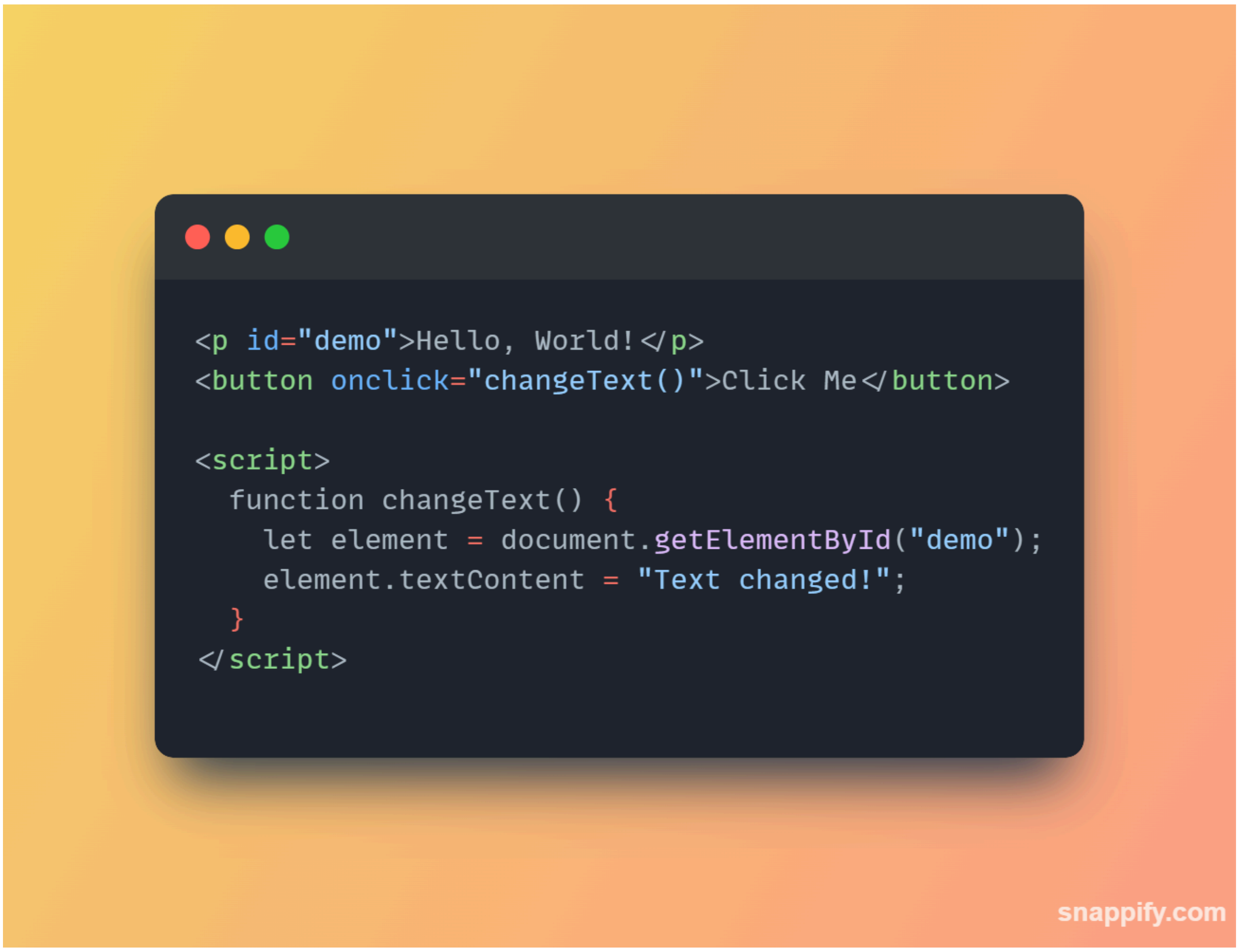
DOM (Document Object Model) Manipulation techniques:

document.getElementById()
Finds an element by its `id` and allows you to modify it.

