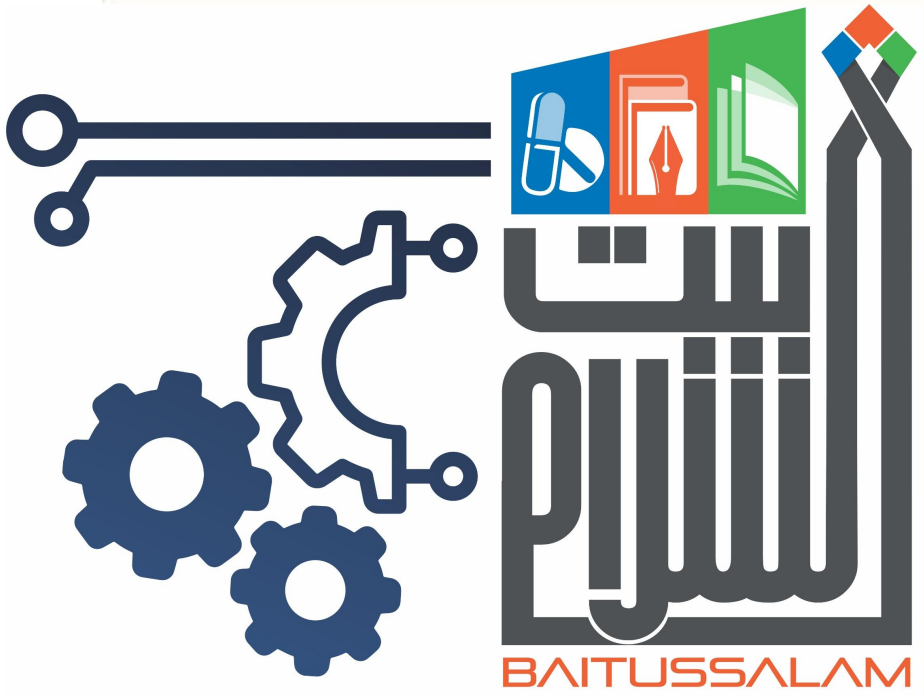



# BAITUSSALAM
# —TECH PARK—

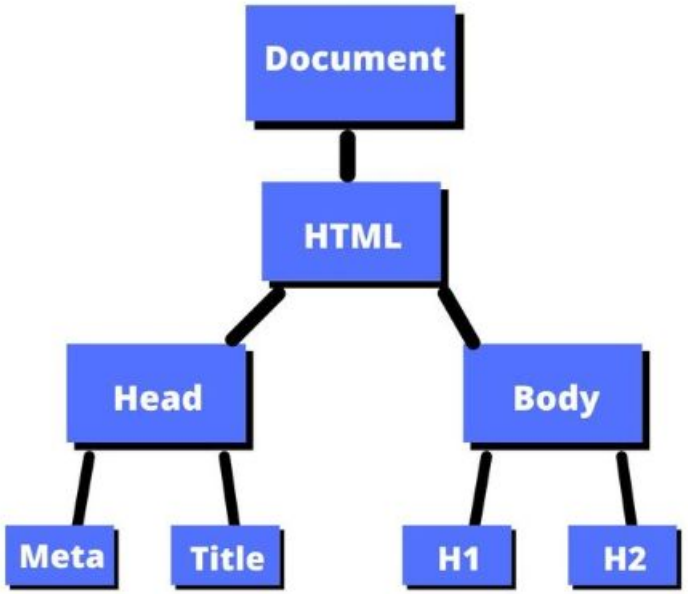
# Class Agenda

BAITUSSALAM
—TECH PARK—

# DOM API, Date and Time forEach, setInterval and setTimeout Function,

# What is DOM

- DOM stands for **Document Object Model**
- It is a way to represent **HTML documents in a tree structure**
- The DOM allows JavaScript to **interact with HTML and CSS**



# DOM Element Selectors

BAITUSSALAM
—TECH PARK—

### Selecting by ID

```
let header = document.getElementById('main-header');
```

### Selecting by Class

```
let items = document.getElementsByClassName('list-item');
```

### Selecting by Tag Name

```
let paragraphs = document.getElementsByTagName('p');
```

# Get, Set and Remove Attribute

BAITUSSALAM
—TECH PARK—

- HTML elements have attributes (e.g., src, href, class)
- Attributes provide additional information about elements
- JavaScript can get and set these attributes

dom.js > ...

```
1  const title = document.getElementById('title')
2
3  console.log(title)
4
5  const titleAttr = title.getAttribute('class')
6
7  console.log(titleAttr)
8
9  title.setAttribute('class', 'main-title heading')
10
11 console.log(title.getAttribute('class'))
12
13 title.removeAttribute('class')
14
15 console.log(title.getAttribute('class'))
16
```

# DOM Style Property

BAITUSSALAM
—TECH PARK—

- The style property allows you to change the CSS styles of HTML elements directly from JavaScript.
- It can be used to modify elements dynamically.
- Use element.style.property = 'value'; to change a style property.

