

Submitted to : *Sir Danish Asim*
Submitted by: *Riffat Jabeen* [78]

Assignment 6
20th August 2023

Module 2 : Front End Development (JS Web)

Submission Requirements: Please upload your codes in PDF File on Google Classroom in the relevant Assignment section.

Note: Plagiarism is a serious violation. Zero marks will be awarded in case plagiarism is found. You are required to perform the following tasks.

Class Questions:

- Remove direct text placed inside a div

```
<div>
  <h1 class="heading">Class Questions (Task 1)</h1>
  <h3>Remove direct text placed inside a div</h3>
  <div id="direct">
    Direct text here...
    <h3>heading 1</h3>
    <h3>heading 2</h3>
  </div>
  <script>
    const direct = document.getElementById('direct');
    direct.removeChild(direct.firstChild);
  </script>
</div>
```

Class Questions (Task 1)

Remove direct text placed inside a div

Direct text here...

heading 1

heading 2

Class Questions (Task 1)

Remove direct text placed inside a div

heading 1

heading 2

- Add/Remove text in an element based on hover and blur on button respectively.

```
<div>
  <h1 class="heading">Class Questions (Task 2)</h1>
  <h3>Add/Remove text in an element based on hover and blur on button respectively</h3>
  <div>
    <h1 id="hover"></h1>
    <Button onmouseover="hoverOn()">Hover On</Button>
    <h1 id="blur">Text should be removed when you blur the button</h1>
    <Button onblur="blurOn()">Blur On</Button>
  </div>
  <script>
    const hoverOn = () => {
      const hover = document.getElementById('hover');
      hover.innerHTML = 'Text Added';
    }
    const blurOn = () => {
      const hover = document.getElementById('blur');
      hover.innerHTML = '';
    }
  </script>
</div>
```

Class Questions (Task 2)

Add/Remove text in an element based on hover and blur on button respectively

Hover On

Text should be removed when you blur the button

Blur On

Class Questions (Task 2)

Add/Remove text in an element based on hover and blur on button respectively

Text Added

Hover On

Blur On

Part 1:

- Use `document.getElementById(id)` to find an element with a specific ID and store it in a variable.

```
<div>
  <h1 class="heading">Task 1</h1>
  <h3>Use document.getElementById(id) to find an element with
  a specific ID and store it in a variable</h3>
  <h1 id="head">123</h1>
  <script>
    |   const store = document.getElementById('head');
  </script>
</div>
```

- Utilize `document.getElementsByTagName(name)` to find all elements with a specific tag name and log them to the console.

```
<div>
  <h1 class="heading">Task 2</h1>
  <h3>Utilize document.getElementsByTagName(name) to find all
  elements with a specific tag name and log them to the
  console</h3>
  <h1>heading 1</h1>
  <h1>heading 2</h1>
  <script>
    |   const log = document.getElementsByTagName('h1');
    |   console.log(log);
  </script>
</div>
```

Task 2

Utilize `document.getElementsByTagName(name)` to find all elements with a specific tag name and log them to the console

heading 1

heading 2



- Use `document.getElementsByClassName(name)` to find all elements with a specific class name and apply a CSS style to each of them.

```
<div>
  <h1 class="heading">Task 3</h1>
  <h3>Use document.getElementsByClassName(name) to find all
  elements with a specific class name and apply a CSS style
  to each of them</h3>
  <h2 class="style">h2</h2>
  <p class="style">p</p>
  <h3 class="style">h3</h3>
  <script>
    const byName = document.getElementsByClassName('style');

    for (let i = 0; i < byName.length; i++) {
      byName[i].style.backgroundColor = 'indianred';
    }
  </script>
</div>
```

```
.style{
  background-color: rgb(144, 96, 165);
  color: firebrick;
  font-weight: bold;
  width: 30%;
}
```

Task 3

Use `document.getElementsByClassName(name)` to find all elements with a specific class name and apply a CSS style to each of them

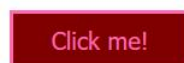


- Change the inner HTML of an element with `element.innerHTML` to display new content when a button is clicked.

```
<div>
  <h1 class="heading">Task 4</h1>
  <h3>Change the inner HTML of an element with element.
  innerHTML to display new content when a button is clicked
</h3>
  <button onclick="newContent()">Click me!</button>
  <p id="content">Content</p>
  <script>
    const newContent = () => {
      const content = document.getElementById('content');
      content.innerHTML = '=> NEW Content <=' ;
    };
  </script>
</div>
```

Task 4

Change the inner HTML of an element with `element.innerHTML` to display new content when a button is clicked