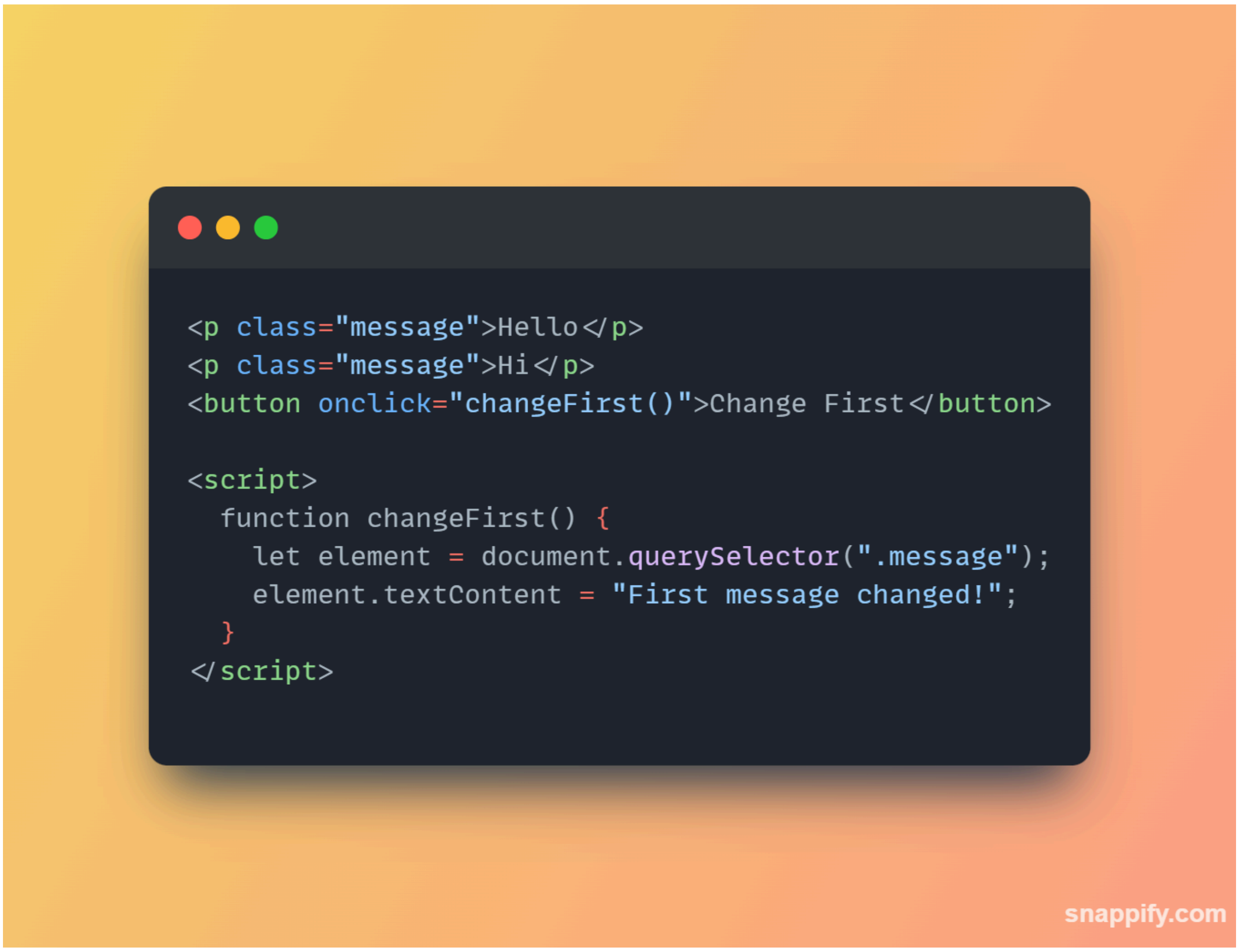


```html
<p id="demo">Hello, World!</p>
<button onclick="changeText()">Click Me</button>

<script>
  function changeText() {
    let element = document.getElementById("demo");
    element.textContent = "Text changed!";
  }
</script>
```

Usage: Finds the element with `id="demo"` and changes its text.

document.querySelector()
Selects the first matching element (by class, id, or tag).

