';

document.body.style.backgroundColor='green';

}

// direct

document.getElementById('handle-
event').addEventListener('click',function(){

document.body.style.backgroundColor='green';

```

    });

    // direct with input field
    document.getElementById('update-
name').addEventListener('click',function(){
        const nameField=document.getElementById('name-field');
        console.log(nameField.value);

        const clickMethod=document.getElementById('clicking-
method');

        clickMethod.innerText=nameField.value;
        nameField.value="";

        document.body.style.color='white';
        document.body.style.backgroundColor='green';

    });
</script>
</body>

```

25-5 Create a comment box and display comment

```

<body>
    <h2>Add your comment, Please!!</h2>
    <section>
        <h4>Your comments</h4>
        <div id="comment-container">
            <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Eos veniam aspernatur totam id adipisci,
            voluptate a. Architecto nesciunt aliquid, laudantium
iste nihil saepe at deleniti, porro culpa neque
            distinctio ad?</p>
            <p>Vero sint exercitationem ad quasi explicabo esse quae
dignissimos aperiam, mollitia rerum eligendi
            dolorem porro vel perferendis! Quidem, autem quasi!
Deleniti totam odit officia esse suscipit nam fugit
            ipsum numquam!</p>

        </div>
        <div>
            <textarea name="" id="new-comment" cols="60"
rows="4"></textarea>
            <br>
            <button id="submit-comment">Post</button>

```

```

        </div>
    </section>
    <script>
        // 1. add button event handler
        document.getElementById('submit-
comment').addEventListener('click', function () {

            // 2. get user comment
            const commentBox = document.getElementById('new-comment');
            // const userComment = commentBox.value;

            // 3. create comment paragraph
            const newComment = document.createElement('p');
            newComment.innerText = commentBox.value;

            // 4. append the comment
            const commentContainer = document.getElementById('comment-
container');
            commentContainer.appendChild(newComment);

            // 5. clean the comment box
            commentBox.value = '';

        })
    </script>
</body>

```

25-6 Simple github like delete confirmation button

```

<body>

    <h2>More Events</h2>
    <section>
        <h3 id="secret-info">Very very important secret
information</h3>
        <input id="delete-confirm" type="text" placeholder="Please
type delete">
        <button id="delete-btn" disabled>Delete</button>
    </section>

    <script>
        document.getElementById('delete-
btn').addEventListener('click', function() {
            document.getElementById('secret-info').style.display =
'none';

```

```

    });

    // focus section
    document.getElementById('delete-
confirm').addEventListener('focus', function() {
        document.body.style.backgroundColor = 'red';
    });

    // blur section
    document.getElementById('delete-
confirm').addEventListener('blur', function() {
        document.body.style.backgroundColor = 'white';
    });

    // // keydown section
    // document.getElementById('delete-
confirm').addEventListener('keydown', function() {
    //     const deleteField = document.getElementById('delete-
confirm');
    //     console.log(deleteField.value);
    // });

    // // keypress section
    // document.getElementById('delete-
confirm').addEventListener('keypress', function() {
    //     const deleteField = document.getElementById('delete-
confirm');
    //     console.log(deleteField.value);
    // });

    // keyup section
    document.getElementById('delete-
confirm').addEventListener('keyup', function (event) {
        const deleteBtn = document.getElementById('delete-btn');
        if (event.target.value == 'delete' || event.target.value
== 'Delete') {
            deleteBtn.removeAttribute('disabled');
        }
        else {
            deleteBtn.setAttribute('disabled', true);
        }
    });

    // // change section

```

```

        // document.getElementById('delete-
confirm').addEventListener('change', function() {
        //     const deleteField = document.getElementById('delete-
confirm');
        //     console.log(deleteField.value);
        // });
    </script>
</body>

```

25-7 (semi-advanced) Event bubble and Stop propagating

Event bubbling is a **method of event propagation** in the HTML DOM API when an event is in an element inside another element, and both elements have registered a handle to that event. ... In event bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

Event bubble stop: stopPropagation()

stopPropagation() The stopPropagation() method of the Event interface **prevents further propagation of the current event in the capturing** and bubbling phases. It does not, however, prevent any default behaviors from occurring; for instance, clicks on links are still processed.

stopImmediatePropagation() method **stops the rest of the event handlers from being executed**. This method also stops the event from bubbling up the DOM tree. Tip: Use the event.isImmediatePropagationStopped() method to check whether this method was called for the event.