Content

Task 4

Change the inner HTML of an element with `element.innerHTML` to display new content when a button is clicked



=> NEW Content <=

- Update the value of an attribute in an HTML element using the `element.attribute` property based on user input.


```

<div>
  <h1 class="heading">Task 5</h1>
  <h3>Update the value of an attribute in an HTML element
    using the element.attribute property based on user input
  </h3>
  <input id="input" type="text">
  <script>
    const byAttribute = document.getElementById('input');
    byAttribute.setAttribute('placeholder', 'Enter
      something');
  </script>
</div>

```

Task 5

Update the value of an attribute in an HTML element using the element.attribute property based on user input

- Create a function that changes the background color of an element using the element.style.property.

```

<div>
  <h1 class="heading">Task 6</h1>
  <h3>Create a function that changes the background color of
    an element using the element.style.property</h3>
  <div id="bgclr">twist</div>
  <script>
    const backgroundColor = document.getElementById
      ('bgclr');
    backgroundColor.style.backgroundColor = 'indianred';
  </script>
</div>

```

Task 6

Create a function that changes the background color of an element using the element.style.property

twist

- Create a button that, when clicked, uses element.setAttribute(attribute, value) to modify the 'src' attribute of an image.

```

<div>
  <h1 class="heading">Task 7</h1>
  <h3>Create a button that, when clicked, uses element.
  setAttribute(attribute, value) to modify the 'src'
  attribute of an image</h3>
  <button onclick="Setsrc()">Click Me!</button>
  <img class="image" id="image" alt="Cat here">
  <script>
    const Setsrc = () => {
      document.getElementById('image').setAttribute('src',
        'cat.jpg');
    };
  </script>
</div>

```

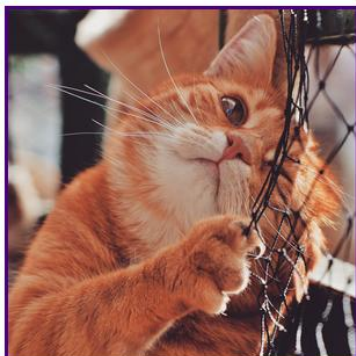
Task 7

Create a button that, when clicked, uses element.setAttribute(attribute, value) to modify the 'src' attribute of an image



Task 7

Create a button that, when clicked, uses element.setAttribute(attribute, value) to modify the 'src' attribute of an image



- Develop a form where users can input their name and a message. On submission, use element.innerHTML to display a personalized message.

```

<div>
  <h1 class="heading">Task 8</h1>
  <h3>Develop a form where users can input their name and a message. On
  submission, use element.innerHTML to
  | display a personalized message</h3>
  <div>
    <form>
      <label for="name">Enter your name <input type="text" name="name"
      id="getName"
      placeholder="name here"></label>
      <label for="name">Enter your message <input type="text" name="name"
      id="getMsg"
      placeholder="message here"></label>
      <button onclick="perMsg()">Submit</button>
      <h1 id="display"></h1>
    </form>
  </div>
  <script>
    const perMsg = () => {
      document.getElementById('display').innerHTML = 'Your response has been
      submitted!';
    };
  </script>
</div>

```

Task 8

Develop a form where users can input their name and a message. On submission, use element.innerHTML to display a personalized message

Enter your name

Enter your message

Task 8

Develop a form where users can input their name and a message. On submission, use element.innerHTML to display a personalized message

Enter your name

Enter your message

Your response has been submitted!

- Build a list of items with checkboxes. When a checkbox is clicked, use element.style.property to strike through the item's text.


```

<div>
  <h1 class="heading">Task 9</h1>
  <h3>Build a list of items with checkboxes. When a checkbox
  is clicked, use element.style.property to strike
  |   through the item's text</h3>
  <div>
    <ol>
      <h3>Select your destination</h3>
      <li>
        <label id="country_1" for="country_1"><input
        onclick="style_1()" type="checkbox"
        name="country_1">Pakistan</label>
      </li>
      <li>
        <label id="country_2" for="country_2"><input
        onclick="style_2()" type="checkbox"
        name="country_2">Turkey</label>
      </li>
      <li>
        <label id="country_3" for="country_3"><input onclick="style_3()"
        type="checkbox"
        name="country_3">Korea</label>
      </li>
    </ol>
  </div>
  <script>
    const style_1 = () => {
      const Style = document.getElementById('country_1');
      Style.style.textDecoration = 'line-through';
    };
    const style_2 = () => {
      const Style = document.getElementById('country_2');
      Style.style.textDecoration = 'line-through';
    };
    const style_3 = () => {
      const Style = document.getElementById('country_3');
      Style.style.textDecoration = 'line-through';
    };
  </script>
</div>