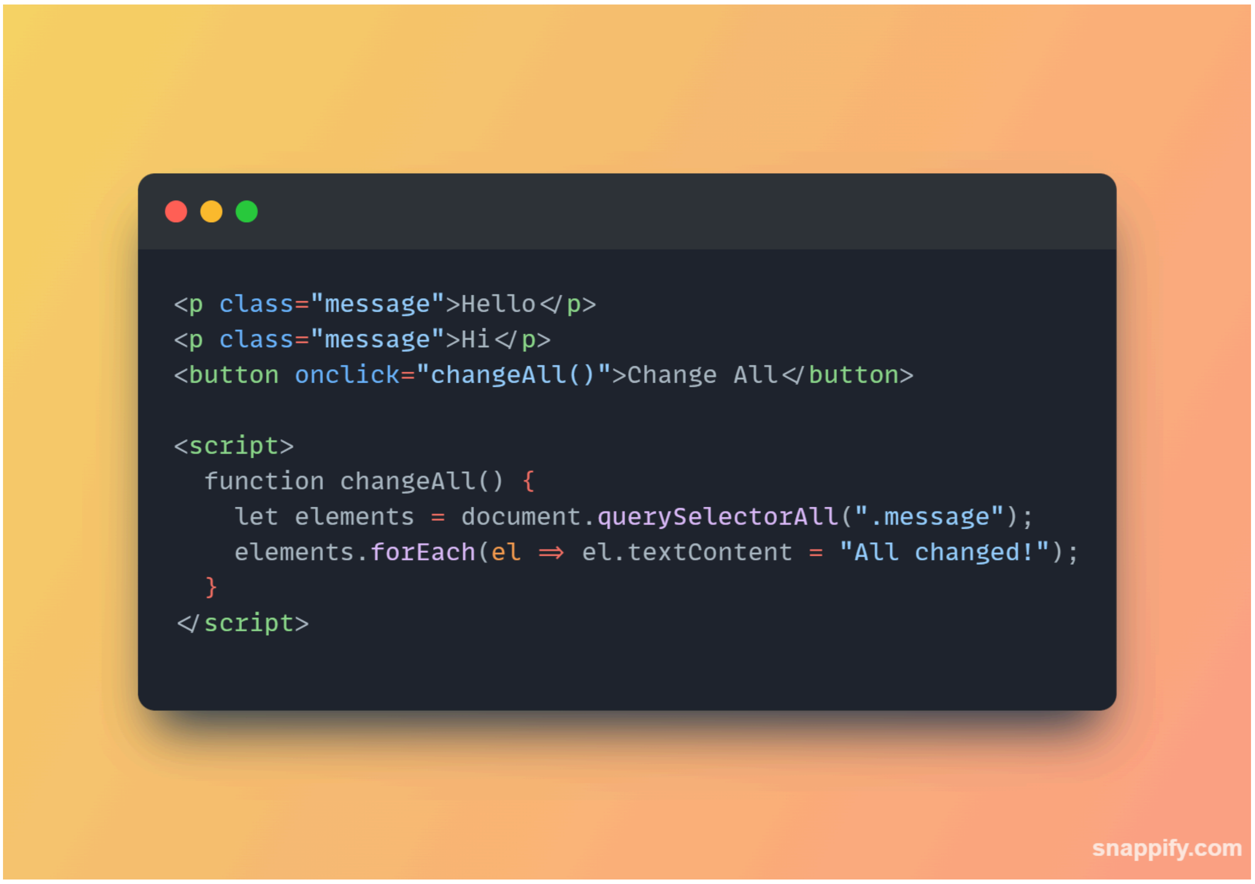


```
<p class="message">Hello</p>
<p class="message">Hi</p>
<button onclick="changeFirst()">Change First</button>

<script>
  function changeFirst() {
    let element = document.querySelector(".message");
    element.textContent = "First message changed!";
  }
</script>
```

Usage: Only the first `<p class="message">` is changed.

document.querySelectorAll()
Selects all elements that match a CSS selector and returns a `NodeList`.

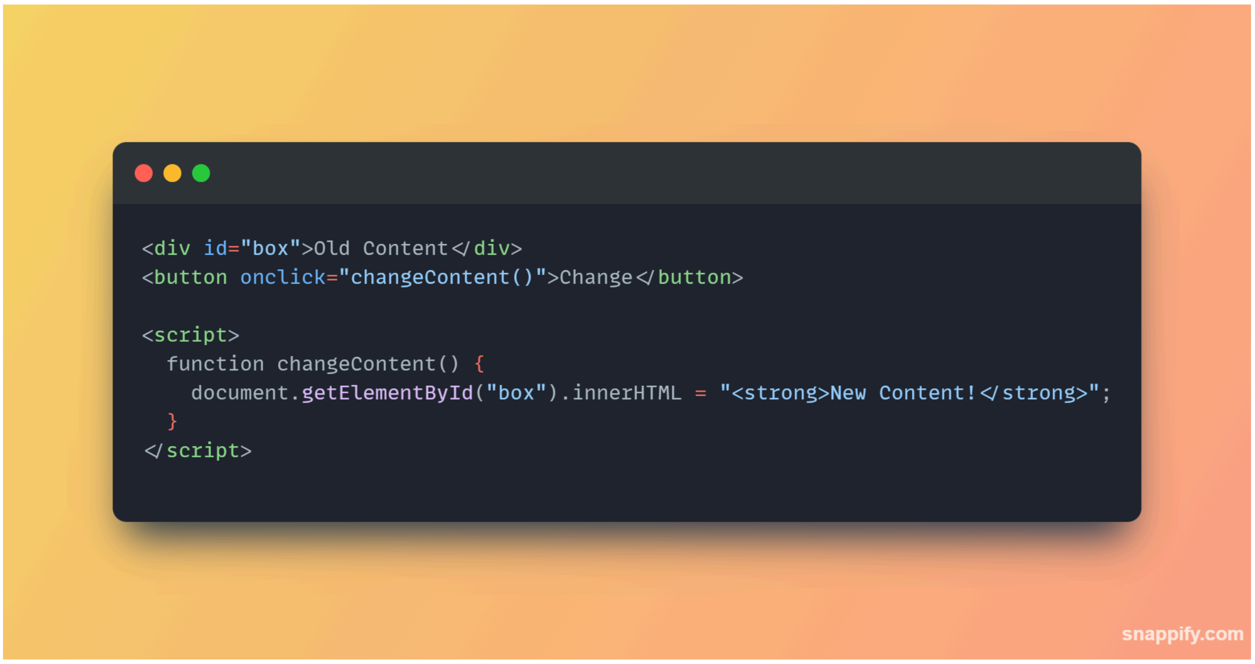


```
<p class="message">Hello</p>
<p class="message">Hi</p>
<button onclick="changeAll()">Change All</button>

<script>
  function changeAll() {
    let elements = document.querySelectorAll(".message");
    elements.forEach(el => el.textContent = "All changed!");
  }
</script>
```

Usage: Changes text in all elements with class `"message"`.

innerHTML
Changes the HTML inside an element.

