Arrays & More DOM Manipulations | Cheat Sheet

1. Data Structures

Data Structures allow us to store and organize data efficiently. This makes us access and performs operations on the data smoothly.

In JavaScript, we have built-in Data Structures like,

- Arrays
- Objects
- Maps
- Sets

2. Array

An Array holds an ordered sequence of items.

2.1 Creating an Array

```
let myArray = [5, "six", 2, 8.2];

console.log(myArray);  // [5, "six", 2, 8.2]
```

2.2 Accessing an Array Item

```
let myArray = [5, "six", 2, 8.2];

console.log(myArray[0]);  // 5

console.log(myArray[1]);  // six
```

2.3 Modifying an Array Item

```
let myArray = [5, "six", 2, 8.2];
myArray[1] = 6;

console.log(myArray);  // [5, 6, 2, 8.2]
```

2.4 Finding Array Length

The

`array.length` is used to find the number of items in the array.

```
let myArray = [5, "six", 2, 8.2];
let lengthOfArray = myArray.length;

console.log(lengthOfArray);  // 4
```

2.5 Array Methods

2.5.1 push()

The

`push()` method adds new items to the end of the array.

```
let myArray = [5, "six", 2, 8.2];
myArray.push(true);

console.log(myArray);  // [5, "six", 2, 8.2, true]
```

2.5.2 pop()

The

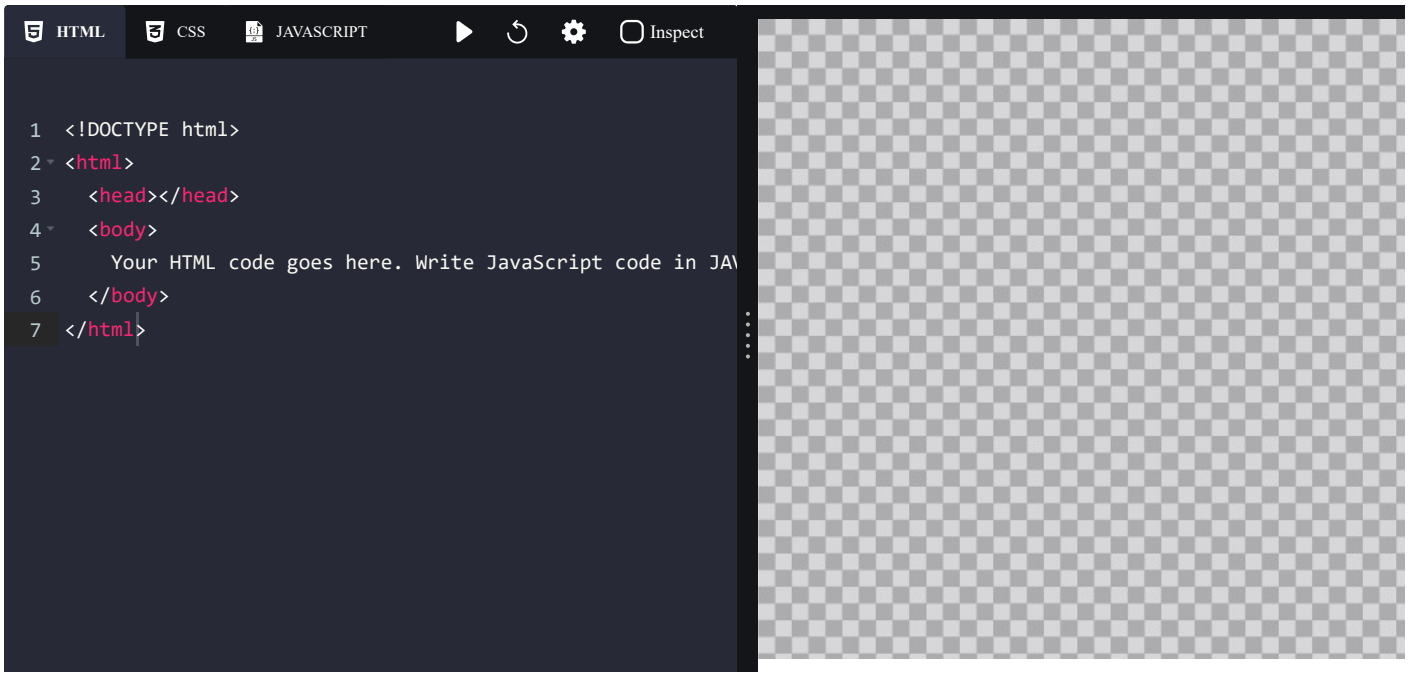
`pop()` method removes the last item of an array and returns that item.

```
let myArray = [5, "six", 2, 8.2];
let lastItem = myArray.pop();

console.log(myArray);  // [5, "six", 2]

console.log(lastItem);  // 8.2
```

Try out creating an array, accessing, modifying its array items, and apply array methods to them in the below Code Playground.