<body>

```

<h2>Event Bubble vai</h2>
<section id="section-container">
    <h4>Bubble vai list</h4>
    <ul id="list-container">
        <li class="item">Lorem, ipsum.</li>
        <li id="second-item" class="item">Minus, ducimus!</li>
        <li class="item">Facilis, voluptates.</li>
        <li class="item">Nemo, sunt?</li>
        <li class="item">Reiciendis, aliquid!</li>
        <li class="item">Nesciunt, excepturi!</li>
    </ul>
</section>

<script>

```



```

        document.getElementById('second-
item').addEventListener('click',function(event){
            console.log("second item click");
            // event.stopPropagation();
            event.stopImmediatePropagation();
        });

        document.getElementById('second-
item').addEventListener('click', function (event) {
            console.log('second2 item clicked');

        });
        document.getElementById('second-
item').addEventListener('click', function (event) {
            console.log('second3 item clicked');

        });
        document.getElementById('second-
item').addEventListener('click', function (event) {
            console.log('second4 item clicked');

        });

        // ul list
        document.getElementById('list-
container').addEventListener('click',function(){
            console.log("list-container click");
        });

        // section
        document.getElementById('section-
container').addEventListener('click',function(){
            console.log("section-container click");
        });
    </script>

</body>

```

25-8 (advanced) Event delegate and benefit of Event bubble

Event Delegation is basically a pattern to handle events efficiently. Instead of adding an event listener to each and every similar element, we can add an event listener to a parent element and call an event on a particular target using the.target property of the event object.

```

<body>

```

```

<h2>Event Delegate</h2>
<ul id="item-container">
  <li class="item">Lorem, ipsum.</li>
  <li class="item">Tempore, distinctio!</li>
  <li class="item">Iste, omnis?</li>
  <li class="item">Facere, adipisci?</li>
  <li class="item">Molestiae, nemo!</li>
  <li class="item">Molestiae, eius.</li>
</ul>
<button id="add-item">Add Item</button>

<script>
  // remove item
  const items = document.getElementsByClassName("item");
  for (const item of items) {
    console.log(item);
    item.addEventListener("click", function (event) {
      // document.getElementById('item-
container').removeChild(item);
      event.target.parentNode.removeChild(event.target);
    });
  }

  // add item
  document
    .getElementById("add-item")
    .addEventListener("click", function () {
      const li = document.createElement("li");
      li.innerText = "Hamid Hosen";
      li.classList.add("item");

      const parent = document.getElementById("item-container");
      parent.appendChild(li);

      // const items = document.getElementsByClassName("item");
      // for (const item of items) {
      //   console.log(item);
      //   item.addEventListener("click", function (event) {
      //     // document.getElementById('item-
container').removeChild(item);
      //     event.target.parentNode.removeChild(event.target);
      //   });
      // }
    });

```

```

        document.getElementById('item-
container').addEventListener('click', function (event) {
            event.target.parentNode.removeChild(event.target);
        })
    </script>
</body>

```

25-9 DOM events summary and home work

```

<body>

    <h1 id="mouse-header">More Mouse event</h1>
    <input id="email-field" type="text" placeholder="Email Address">
    <button onclick="handleClickGreen()">Click me: Green</button>

    <script>
        function handleClickGreen()
        {
            document.body.style.backgroundColor='green';
        }

        document.getElementById('email-
field').addEventListener('keyup',function(event){
            console.log(event.target.value);
        });

        // document.getElementById('mouse-
header').addEventListener('mousemove',function(){
            // console.log('mouse is moving...giving me shurshuri');
            // });

        // document.getElementById('mouse-
header').addEventListener('mouseenter',function(){
            // console.log('mouse is entering...giving me shurshuri')
            // });

        document.getElementById('mouse-
header').addEventListener('mouseout',function(){
            console.log('mouse tumi kotha na koiya jaio na');
        });
    </script>

</body>

```