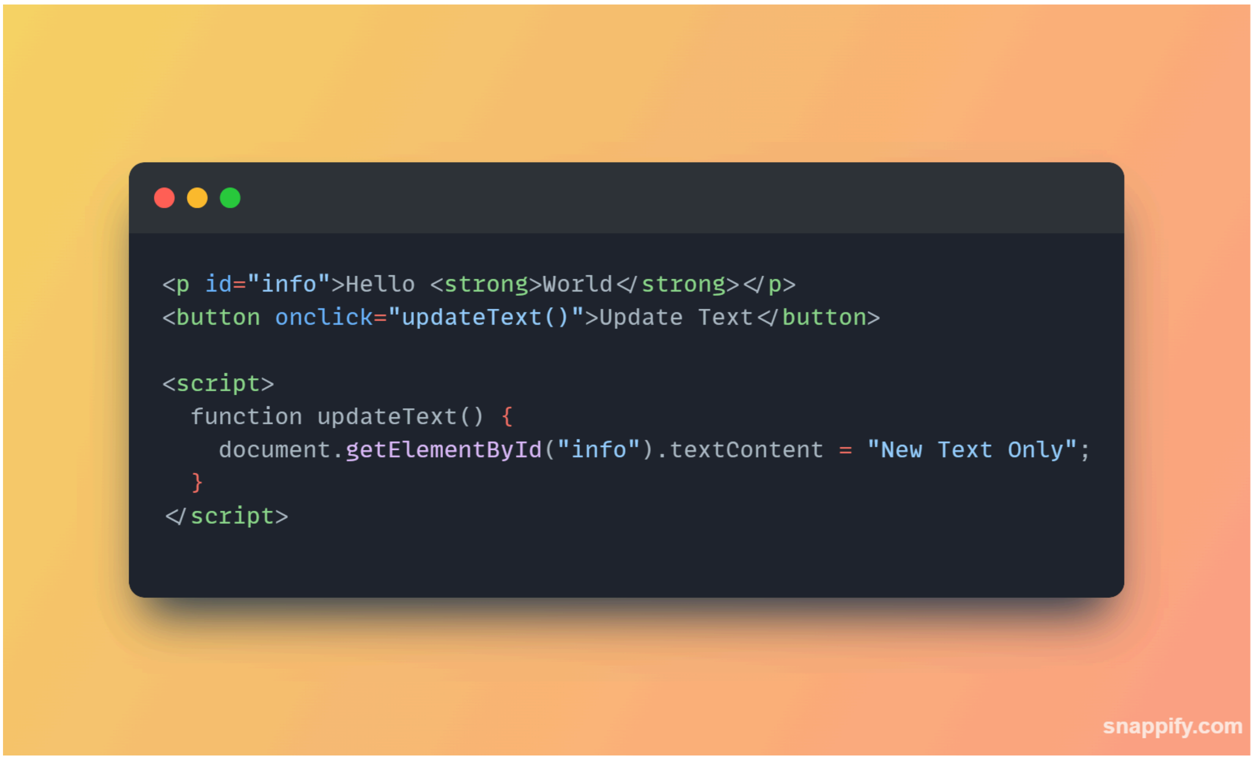


```
<div id="box">Old Content</div>
<button onclick="changeContent()">Change</button>

<script>
  function changeContent() {
    document.getElementById("box").innerHTML = "<strong>New Content!</strong>";
  }
</script>
```

Usage: Modifies the entire inner content of the `<div>`.

textContent
Changes the text inside an element (ignores HTML).

