**Topic:  Bubbling, capturing, binding, local storage and session storage**

**Event Bubbling:**

   - Event bubbling is a mechanism where when an event is triggered on a nested element inside another element, the event 'bubbles up' through its ancestors.

   - By default, most events bubble.

   - You can stop the bubbling phase using `**event.stopPropagation()**`.

  <div id="parent">

        <button id="child">

            click me

        </button>

    </div>

    <script>

    let parent=document.getElementById("parent");

    let child=document.getElementById("child");

    child.addEventListener("click", function(event){

        event.stopPropagation()// this will prevent the bubbling to the parent

        console.log("child is clicked");

    })

    parent.addEventListener("click", function(){

        console.log("parent is clicked")

    })

    </script>

**Event Capturing:**

   - Event capturing is the opposite of event bubbling.

   - During the capturing phase, the event is first captured by the outermost element and then propagated to the innermost element.

   - You can listen to events during the capturing phase by passing `true` as the third parameter to `**addEventListener()`.**

  <div id="parent">

        <button id="child">

            click me

        </button>

    </div>

    <script>

    let parent=document.getElementById("parent");

    let child=document.getElementById("child");

**child.addEventListener**("click", function(event){

        console.log("child is clicked");

    })

**parent.addEventListener**("click", function(){

        console.log("parent is clicked")

    },true)

**Event Binding:**

   - Event binding refers to the **process of attaching event listeners to DOM elements**.

   - This is typically done using `**addEventListener()**` or by assigning event handler properties like `onclick`.

**Session storage and local storage**

**Session storage** is a part of the Web Storage API in web browsers that provides a way to store key-value pairs locally on the client-side.

* sessionStorage maintains a separate storage area for each given origin that's available for the duration of the page session (as long as the browser is open, including page reloads and restores).
* Data stored in sessionStorage is cleared when the page session ends.
* Data is only accessible within the window/tab that set it.

**// Storing data in sessionStorage**

sessionStorage.setItem('username', 'John');

**// Retrieving data from sessionStorage**

let username = sessionStorage.getItem('username');

console.log(username); // Output: John

**// Removing data from sessionStorage**

sessionStorage.removeItem('username');

**localStorage:**

localStorage is a feature of web browsers that allows web applications to store key-value pairs locally on the client-side. It provides a persistent storage mechanism, meaning that the data stored in localStorage remains available even after the browser is closed and reopened, and across browser sessions.

* localStorage does almost the same thing as sessionStorage, but it persists even when the browser is closed and reopened.
* Data stored in localStorage has no expiration time.
* Data is accessible across windows and tabs within the same origin.

**// Storing data in localStorage**

localStorage.setItem('email', 'example@example.com');

**// Retrieving data from localStorage**

let email = localStorage.getItem('email');

console.log(email); // Output: example@example.com

**// Removing data from localStorage**

localStorage.removeItem('email');

**how to display some data from one page to another page using local storage**

local storage limited to handle only string key/value pairs you can do like below using **JSON.stringify** and while getting value **JSON.parse**

var testObject ={name:"test", time:"Date 2017-02-03T08:38:04.449Z"};

**Put the object into storage:**

localStorage.setItem('testObject', JSON.stringify(testObject));

**Retrieve the object from storage:**

var retrievedObject = localStorage.getItem('testObject');

console.log('retrievedObject: ', JSON.parse(retrievedObject));

**Example: Add to cart functionality**

**//first file**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

    <style>

      .container {

        display: grid;

        grid-template-columns: auto auto auto;

        gap: 20px;

      }

      .container > div {

        padding: 20px;

        border: 1px solid red;

      }

      .container > div > div,

      h1 {

        padding: 10px;

        border: 2px solid blue;

      }

    </style>

  </head>

  <body>

    <h1>

      <button onclick="cart()">cart</button>

    </h1>

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

    <script>

      async function apicall() {

        var newarr = [];

        var result = await fetch("https://fakestoreapi.com/products");

        var apidata = await result.json();

        console.log(apidata);

        var iterated = apidata.map((val) => {

          // console.log(val);

          var row = document.getElementById("row");

          var main = document.createElement("div");

          var child1 = document.createElement("h1");

          var child2 = document.createElement("div");

          var child3 = document.createElement("div");

          var child4 = document.createElement("div");

          child1.innerHTML = val.id + " <br>";

          child2.innerHTML = val.title + " <br>";

          child3.innerHTML = val.description + " <br>";

          child4.innerHTML = val.price + " <br>";

          var btn = document.createElement("button");

          btn.innerHTML = "click";

          btn.addEventListener("click", function () {

            newarr.push(val);

            sessionStorage.setItem("arr", JSON.stringify(newarr));

          });

          main.append(child1, child2, child3, child4, btn);

          row.appendChild(main);

        });

      }

      apicall();

      function cart() {

        window.open("sub.html", "\_self");

      }

    </script>

  </body>

</html>

**//second file**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

  </head>

  <body>

    <div id="row"></div>

    <script>

      var newarrdata = JSON.parse(sessionStorage.getItem("arr"));

      console.log(newarrdata);

      var iterated = newarrdata.map((val) => {

        // console.log(val);

        var row = document.getElementById("row");

        var main = document.createElement("div");

        var child1 = document.createElement("h1");

        var child2 = document.createElement("div");

        var child3 = document.createElement("div");

        var child4 = document.createElement("div");

        child1.innerHTML = val.id + " <br>";

        child2.innerHTML = val.title + " <br>";

        child3.innerHTML = val.description + " <br>";

        child4.innerHTML = val.price + " <br>";

        var btn = document.createElement("button");

        btn.innerHTML = "click";

        btn.addEventListener("click", function () {

          main.style.display = "none";

        });

        main.append(child1, child2, child3, child4, btn);

        row.appendChild(main);

      });

    </script>

  </body>

</html>