

Module 39-Browser API and methods

আজকে যে যে জিনিস তোমাকে জানতে হবে

১. alert, confirm, prompt এই গুলো কোনটা দিয়ে কী কাজ করে। এবং এইগুলোর মধ্যে পার্থক্য কি?
২. URL এর মধ্যে মেজর যে যে অংশ থাকে যেমন query string, href, hash, port এইগুলো জানতে হবে।
৩. location এর মধ্যে href দিয়ে যে যে ওয়েবসাইট দেখতেছো সেটা চেইঞ্জ করে ফেলা যায় সেটা জানতে হবে।
৪. History api এর মধ্যে যে go দিয়ে সামনে পিছনে যেতে পারো সেটা একটু খেয়াল করতে হবে।
৫. Cookies কি জিনিস। এইটা দিয়ে কি করা হয়। সেটা সম্পর্কে হালকা করে জানতে হবে।
৬. local storage আর session storage কখন কোনটা ইউজ করতে হয় সেটা জানতে হবে
৭. Cookies, local storage, session storage এর মধ্যে পার্থক্য কি সেটা জানতে হবে।

39-1 Module Overview, BOM vs DOM, Browser API

39-2 Browser alert, confirm, prompt with examples

What is alert JavaScript?

The alert() method in JavaScript is **used to display a virtual alert box**. It is mostly used to give a warning message to the users. It displays an alert dialog box that consists of some specified message (which is optional) and an OK button.

How do you confirm in JavaScript?

The confirm() method displays a dialog box with a message, an OK button, and a Cancel button. The confirm() method returns true if the user clicked "OK", otherwise false .

What is a prompt in JavaScript?

JavaScript prompt() dialog box. The prompt() method in JavaScript is **used to display a prompt box that prompts the user for the input**. It is generally used to take the input from the user before entering the page. It can be written without using the window prefix.

39-3 Location, URL parts, query string, href, hash, assign, reload

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>

<body>
  <h1>Location</h1>
  <button onclick="refreshPage()">Refresh</button>
  <button onclick="visitPH()">Visit PH</button>
  <script>
    const refreshPage = () => {
      location.reload();
    }
    const visitPH = () => {
      location.assign('https://web.programming-hero.com/');
    }
  </script>
</body>

</html>
```

39-4 History api to navigate back, forward, go, history length

https://developer.mozilla.org/en-US/docs/Web/API/History_API

39-5 (advanced) Cookies, dev tool application tab, get cookie value

39-6 local storage, session storage, storage info in browser storage

Local storage has 4 methods: এ ডাটা স্থায়ীভাবে থাকে

- **setItem() Method** – This method takes two parameters one is key and another one is value. It is used to store the value in a particular location with the name of the key.
`localStorage.setItem(key, value)`
- **getItem() Method** – This method takes one parameter that is key which is used to get the value stored with a particular key name.
`localStorage.getItem(key)`
- **removeItem() Method** – This is method is used to remove the value stored in the memory in reference to key.
`localStorage.removeItem(key)`
- **clear() Method** – This method is used to clear all the values stored in localStorage.
`localStorage.clear()`

Session Storage has 4 methods: এ ডাটা সাময়িকভাবে থাকে

- **setItem() Method** – This method takes two parameters one is key and another one is value. It is used to store the value in a particular location with the name of the key.
`sessionStorage.setItem(key, value)`
- **getItem() Method** – This method takes one parameter that is key which is used to get the value stored with a particular key name.
`sessionStorage.getItem(key)`
- **removeItem() Method** – This is method is used to remove the value stored in the memory in reference to key.
`sessionStorage.removeItem(key)`
- **clear() Method** – This method is used to clear all the values stored in the session storage
`sessionStorage.clear()`

getItem() ব্যবহার করে Local Storages এবং Session Storage থেকে নির্দিষ্ট মান রিটার্ন করা হয়।

