








LEARN JAVASCRIPT

 Elevate Your JavaScript Skills with
DOM Coding Exercises!  


Exercise JavaScript DOM!

Exercise 1: Change Background Color	1
Detailed Explanation of the Code	3
Exercise 2: Toggle Visibility of an Element	4
Detailed Explanation of the Code	6
Exercise 3: Add New List Item	7
Detailed Explanation of the Code	9
Exercise 4: Changing Image Source	11
Detailed Explanation of the Code	13
Exercise 5: Dynamic Styling of Elements	14

Exercise 1: Change Background Color


 Discover the Power of JavaScript in Web Design!  

In today's video, we delve into an exciting aspect of web development: dynamically changing the background color of a webpage using JavaScript! This simple yet powerful technique showcases how JavaScript can interact with HTML elements to enhance user experience and website interactivity.

 What's Inside:

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>


- We explore how to select HTML elements using JavaScript.
- Understanding the significance of event listeners in JavaScript.
- Implementing conditional logic to alter styles based on user interaction.

 Perfect for Beginners:

Whether you're just starting out in web development or looking to brush up on your JavaScript skills, this video offers a clear and practical demonstration of basic but essential concepts.

 Why This Matters:

- Enhance website interactivity and user engagement.
- Learn the fundamentals of JavaScript, a key tool in web development.
- Unlock the potential for more complex and dynamic web designs.

 Don't forget to like, share, and subscribe for more tips and tutorials on web development and JavaScript!

#JavaScriptBasics #WebDevelopment #CodingTutorial #InteractiveDesign
#WebDesign #FrontEndDevelopment #LearnToCode #Programming
#TechEducation #WebDev #DigitalCreativity

Objective: Write a JavaScript function to change the background color of a webpage.

HTML:

```
<button id="btn1">Change Background Color</button>
```

JavaScript:

```
const btn = document.querySelector('#btn1');
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
console.log(btn);
btn.addEventListener('click',()=>{
  const bg = document.body.style;
  if(bg.backgroundColor!= 'red'){
    bg.backgroundColor = 'red';
  }else{
    bg.backgroundColor = 'white';
  }
})
```

Explanation: This script adds an event listener to a button with the ID changeColorButton. When clicked, it changes the body's background color to light blue.

Detailed Explanation of the Code

The provided JavaScript code snippet is designed to change the background color of a webpage when a button is clicked. Here's a step-by-step explanation:

Selecting the Button:

- `const btn = document.querySelector('#btn1');`
- This line selects the first HTML element with the ID `btn1` and assigns it to the variable `btn`. `document.querySelector` is a method that returns the first element that matches a specified CSS selector in the document.

Logging the Button to Console:

Learn more about JavaScript with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

- `console.log(btn);`
- This line logs the button element to the console. It's a way to verify that the correct element has been selected.

Adding Event Listener to the Button:

- `btn.addEventListener('click', () => { ... })`
- An event listener is added to the button for the 'click' event. This means that the anonymous function (arrow function) provided as the second argument to `addEventListener` will be executed every time the button is clicked.

Defining the Click Event Behavior:

- Inside the arrow function, the code checks and toggles the background color of the page body.
- `const bg = document.body.style;`
 - This line gets the style object of the body element of the document and assigns it to the variable `bg`.
- The if statement checks if the `backgroundColor` of the body is currently 'red'.
- If it is red, it changes the `backgroundColor` to 'white'. If it's not red, it changes the `backgroundColor` to 'red'.

Exercise 2: Toggle Visibility of an Element


🎉 Unlock the Magic of JavaScript: Toggle Elements with Ease! ✨

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Dive into the fascinating world of JavaScript with us today as we explore how to toggle the visibility of webpage elements with just a click! This powerful technique is a fundamental skill for any aspiring web developer and adds a dynamic flair to your web pages.

 In This Video, You'll Learn:

- How to select HTML elements using JavaScript's `querySelector`.
- The power of event listeners in responding to user interactions.
- Implementing conditional logic to toggle the visibility of elements.