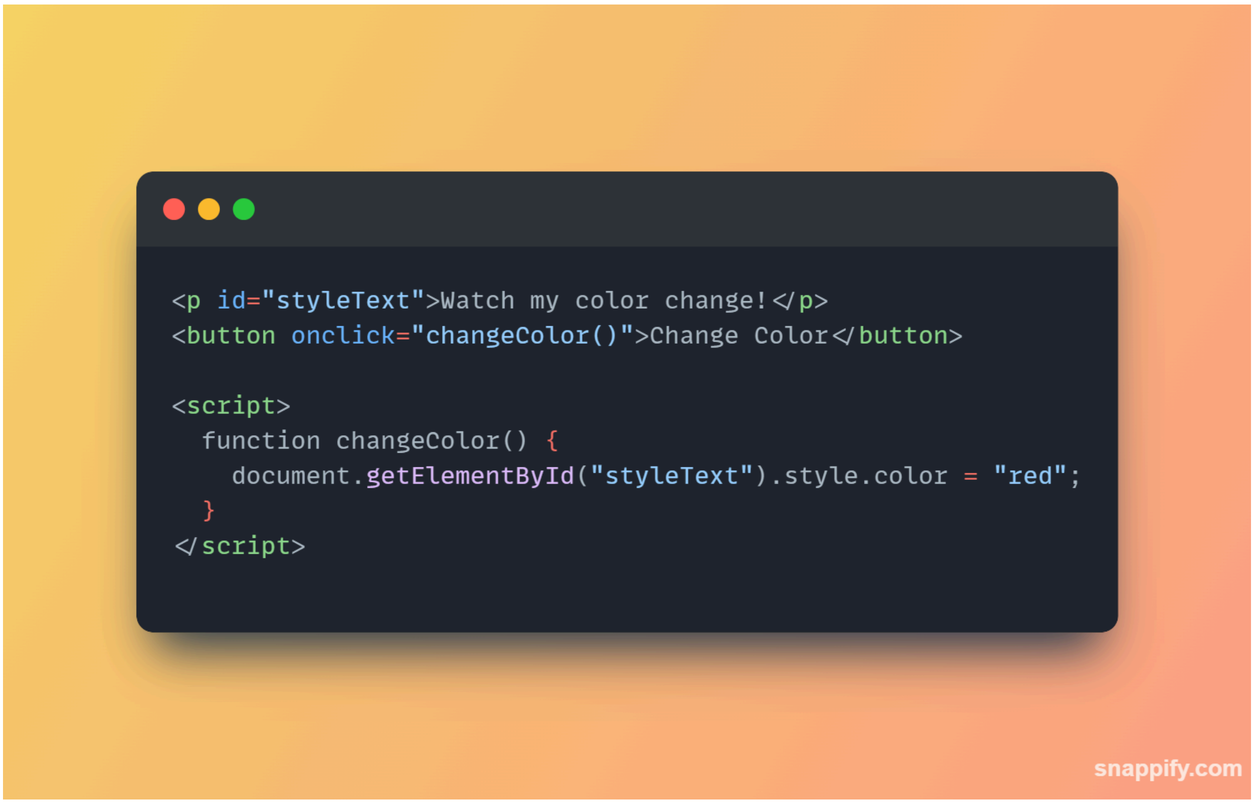


```html
<p id="info">Hello <strong>World</strong></p>
<button onclick="updateText()">Update Text</button>

<script>
  function updateText() {
    document.getElementById("info").textContent = "New Text Only";
  }
</script>
```

Usage: Unlike `innerHTML`, it removes HTML formatting.

`style` (Inline CSS)
Changes the CSS styles of an element.