```
storage.setItem(keyName, keyvalue)
```

setItem() ব্যবহার করে Local storage এবং Session Storage এ মান সেট করা হয়।

Local Storage এবং Session Storage এর অবজেক্ট নিয়ে কাজ করতে হলে সেটাকে প্রথমে JSON.parse() করতে হবে।

Local Storage এবং Session Storage এ কোন অবজেক্ট সেট করার আগে JSON.stringify() করতে হবে।

39-7 Dynamically set and read from local, session storage

```
const addItem = () => {
  const nameField = document.getElementById("product-name");
  const name = nameField.value;

  // display in ui
  displayProducts(name);

  // add to local storage
  addProductToCard(name);

  nameField.value = "";
};

const displayProducts = (name) => {
  const ul = document.getElementById("products");
  const li = document.createElement("li");
  li.innerText = name;

  ul.appendChild(li);
};
```

```

const getCard = () => {
  // getItem() ব্যবহার করে Local Storages এবং Session Storage থেকে
  নির্দিষ্ট মান রিটার্ন করা হই।
  const card = localStorage.getItem("card");
  let cartObj;

  if (card) {
    // Local Storage এবং Session Storage এর অবজেক্ট নিয়ে কাজ করতে
    হলে সেটাকে প্রথমে JSON.parse() করতে হবে।
    cartObj = JSON.parse(card);
  } else {
    cartObj = {};
  }
  return cartObj;
};

const addProductToCard = (name) => {
  const cart = getCard();
  cart[name] = 1;

  // Local Storage এবং Session Storage এ কোন অবজেক্ট সেট করার আগে
  JSON.stringify() করতে হবে।
  const cartStringified = JSON.stringify(cart);
  localStorage.setItem("card", cartStringified);

  // console.log(cart);
};

```

39-8 (advanced) retrieve local storage value and display them

```

const displayLocalStorageCart = () => {
  const cart = getCard();

  for (const name in cart) {
    displayProducts(name);
  }
};

const addItem = () => {
  const nameField = document.getElementById("product-name");
  const name = nameField.value;

  if (!name) {

```

```

    return;
}

// display in ui
displayProducts(name);

// add to local storage
addProductToCard(name);

//clear
nameField.value = "";
};

const displayProducts = (name) => {
  const ul = document.getElementById("products");
  const li = document.createElement("li");
  li.innerText = name;

  ul.appendChild(li);
};

const getCard = () => {
  // getItem() ব্যবহার করে Local Storages এবং Session Storage থেকে
  নির্দিষ্ট মান রিটার্ন করা হই।
  const card = localStorage.getItem("cart");
  let cartObj;

  if (card) {
    // Local Storage এবং Session Storage এর অবজেক্ট নিয়ে কাজ করতে
    হলে সেটাকে প্রথমে JSON.parse() করতে হবে।
    cartObj = JSON.parse(card);
  } else {
    cartObj = {};
  }
  return cartObj;
};

const addProductToCard = (name) => {
  const cart = getCard();

  if (cart[name]) {
    cart[name] = cart[name] + 1;
  } else {
    cart[name] = 1;
  }
};

```

```

    }

    // Local Storage এবং Session Storage এ কোন অবজেক্ট সেট করার আগে
    JSON.stringify() করতে হবে।
    const cartStringified = JSON.stringify(cart);
    localStorage.setItem("cart", cartStringified);

    // console.log(cart);
};

const placeOrder = () => {
    document.getElementById("products").textContent = "";
    localStorage.removeItem("cart");
};

displayLocalStorageCart();

```

39-9 Module Summary, ContentEditable, chrome restart

```

alert("");
confirm("are you single?");
prompt("name");

// location.reload()
// location.href = ''
// location.hash =
// http://127.0.0.1:5500/shopping.html

// history.back()
// history.go()

// cookies

// sessionStorage vs localStorage
// cookies vs sessionStorage vs localStorage

```