Day-11:DOM-III

Forms in Javascript:

- Forms will generally contain input tags.
- Take an example of any application form online, it consists of many input tags.
- So, whenever you are using some kind of input tags, where you want to take input from user, use forms.
- There are certain steps needs to be followed while using forms

Steps:

1. Wrap your input tags inside form tag.
2. Instead of button tag you should use `<input type="submit"/>`
3. Add eventListener to `form` tag.
4. Event name should be `submit`

   ```
   addEventListener("submit", myFunction)
   ```

5. Forms by default tries to send data to backend when you click on submit, to stop default behaviour use `event.preventDefault()`

- Let’s see the difference between onClick and addEventListener by looking into following example

Without using form tag

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
  </head>
  <body>
```

```
    <input id="name" type="text" placeholder="enter name" />
    <input id="mail" type="email" placeholder="Enter email address" />
    <button onClick="getData()">Submit</button>
    <h1 id="nameDisplay">display name here</h1>
    <h2 id="emailDisplay">display email here</h2>
  </body>
</html>

<script>
  function getData() {
    var username = document.getElementById("name").value;
    var email = document.getElementById("mail").value;
    document.getElementById("nameDisplay").innerText = username;
    document.getElementById("emailDisplay").innerText = email;
  }
</script>
```

Live code : Codepen

With Form tag

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
  </head>
  <body>
    <form>
      <input id="name" type="text" placeholder="enter name" />
      <input id="mail" type="email" placeholder="Enter email address" />
      <input type="submit" />
    </form>
    <h1 id="nameDisplay">display name here</h1>
    <h2 id="emailDisplay">display email here</h2>
  </body>
</html>

<script>
  document.querySelector("form").addEventListener("submit", getData);
  function getData() {
    event.preventDefault();
    var username = document.getElementById("name").value;
    var email = document.getElementById("mail").value;
    document.getElementById("nameDisplay").innerText = username;
    document.getElementById("emailDisplay").innerText = email;
  }
</script>
```

Live code : Codepen

preventDefault()

- The preventDefault() method cancels the event if it is cancelable, meaning that the default action that belongs to the event will not occur.
- For example, this can be useful when:
  - Clicking on a "Submit" button, prevent it from submitting a form.
  - Clicking on a link, prevent the link from following the URL.

Document createElement()

- The `createElement()` function in JavaScript is used to programatically add elements to the DOM.

Syntax

```
document.createElement(type)
```

- It has one required argument, the type of element to create, like `'div'` or `'img'`.
- Let's see how we generally create an element using HTML.
- For Example I want to create a <h1> tag with innertext of Masai School

```
<html lang="en">
<head>
  <title>Document</title>
</head>
<body>
  <h1>Masai School</h1>
</body>
</html>
```

- What are the steps we followed here?
  1. We have created `<h1>` tag - `<h1> </h1>`
  2. We have added innerText to it - `<h1>Masai School</h1>`
  3. We have added(appended) `h1` tag to `body` tag in order to display it on browser.

- Now let’s try to create same h1 tag using javascript by following same steps

```
<html lang="en">
<head>
  <title>Document</title>
</head>
<body>
</body>
</html>

<script>
  // Step1: Creating h1 tag
  let heading = document.createElement("h1");
  // Step2: Adding innerText
  heading.innerText = "Masai School";
  // Step3: Appending to body
  document.querySelector("body").append(heading)
</script>
```

Live code : Codepen

Example

- In the following example, initially the **div** section consists of only **one p tag.** But later on, one more p tag is created and added to the div section as shown below.

```
<html>
  <body>
    <div id="new">
      <p>Masai School</p>
    </div>
  </body>
</html>
<script></script>
```

```
<html>
  <body>
    <div id="new">
      <p>Masai School</p>
    </div>
  </body>
</html>
<script>
      var newPara = document.createElement("p");
      newPara.innerText = "The Coding School that cares about you";
```

```
      document.getElementById("new").append(newPara);
</script>
```

Live code : Codepen

- Now if you want to add a class/id to `<p>Masai School</p>`, we can use setAttribute property

Element setAttribute()

- The `setAttribute()` method sets a new value to an attribute.
- Here attributes can be any of the following
  - id
  - class
  - href
  - src, etc

Syntax

```
element.setAttribute(attributeName, attributeValue)
```

Example:

```
<html>
  <body>
    <div id="new">
      <p>Masai School</p>
    </div>
  </body>
</html>
<script>
      var newPara = document.createElement("p");
      newPara.innerText = "The Coding School that cares about you";
      // Setting class of "para" to p tag
      newPara.setAttribute("class","para")
      // Setting id of "container" to p tag
      newPara.setAttribute("id","container")
      document.getElementById("new").append(newPara);
</script>
```

- Now whatever styles we write for that class “para” or id “container” will be applied to that <p> tag

```
<html>
<style>
.para {
  font-size:40px
}
</style>
  <body>
    <div id="new">
      <p>Masai School</p>
    </div>
  </body>
</html>
<script>
      var newPara = document.createElement("p");
      newPara.innerText = "The Coding School that cares about you";

      // Setting class of "para" to p tag
      newPara.setAttribute("class","para")

      // Setting id of "container" to p tag
      newPara.setAttribute("id","container")

      document.getElementById("new").append(newPara);
</script>
```

Live code : Codepen

- Add a href attribute to an <a> element:

```
myAnchor.setAttribute("href", "https://www.google.com");
```

Element getAttribute()

- The `getAttribute()` method returns the value of an element's attribute.

Syntax

```
element.getAttribute(name)
```

Example:

```
<html>
  <body>
    <h2>The getAttribute() Method</h2>
    <button id="myButton">BUTTON</button>
    <p>The value of the "id" attribute of the button is:</p>
    <p id="demo"></p>
  </body>
</html>

<script>
  const myButton = document.getElementById("myButton");
  let out = myButton.getAttribute("id");
  document.getElementById("demo").innerText = out;
</script>
```

Live code: Codepen