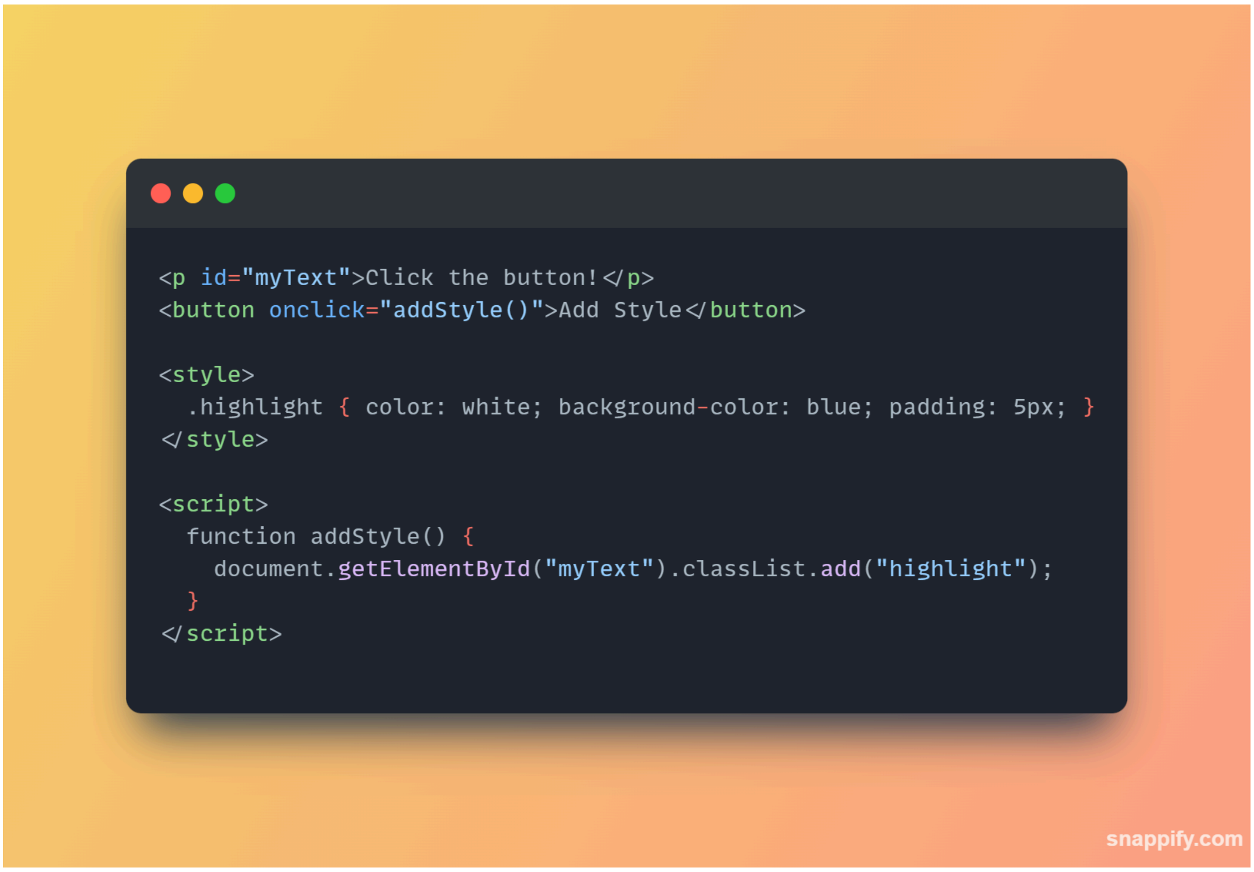
```html
<p id="styleText">Watch my color change!</p>
<button onclick="changeColor()">Change Color</button>

<script>
  function changeColor() {
    document.getElementById("styleText").style.color = "red";
  }
</script>
```

Usage: Directly modifies the CSS property of the element.

`classList.add()` and `classList.remove()`

`Adds or removes CSS classes dynamically.`



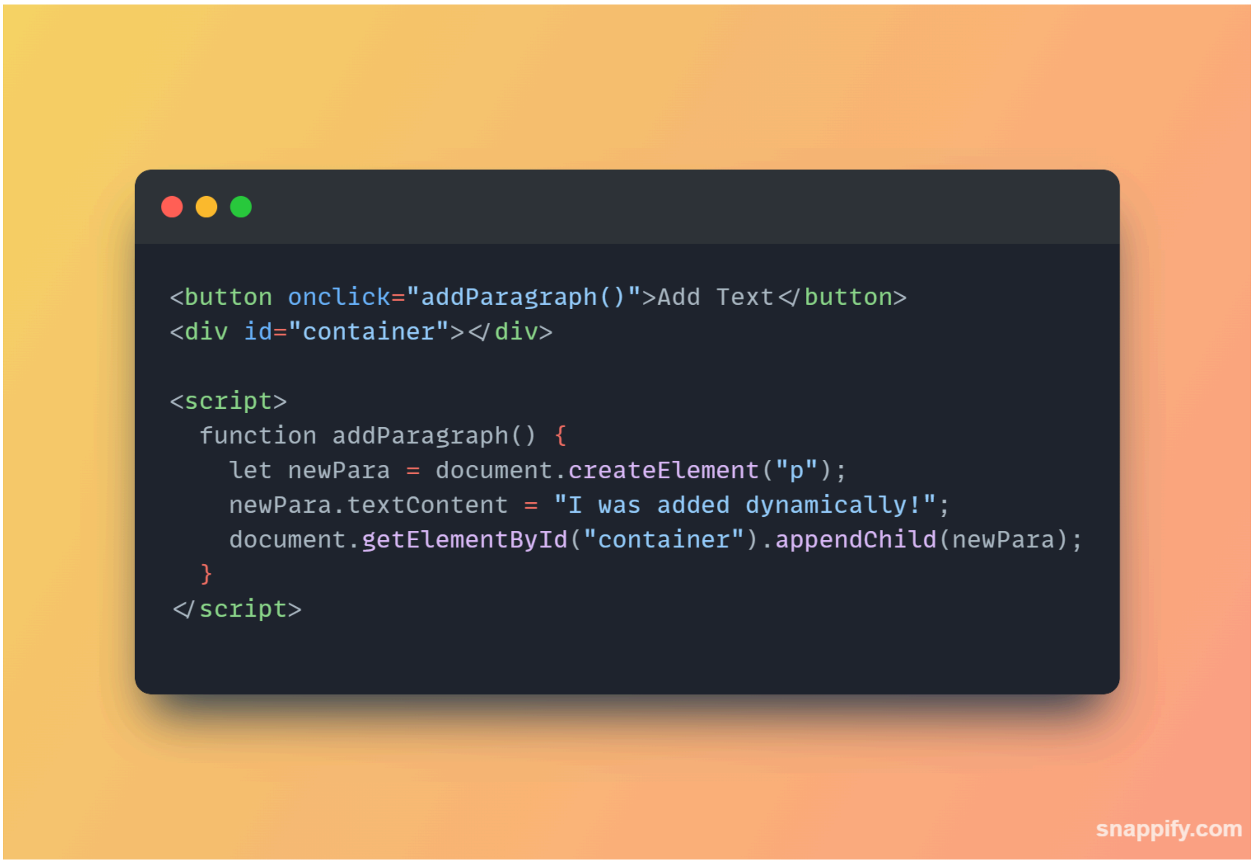
```
<p id="myText">Click the button!</p>
<button onclick="addStyle()">Add Style</button>

<style>
  .highlight { color: white; background-color: blue; padding: 5px; }
</style>

<script>
  function addStyle() {
    document.getElementById("myText").classList.add("highlight");
  }
</script>
```