```

Task 9

Build a list of items with checkboxes. When a checkbox is clicked, use element.style.property to strike through the item's text

Select your destination

1. ☐ Pakistan
2. ☐ Turkey
3. ☐ Korea

Task 9

Build a list of items with checkboxes. When a checkbox is clicked, use `element.style.property` to strike through the item's text

Select your destination

1. ☒ Pakistan
2. ☒ Turkey
3. ☒ Korea

- Use `document.createElement(element)` to create a new paragraph element and append it to an existing div.

```
<div>
  <h1 class="heading">Task 10</h1>
  <h3 id="paraAppend">Use document.createElement(element) to create a new paragraph
  element and append it to an existing div</h3>
  <script>
    const newPara = document.createElement('h5');
    const addText = document.createTextNode('=> This is the appended data : 12345678');
    newPara.appendChild(addText);
    const paraAppend = document.getElementById('paraAppend')
    paraAppend.appendChild(newPara);
  </script>
</div>
```

Task 10

Use `document.createElement(element)` to create a new paragraph element and append it to an existing div

⇒ This is the appended data : 12345678

- Create a button that, when clicked, removes a specific list item using `document.removeChild(element)`.

```
<div>
  <h1 class="heading">Task 11</h1>
  <h3>Create a button that, when clicked, removes a specific list item using document.
  removeChild(element)</h3>
  <div>
    <button onclick="remove()">Click Me!</button>
    <ol id="items">
      <li>Header</li>
      <li>Main</li>
      <li>Footer</li>
    </ol>
  </div>
  <script>
    const remove = () => {
      const toRemove = document.getElementById('items');
      toRemove.removeChild(toRemove.children[1])
    }
  </script>
</div>
```

Task 11

Create a button that, when clicked, removes a specific list item using `document.removeChild(element)`

Click Me!

1. Header
2. Main
3. Footer

Task 11

Create a button that, when clicked, removes a specific list item using `document.removeChild(element)`

Click Me!

1. Header
2. Footer

- Implement a feature that adds a new list item to an existing unordered list using `document.appendChild(element)`.

```
<div>
  <h1 class="heading">Task 12</h1>
  <h3>Implement a feature that adds a new list item to an existing unordered list using
  | document.appendChild(element)</h3>
  <ul id="newItem">
  |   <li>Header</li>
  |   <li>Main</li>
  |   <li>Footer</li>
  </ul>
  <script>
  |   const newItem = document.createElement('li');
  |   const newText = document.createTextNode('Sidebar');
  |   newItem.appendChild(newText);
  |   const itemAppend = document.getElementById('newItem')
  |   itemAppend.appendChild(newItem);
  </script>
</div>
```

Task 12

Implement a feature that adds a new list item to an existing unordered list using `document.appendChild(element)`

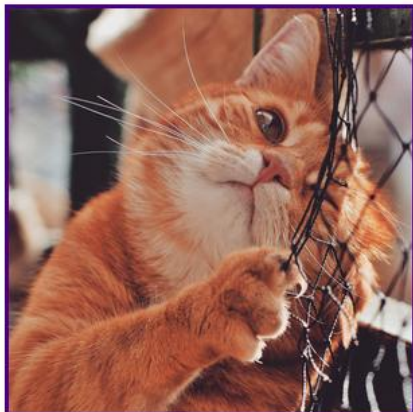
- Header
- Main
- Footer
- Sidebar

- Replace an existing image on a webpage with a new image by using `document.replaceChild(new, old)`.

```
<div>
  <h1 class="heading">Task 13</h1>
  <h3>Replace an existing image on a webpage with a new image by using document.
  replaceChild(new, old)</h3>
  <div id="replaceImage">
    |   
  </div>
  <script>
    |   const replaceImage = document.getElementById('replaceImage');
    |   const oldImage = document.getElementById('preImg');
    |   const newImage = document.createElement('img');
    |   newImage.src = "flower.jpg";
    |   newImage.className = 'image';
    |   newImage.alt = 'flower';
    |   replaceImage.replaceChild(newImage , oldImage);
  </script>
</div>
```

Task 13

Replace an existing image on a webpage with a new image by using document.replaceChild(new, old)



Task 13

Replace an existing image on a webpage with a new image by using `document.replaceChild(new, old)`