dom.js > ...

```
1  const title = document.getElementById('title')
2
3  header.style.backgroundColor = 'orange'
4  header.style.fontSize = '48px'
5
6  // Adding Multiple Styles
7  title.style.cssText = 'color: red; background-color: yellow; font-size: 20px;'
```

# DOM Exercise #1

```html
<p id="text">Change my style!</p>
<button id="btn">Change Style</button>
```

**Try these:**

1. Change the text color to `green`.
2. Set the background color to `lightgray`.
3. Increase the font size to `18px`.

# Introduction to Text Nodes

- Text nodes are parts of the DOM that contain text
- JavaScript can change text nodes to update content dynamically

```
// Using textContent

// textContent sets or gets the text of an element.
// It includes all text, even hidden elements.
let paragraph = document.querySelector('p');
paragraph.textContent = 'This is new text!';


// Using innerHTML

// innerHTML sets or gets the HTML content inside an element.
// It can insert HTML tags and elements.
let div = document.querySelector('div');
div.innerHTML = '<strong>Bold Text</strong>';

// Using innerText

// innerText sets or gets the visible text of an element.
// It only includes visible text, ignoring hidden elements.
let heading = document.querySelector('h1');
heading.innerText = 'New Heading Text';

```

# DOM Exercise #2

BAITUSSALAM
—TECH PARK—

```html
<p id="paragraph">Old text.</p>
<h1 id="heading">Old Heading</h1>
```

Try these:

1. Use `textContent` to change the paragraph text to "New paragraph text."
2. Use `innerText` to change the heading text to "New Heading Text."

# DOM Exercise #3

BAITUSSALAM
—TECH PARK—

```
<div id="content">This is some content.</div>
```

**Try this:**

1. Use `innerHTML` to change the content to include a bold text: "This is bold content."

# Adding Elements to the DOM

Use parentElement.appendChild(newElement) to add the new element to the DOM.

dom.js > ...

```
1  let subtitle = document.createElement('h2')
2  let newText = document.createTextNode('Baitussalam Tech Park!')
3  subtitle.appendChild(newText)
4  document.body.appendChild(subtitle)
5
```

# DOM Exercise # 4

BAITUSSALAM
—TECH PARK—

- Create a new paragraph element, add text, change its style, and add it to a section.
- Given the HTML:

HTML Copy Caption

```html
<section id="content-section"></section>
```

**Try this:**

1. Create a new `<p>` element with the text "Hello, Students!"
2. Set the text color to green and font size to 20px.
3. Append it to the section with the ID `content-section`.

# Date and Time in JavaScript

- JavaScript has a built-in Date object for working with dates and times.
- It allows you to create, manipulate, and format dates and times.

dom.js > ...

```
let currentDate = new Date()
// Mon Jun 10 2024 16:38:58 GMT+0500 (Pakistan Standard Time)

// creating specific date
//                        year-mon-date- time
let birthday = new Date(2010, 5, 15, 12, 30, 0)
// Tue Jun 15 2010 12:30:00 GMT+0500 (Pakistan Standard Time)

// Getting Hours, Minutes, and Seconds
let currentHours = currentDate.getHours() // 16
let currentMinutes = currentDate.getMinutes() // 51
let currentSeconds = currentDate.getSeconds() // 44
```

# JavaScript Date Object Methods

```
14  // get individual date and time components.
15  let year = currentDate.getFullYear() // 2024
16  let month = currentDate.getMonth() // 5
17  let day = currentDate.getDate() // 10
18
19  // Formatting Dates and Times
20  // toLocaleDateString() and toLocaleTimeString() to format dates and times.
21
22  let formattedDate = currentDate.toLocaleDateString() // 6/10/2024
23  let formattedTime = currentDate.toLocaleTimeString() // 4:47:52 PM
24
```

# For Each Loop

- forEach is a built-in array method in JavaScript.
- It allows you to execute a function once for each array element.
- Useful for performing actions on each item in an array.

## Syntax

```
array.forEach(function(element, index, array) {
  // code block
})
```

- `element` : The current element being processed in the array.
- `index` : (Optional) The index of the current element.
- `array` : (Optional) The array `forEach` was called upon.

# setInterval() and setTimeout() methods

- **JavaScript provides setInterval() and setTimeout() methods for executing code at specified intervals or after a delay**
- **They are commonly used for animations, periodic updates, and asynchronous operations.**

# setTimeout() method

**setTimeout()** executes a function once after a specified delay (in milliseconds)

**Syntax: setTimeout(callback, delay)**

dom.js > ...

```
1   setTimeout(() => {
2       console.log('Delayed message')
3   }, 2000) // Executes after 2 seconds
```

# setInterval() methods

**setInterval()** executes a function repeatedly at a specified interval (in milliseconds)

**Syntax: setInterval(callback, interval)**

```
5   let secondsLeft = 10
6   let countdownId = setInterval(() => {
7     console.log(secondsLeft)
8     if (secondsLeft === 0) {
9       clearInterval(countdownId)
10      console.log('Countdown finished!')
11    } else {
12      secondsLeft--
13    }
14  }, 1000)
15
```

# The End

BAITUSSALAM
—TECH PARK—