 Perfect for All Skill Levels:

Whether you're just starting your coding journey or looking to enhance your web development skills, this tutorial is tailored for easy understanding and practical application.

 Why This Matters:

- Improves user engagement and interactivity on your website.
- Essential technique for dynamic content display in web design.
- Encourages cleaner and more efficient code practices.

Objective: Create a function to toggle the visibility of a paragraph on the page.

HTML:

```
<button id="btn1">Toggle Visibility</button>
<p id="toggleText">This is a paragraph to be shown/hidden.</p>
```

JavaScript:

```
const btn1 = document.querySelector('#btn1');
const p = document.querySelector('#toggleText');
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
btn1.addEventListener('click',()=>{
  p.style.display = p.style.display === 'none' ? 'block' : 'none';
})
```

Explanation: The script toggles the display style of a paragraph between 'none' and 'block', effectively showing and hiding it when the button is clicked.

Detailed Explanation of the Code

The given JavaScript code snippet is a simple yet effective example of how to toggle the visibility of an element on a webpage. Here's a breakdown of what each part of the code does:

Selecting Elements:

- `const btn1 = document.querySelector('#btn1');`
- `const p = document.querySelector('#toggleText');`
- These lines of code select two elements from the HTML document: a button with the ID `btn1` and a paragraph (or any element) with the ID `toggleText`. The `document.querySelector` method is used to find the first element that matches the given CSS selector (ID in this case).

Adding an Event Listener to the Button:

- `btn1.addEventListener('click', () => { ... });`
- This line attaches an event listener to the button (`btn1`). The event listener is set to listen for 'click' events. When the button is clicked, it


Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

triggers the execution of the arrow function provided as the second argument.

Toggling Visibility of the Paragraph:

- Inside the arrow function, the style of the paragraph (p) is toggled.
- `p.style.display = p.style.display === 'none' ? 'block' : 'none';`
- This line uses a ternary operator to check the current display style of the paragraph. If it is set to 'none' (hidden), it changes it to 'block' (visible). Conversely, if it is visible, it sets the display style to 'none', effectively hiding it.

Exercise 3: Add New List Item


 Expand Your Web Development Skills: Dynamic List Manipulation in JavaScript!



Join us in today's tutorial where we dive into an exciting JavaScript feature: dynamically adding new items to a list on a webpage! This practical skill is a cornerstone in interactive web development and is essential for creating responsive user interfaces.

 What You'll Learn:

- How to select HTML elements using JavaScript.
- The process of adding event listeners to respond to user actions.
- Creating and appending new elements to the DOM dynamically.

 Perfect for All Levels:

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Whether you're a beginner or an experienced developer, this tutorial will enhance your understanding of JavaScript's power in creating interactive web pages.

🔥 Why This Is Important:

- Essential for building dynamic and responsive web applications.
- Improves user engagement by allowing interaction with the web page.
- Lays the foundation for more complex JavaScript functionalities.

#JavaScript #WebDevelopment #DynamicContent #UserInteraction

#FrontEndDevelopment #ProgrammingTutorial #CodingSkills #InteractiveWebsites

#WebDesign #TechEducation #LearnToCode #DevCommunity

Objective: Add a new item to an unordered list when a button is clicked.

HTML:

```
<ul id="itemList">
  <li>Item #1</li>
  <li>Item #2</li>
</ul>
<button id="btn1">Add Item</button>
```

JavaScript:

```
const btn = document.querySelector('#btn1');
const itemList = document.querySelector('#itemList');
const curList = itemList.querySelectorAll('li');
let cnt = curList.length;
btn.addEventListener('click',()=>{
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>