Usage: `Adds the "highlight" class, applying styles.`

createElement() and appendChild()
Dynamically creates and adds a new element to the DOM.



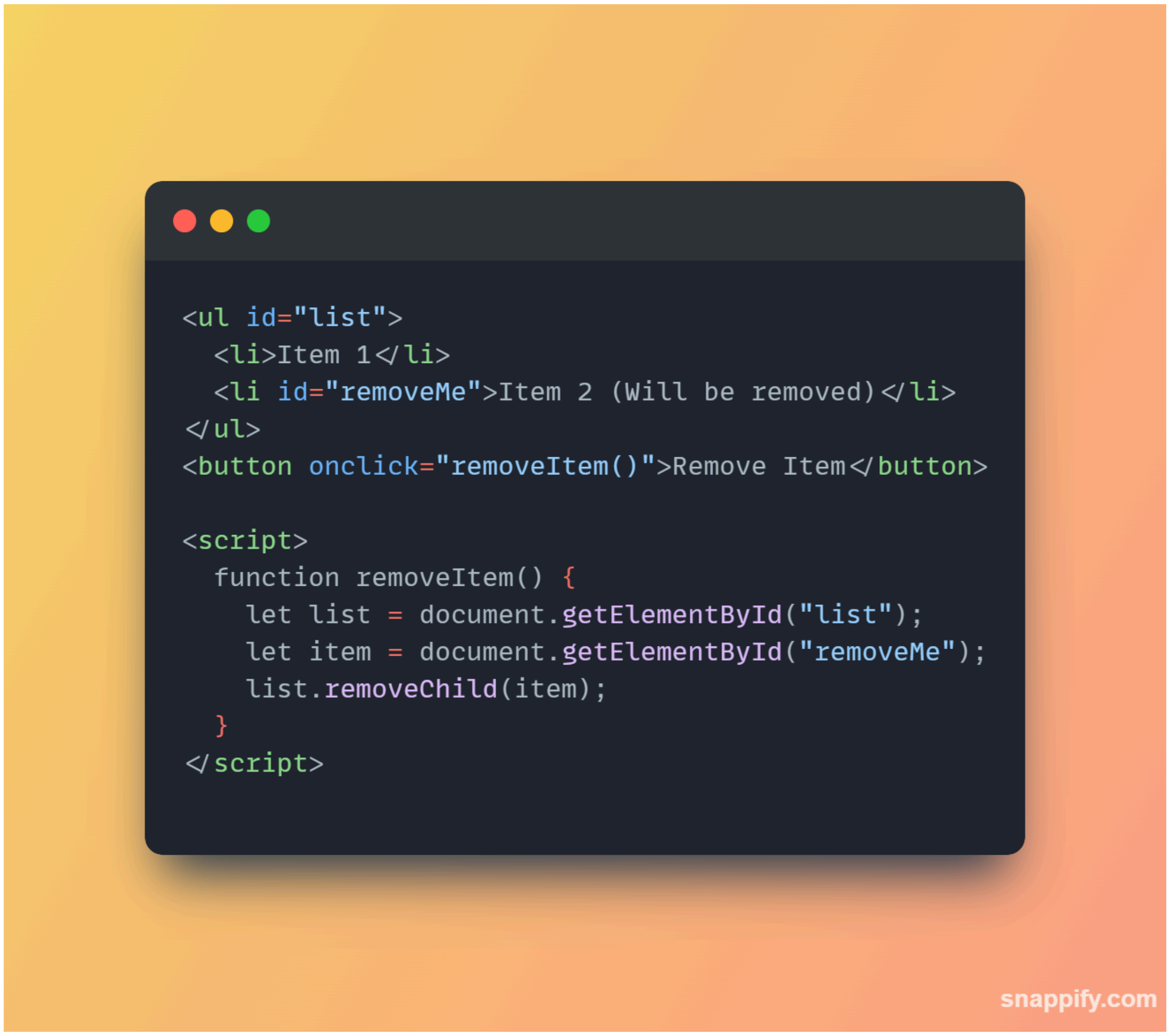
```
<button onclick="addParagraph()">Add Text</button>
<div id="container"></div>

<script>
  function addParagraph() {
    let newPara = document.createElement("p");
    newPara.textContent = "I was added dynamically!";
    document.getElementById("container").appendChild(newPara);
  }
</script>
```

Usage: Creates a new <p> and adds it to the <div>.

removeChild()

Removes an element from the DOM.

