append vs appendChild

`appendChild()` and `append()`

When working with JavaScript to manipulate the DOM, it's essential to understand the distinctions between **`appendChild()`** and **`append()`** methods, as they have different behaviors and use cases for adding elements.

- **`appendChild()`** is a method used to append a single element as a child to another element. It takes an element node as its parameter and adds it as the last child of the specified parent element.
- Example:

```
// Creating new elements
var parentElement = document.getElementById("parent");
var childElement = document.createElement("div");


// Appending the child element to the parent element
parentElement.appendChild(childElement);
```

  In this example, the **`appendChild()`** method is used to add the **`childElement`** as the last child of the **`parentElement`**. This method is useful when you want to add a single element to another element.

- **`append()`**, on the other hand, is a more versatile method that allows you to append multiple elements or text to another element. It accepts a variable number of arguments, which can be elements, text strings, or a combination of both.

- Example:

```
// Creating new elements

var parentElement = document.getElementById("parent");

var childElement = document.createElement("div");

var textNode = document.createTextNode("Hello, World!");


// Appending multiple elements to the parent element

parentElement.append(childElement, textNode);
```

In this example, the **`append()`** method is used to add both the **`childElement`** and **`textNode`** to the **`parentElement`**. It can accept any number of elements or text strings as arguments, allowing you to append multiple elements at once.

Usage of Backticks (``) to Add Elements in JavaScript

In JavaScript, backticks (``) are used to create template literals, which provide a concise and powerful way to create strings that can include variables, expressions, and even HTML elements. This functionality proves especially useful when dynamically adding elements or groups of elements to the DOM.

Adding an Element using Backticks:

- Backticks allow you to define an HTML string with embedded variables or expressions. This approach simplifies the process of creating elements and their associated attributes.
- Example:

```
// Creating a new element using backticks
var className = "my-class";
var content = "This is a new element";
```

```
var element = `<div class="${className}">${content}</div>`;
```

In this example, backticks are used to create an HTML string assigned to the **`element`** variable. The string includes the **`className`** variable to dynamically set the class attribute and the **`content`** variable to insert the desired content. This HTML string can then be inserted into the DOM using methods like **`innerHTML`** or by appending it as a child to another element.

Adding a Group of Elements using Backticks:

- Backticks also enable the creation of a group of elements within a single template literal. This technique allows for cleaner and more readable code when adding multiple elements at once.
- Example:

```
// Creating a group of elements using backticks
var elements = `
  <div class="element">Element 1</div>
  <div class="element">Element 2</div>
  <div class="element">Element 3</div>
`;
```

In this example, backticks are used to define a multi-line string representing a group of elements. Each element is enclosed within **`<div>`** tags and has a shared class name. The resulting string, assigned to the **`elements`** variable, can be inserted into the DOM using appropriate methods like **`innerHTML`** or appended to an existing element using **`appendChild()`** or **`append()`**.

Different Ways to add elements in DOM

1. **appendChild() and append() method:**
   The appendChild() and append() method is used to add a new child element to the end of the specified parent element.

2. **insertAdjacentHTML() method:**
   The insertAdjacentHTML() method allows you to insert HTML content at a specific position relative to the element.

3. **innerHTML property:**
   The innerHTML property allows you to get or set the HTML content of an element. You can add new elements as HTML strings and set them as the innerHTML of the parent element.

4. **insertBefore() method:**
   The insertBefore() method can be used to insert an element before a specified reference element.