```
    cnt++;
    const li = document.createElement('li');
    const txt = document.createTextNode(`New Item #${cnt}`);
    li.appendChild(txt);
    itemList.appendChild(li);
  })
```

Explanation: This script adds a new list item with the text 'New Item' to an unordered list with the ID itemList when a button is clicked.

Detailed Explanation of the Code

The provided code snippet demonstrates a common web development task: dynamically adding new items to a list using JavaScript. Here's an in-depth explanation:

HTML Structure

- The HTML consists of an unordered list () with the ID itemList and initially contains two list items ().
- There is also a button with the ID btn1 which, when clicked, is intended to add new items to the list.

JavaScript Functionality

Selecting Elements:

- `const btn = document.querySelector('#btn1');`: This line selects the button element with the ID btn1.

Learn more about JavaScript with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

- `const itemList = document.querySelector('#itemList');` This line selects the unordered list (``) with the ID `itemList`.

Initial Item Count:

- `const curList = itemList.querySelectorAll('li');` This line selects all existing list items (``) within the `itemList`.
- `let cnt = curList.length;` The length (number) of existing list items is stored in the variable `cnt`.

Adding Event Listener and Creating New List Items:

- `btn.addEventListener('click', () => { ... });` An event listener is added to the button to listen for 'click' events. When the button is clicked, the function inside the event listener is executed.
- Inside the event listener:
 - `cnt++;` The count of list items is incremented.
 - `const li = document.createElement('li');` A new `` element is created.
 - `const txt = document.createTextNode(New Item #${cnt});` A text node is created with the content "New Item #" followed by the current count.
 - `li.appendChild(txt);` The text node is appended to the newly created list item (``).
 - `itemList.appendChild(li);` The new list item (``) is appended to the unordered list (``), effectively adding it to the list on the webpage.

This code demonstrates fundamental JavaScript DOM manipulation techniques: selecting elements, responding to user events, and dynamically modifying the page content.

Exercise 4: Changing Image Source

🌟 Enhance Your Websites with Dynamic JavaScript! 📷 ✨

In our latest tutorial, we dive into a fascinating JavaScript feature: dynamically changing the source of an image with a button click. This technique is a staple in web development, perfect for creating image galleries, sliders, or just adding interactive elements to your site.

👤 What You'll Discover:

- The basics of selecting HTML elements with JavaScript.
- Adding event listeners to handle user interactions.
- Using JavaScript to dynamically change image sources.

🚀 Great for All Skill Levels:

This tutorial is tailored for anyone interested in enhancing their web development skills, from beginners to more experienced developers looking for a refresher on dynamic content manipulation.

📌 Why It's Important:

- A key skill for creating interactive and engaging web pages.
- Understanding this concept opens the door to more advanced JavaScript functionalities.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

- Essential for anyone looking to create visually dynamic websites.

#JavaScript #WebDevelopment #DynamicContent #InteractiveWebDesign
#FrontEndDevelopment #WebDesign #CodingTutorial #TechSkills #LearnToCode
#Programming #DigitalCreativity #WebDevCommunity



Change Image

Objective: Change the source of an image on the webpage.

HTML:

```
  
<button id="btn1">Change Image</button>
```

Javascript

```
const myImage = document.querySelector('#myImage');  
const btn = document.querySelector('#btn1');
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
btn.addEventListener('click',()=>{
  const txt = myImage.src;
  if(txt.substring(txt.length-5) == "1.png"){
    myImage.src = "Images/image2.png";
  }else{
    myImage.src = "Images/image1.png";
  }
})
```

Explanation: The script changes the source of an image to 'image2.jpg' when a button is clicked.

Detailed Explanation of the Code

The code snippet provided is a basic example of how to change the source of an image on a webpage when a button is clicked, using JavaScript. This is a common functionality in web development for creating image sliders or changing content dynamically.

HTML Structure

- The HTML consists of an tag and a <button> tag.
- The tag has an ID myImage and initially displays an image (image1.png located in an Images folder).
- The <button> has an ID btn1 and is labeled "Change Image".

JavaScript Functionality

Selecting Elements:

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

- `const myImage = document.querySelector('#myImage');`: Selects the image element using its ID.
- `const btn = document.querySelector('#btn1');`: Selects the button element using its ID.

Adding an Event Listener to the Button:

- `btn.addEventListener('click', () => { ... });`: Attaches a 'click' event listener to the button. When the button is clicked, the function provided as an argument is executed.

Changing the Image Source:

- Inside the event listener, the code changes the `src` attribute of the image.
- `const txt = myImage.src;`: Gets the current source of the image.
- The `if` statement checks whether the current image source ends with `"1.png"`.
- If it does, the source of the image is changed to `"image2.png"`; otherwise, it's changed back to `"image1.png"`.

This code effectively creates a toggle effect, where clicking the button alternates the image between `"image1.png"` and `"image2.png"`.

Exercise 5: Dynamic Styling of Elements

🌟 Enhance Your Websites with Dynamic JavaScript! 📷 ✨

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

About Us

Paragraph 1

Paragraph 2

This section contains information about the website or company.

Apply Styling

In our latest tutorial, we dive into a fascinating JavaScript feature: dynamically changing the source of an image with a button click. This technique is a staple in web development, perfect for creating image galleries, sliders, or just adding interactive elements to your site.

What You'll Discover:

- The basics of selecting HTML elements with JavaScript.
- Adding event listeners to handle user interactions.
- Using JavaScript to dynamically change image sources.

Great for All Skill Levels:

This tutorial is tailored for anyone interested in enhancing their web development skills, from beginners to more experienced developers looking for a refresher on dynamic content manipulation.

Why It's Important:

- A key skill for creating interactive and engaging web pages.
- Understanding this concept opens the door to more advanced JavaScript functionalities.
- Essential for anyone looking to create visually dynamic websites.

Learn more about JavaScript with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

#JavaScript #WebDevelopment #DynamicContent #InteractiveWebDesign
#FrontEndDevelopment #WebDesign #CodingTutorial #TechSkills #LearnToCode
#Programming #DigitalCreativity #WebDevCommunity

Objective: Apply dynamic styling to paragraphs on button click.

HTML:

```
<section class="main">  
  <h2>About Us</h2>  
  <p>Paragraph 1</p>  
  <p>Paragraph 2</p>  
  <p>This section contains information about the website or  
company.</p>  
  <button id="btn1">Apply Styling</button>  
  
</section>
```

JavaScript:

```
const btn1 = document.querySelector('#btn1');  
const paras = document.querySelectorAll('.main p');  
const arr = ['red', 'blue', 'green', 'purple'];  
let cnt = 0;  
btn1.addEventListener('click', () => {  
  paras.forEach((ele, ind) => {  
    cnt++;
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>


```
    if(cnt%2 == 0){
        ele.style.fontWeight = 'bold';
    }else{
        ele.style.fontWeight = '';
    }
    if(cnt >= arr.length) {cnt = 0;}
    ele.style.color = arr[cnt];
})
})
```

Explanation: This script selects all paragraphs and changes their font color to red and font weight to bold when a button is clicked.

Detailed Explanation of the Code

The code snippet provided is a basic example of how to change the source of an image on a webpage when a button is clicked, using JavaScript. This is a common functionality in web development for creating image sliders or changing content dynamically.

HTML Structure

- The HTML consists of an tag and a <button> tag.
- The tag has an ID myImage and initially displays an image (image1.png located in an Images folder).
- The <button> has an ID btn1 and is labeled "Change Image".

JavaScript Functionality

Selecting Elements:

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

- `const myImage = document.querySelector('#myImage');`: Selects the image element using its ID.
- `const btn = document.querySelector('#btn1');`: Selects the button element using its ID.

Adding an Event Listener to the Button:

- `btn.addEventListener('click', () => { ... });`: Attaches a 'click' event listener to the button. When the button is clicked, the function provided as an argument is executed.

Changing the Image Source:

- Inside the event listener, the code changes the `src` attribute of the image.
- `const txt = myImage.src;`: Gets the current source of the image.
- The `if` statement checks whether the current image source ends with `"1.png"`.
- If it does, the source of the image is changed to `"image2.png"`; otherwise, it's changed back to `"image1.png"`.

This code effectively creates a toggle effect, where clicking the button alternates the image between `"image1.png"` and `"image2.png"`.