- Design a page with two sections. Upon clicking a button, swap the positions of these sections using the `document.replaceChild(new, old)` method.

```
<div>
  <h1 class="heading">Task 14</h1>
  <h3>Design a page with two sections. Upon clicking a button, swap the positions of these sections using the
  document.replaceChild(new, old) method</h3>
  <button onclick="secReplace()">Click Me!</button>
  <div id="section">
    <section id="sec_1"></section>
    <section id="sec_2"></section>
  </div>
  <script>
    const secReplace = () => {
      const section = document.getElementById('section');
      const sec_1 = document.getElementById('sec_1');
      const sec_2 = document.getElementById('sec_2');

      const temp_1 = sec_1.cloneNode(true);
      const temp_2 = sec_2.cloneNode(true);

      section.replaceChild(temp_2, sec_1);
      section.replaceChild(temp_1, sec_2);
    }
  </script>
</div>
```


Task 14

Design a page with two sections. Upon clicking a button, swap the positions of these sections using the `document.replaceChild(new, old)` method



Task 14

Design a page with two sections. Upon clicking a button, swap the positions of these sections using the `document.replaceChild(new, old)` method



- Create a simple website layout with placeholders. Use JavaScript to replace these placeholders with actual content using `document.replaceChild(new, old)`.

```
<div>
  <h1 class="heading">Task 15</h1>
  <h3>Create a simple website layout with placeholders. Use JavaScript to replace these placeholders with
  actual content using document.replaceChild(new, old)</h3>
  <div>
    <header id="header"> Header </header>
    <main id="main"> Main content</main>
    <footer id="footer"> Footer </footer>
  </div>
  <script>
    const header = document.getElementById('header');
    const main = document.getElementById('main');
    const footer = document.getElementById('footer');

    const actualHeader = document.createElement('h1');
    actualHeader.innerHTML = `
    <div class = 'head'>
      <h2> Welcome to our Website</h2>
      <ul class = "navStyl">
        <li>Home</li>
        <li>About</li>
        <li>Services</li>
        <li>Contact Us</li>
      </ul>
    </div>
    `;
  </script>
</div>
```

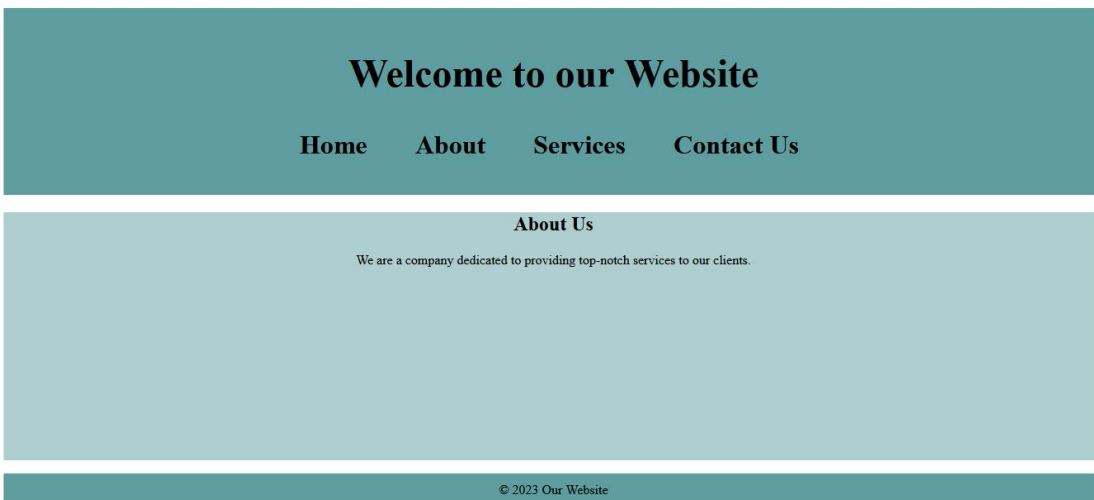
```

const actualContent = document.createElement('section');
actualContent.innerHTML = `
<div class = "main">
  <h2>About Us</h2>
  <p>We are a company dedicated to providing top-notch services to our clients.</p>
</div>
`;

const actualFooter = document.createElement('footer');
actualFooter.innerHTML = '<p class = "head"> © 2023 Our Website </p>';
header.parentNode.replaceChild(actualHeader, header);
main.parentNode.replaceChild(actualContent, main);
footer.parentNode.replaceChild(actualFooter, footer);

</script>
</div>