3. Functions

3.1 Function Declaration

```
function showMessage() {
  console.log("Hello");
}

showMessage();
```

3.2 Function Expression

There is another syntax for creating a function which is called Function Expression.

```
let showMessage = function() {
  console.log("Hello");
};

showMessage();
```

4. More DOM Manipulations

4.1 Creating an HTML Element - `createElement()`

```
let h1Element = document.createElement("h1");
h1Element.textContent = "Web Technologies";

console.log(h1Element);  // <h1>Web Technologies</h1>
```

4.2 Appending to an HTML Element - `appendChild()`

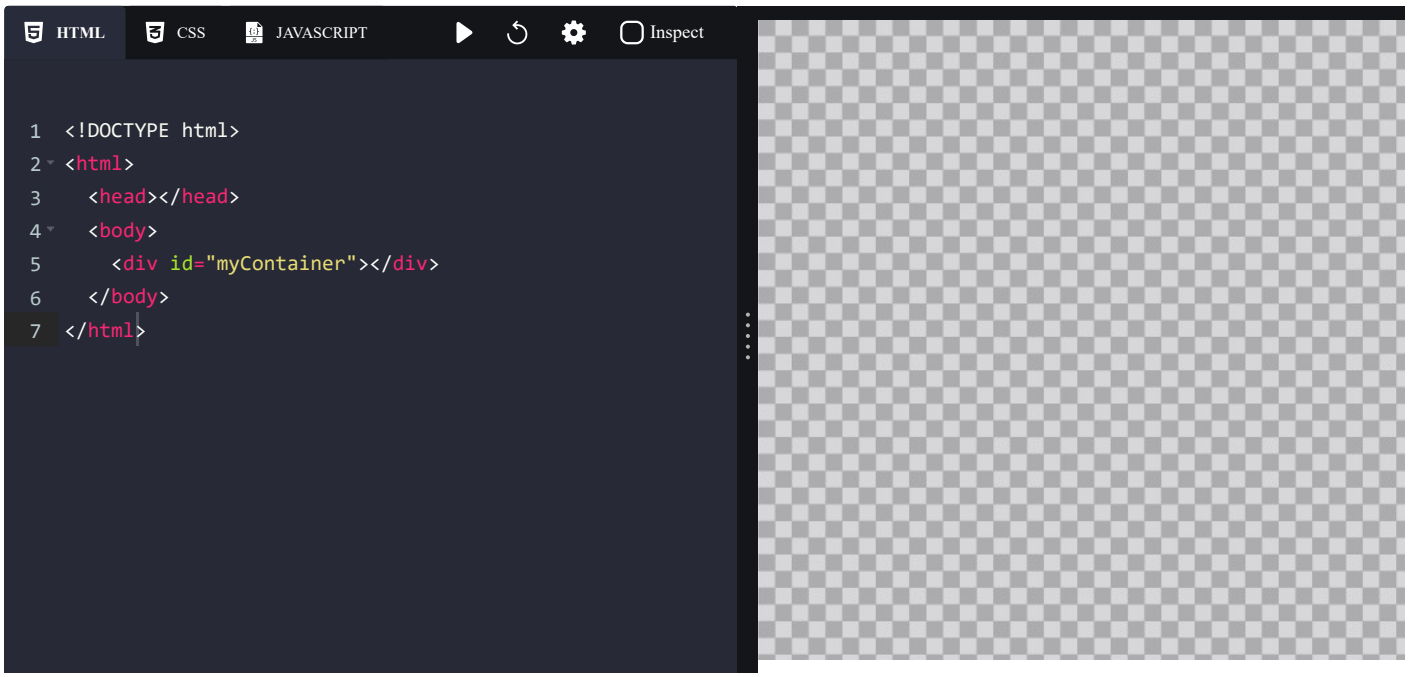
Appending to Document Body Object:

```
document.body.appendChild(h1Element);
```

Appending to Existing Container Element:

```
let containerElement = document.getElementById("myContainer");
containerElement.appendChild(h1Element);
```

Try out creating and appending the HTML elements like a paragraph, image, etc. in the below Code Playground.



4.3 Adding Event Listeners Dynamically

```
let btnElement = document.createElement("button");
```

```
btnElement.textContent = "Change Heading";
document.getElementById("myContainer").appendChild(btnElement);

btnElement.onclick = function(){
  console.log("click event triggered");
};
```

4.4 Providing Class Names Dynamically - classList.add()

```
btnElement.onclick = function(){
  h1Element.textContent = "4.0 Technologies";
  h1Element.classList.add("heading");

  console.log(h1Element);
};
```

```
.heading {
  color: blue;
  font-family: "Caveat";
  font-size: 40px;
  text-decoration: underline;
}
```

4.5 Removing Class Names Dynamically - classList.remove()

```
let removeStylesBtnElement = document.createElement("button");
removeStylesBtnElement.textContent = "Remove Styles";

document.getElementById("myContainer").appendChild(removeStylesBtnElement);

removeStylesBtnElement.onclick = function(){
  h1Element.classList.remove("heading");
};
```

Try out adding the event listeners, class names and removing class names dynamically in the below Code Playground.

HTML CSS JAVASCRIPT Inspect

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <div id="myContainer"></div>
  </body>
</html>
```

Notes Discussions

Notes