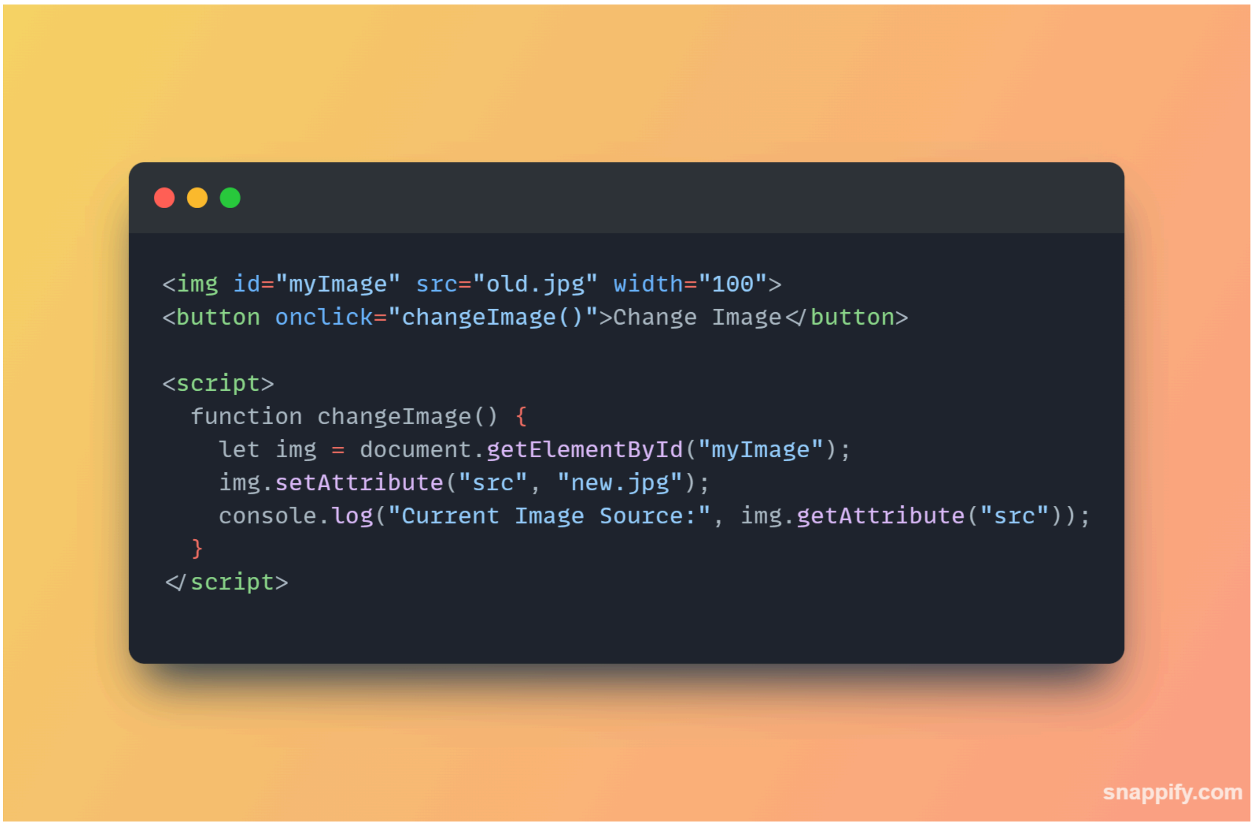


```
<ul id="list">
  <li>Item 1</li>
  <li id="removeMe">Item 2 (Will be removed)</li>
</ul>
<button onclick="removeItem()">Remove Item</button>

<script>
  function removeItem() {
    let list = document.getElementById("list");
    let item = document.getElementById("removeMe");
    list.removeChild(item);
  }
</script>
```

Usage: Removes a specific <li> from the list.

setAttribute() and getAttribute()
Modifies or retrieves an element's attribute.



```
<img id="myImage" src="old.jpg" width="100">
<button onclick="changeImage()">Change Image</button>

<script>
  function changeImage() {
    let img = document.getElementById("myImage");
    img.setAttribute("src", "new.jpg");
    console.log("Current Image Source:", img.getAttribute("src"));
  }
</script>
```

Usage: Updates the src attribute and logs the new value.