```



- Create a button on a webpage. Attach an event listener using `addEventListener` to change the button's text to "Clicked!" when it's clicked.

```

<div>
  <h1 class="heading">Task 16</h1>
  <h3>Create a button on a webpage. Attach an event listener using addEventListener to
  change the button's text to "Clicked!" when it's clicked</h3>
  <div id="listener">
    <button id="text">Click Me!</button>
  </div>
  <script>
    const listener = document.getElementById('listener');
    listener.addEventListener('click' , function(event){
      const text = document.getElementById('text');
      text.innerHTML = 'Changed Text'
    });
  </script>
</div>

```

Task 16

Create a button on a webpage. Attach an event listener using `addEventListener` to change the button's text to "Clicked!" when it's clicked



Task 16

Create a button on a webpage. Attach an event listener using `addEventListener` to change the button's text to "Clicked!" when it's clicked

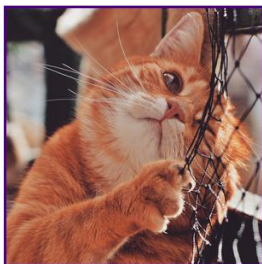
Changed Text

- Design an image that changes its source to another image when hovered over by the mouse pointer. Use the `mouseover` and `mouseout` events to achieve this effect.

```
<div>
  <h1 class="heading">Task 17</h1>
  <h3>Design an image that changes its source to another image when hovered over by the
  mouse pointer. Use the mouseover and mouseout events to achieve this effect</h3>
  <div>
    
  </div>
  <script>
    const hover = document.getElementById('hoverOn');
    const originalSrc = hover.src;
    const newSrc = 'flower.jpg';
    hover.addEventListener('mouseover', () => {
      hover.src = newSrc;
    });
    hover.addEventListener('mouseout', () => {
      hover.src = originalSrc;
    });
  </script>
</div>
```

Task 17

Design an image that changes its source to another image when hovered over by the mouse pointer. Use the `mouseover` and `mouseout` events to achieve this effect



Task 17

Design an image that changes its source to another image when hovered over by the mouse pointer. Use the mouseover and mouseout events to achieve this effect



- Build a registration form with input fields for name, email, and password. Use the submit event on the form to validate that the password is at least 8 characters long. Display an error message if the validation fails.

```
<div>
  <h1 class="heading">Task 18</h1>
  <h3>Build a registration form with input fields for name, email, and password. Use the
  submit event on the form to validate that the password is at least 8 characters long.
  Display an error message if the validation fails</h3>
  <div>
    <form id="regForm">
      <label for="name">Enter your name <input type="text" name="name" id="getName"
      placeholder="name here"></label>
      <label for="email">Enter your email <input type="email" name="email"
      id="getemail" placeholder="email here"></label>
      <label for="password">Enter password <input type="password" name="password"
      id="getpass" placeholder="password here"></label>
      <input type="submit" value="Register">
      <span id="error"></span>
    </form>
    <script>
      const regForm = document.getElementById('regForm');
      const password = document.getElementById('getpass');
      const error = document.getElementById('error');

      regForm.addEventListener('submit', function(event) {
        const passwordValue = password.value;
        if (passwordValue.length < 8) {
          const error = document.getElementById('error');
          error.innerHTML = 'Password must be atleast 8 char!';
          event.preventDefault();
        }
      });
    </script>
  </div>
</div>
```

Task 18

Build a registration form with input fields for name, email, and password. Use the submit event on the form to validate that the password is at least 8 characters long. Display an error message if the validation fails

Enter your name

Enter your email

Enter password

Task 18

Build a registration form with input fields for name, email, and password. Use the submit event on the form to validate that the password is at least 8 characters long. Display an error message if the validation fails

Enter your name

Enter your email

Enter password

Password must be atleast 8 char!

- Create a text input field. Use the keydown event to count and display in real-time the number of characters entered in the input field. Update the count as the user types.

```
<div>
  <h1 class="heading">Task 19</h1>
  <h3>Create a text input field. Use the keydown event to count and display in real-time
  the number of characters entered in the input field. Update the count as the user type</
  h3>
  <label for="textInput">Type something <input type="text" id="textInput"
  placeholder="start typing"></label>
  <p id="Count">Character count : 0</p>

  <script>
    const inputField = document.getElementById('textInput');
    const CountDisplay = document.getElementById('Count');

    inputField.addEventListener('keydown', function(event){
      const text = inputField.value;
      const Count = text.length;
      CountDisplay.textContent = `Character count: ${Count}`;
    });
  </script>
</div>
```

Task 19

Create a text input field. Use the keydown event to count and display in real-time the number of characters entered in the input field. Update the count as the user type

Type something

Character count : 0

Task 19

Create a text input field. Use the keydown event to count and display in real-time the number of characters entered in the input field. Update the count as the user type

Type something

Character count: 8