

Javascript Code Snippets

JS



Ashfifquzzaman Sajal

Looping through an array:

```
JS index.js

const array = [1, 2, 3, 4, 5];
for (let i = 0; i < array.length; i++) {
  console.log(array[i]);
}
```



Creating a function:



```
function greet(name) {  
  console.log(`Hello, ${name}!`);  
}  
  
greet("John");
```



Adding an event listener:

```
JS index.js

const button = document.querySelector("#myButton");
button.addEventListener("click", () => {
  console.log("Button clicked!");
});
```



Manipulating the DOM:



JS index.js

```
const element = document.getElementById("myElement");  
element.innerHTML = "New content";
```



Making an HTTP request using Fetch API:



JS index.js

```
fetch("https://api.example.com/data")  
  .then(response => response.json())  
  .then(data => console.log(data))  
  .catch(error => console.log(error));
```



Handling errors with try-catch:

```
JS index.js

try {
  // Your code that might throw an error
} catch (error) {
  // Handle the error
  console.Log(error);
}
```



Using arrow functions:



JS index.js

```
const multiply = (a, b) => a * b;  
console.log(multiply(3, 4));
```



Checking if an element exists in an array:



JS index.js

```
const array = [1, 2, 3, 4, 5];  
const exists = array.includes(3);  
console.log(exists); // true
```



Using the spread operator to concatenate arrays:

```
JS index.js

const array1 = [1, 2, 3];
const array2 = [4, 5, 6];
const combinedArray = [...array1, ...array2];
console.Log(combinedArray);
```



Performing asynchronous operations with async/await:

```
JS index.js

async function fetchData() {
  try {
    const response = await fetch("https://api.example.com/data");
    const data = await response.json();
    console.Log(data);
  } catch (error) {
    console.Log(error);
  }
}

fetchData();
```



OTHER PROJECTS

Coming
Soon.....





Ashfifquzzaman Sajal

Web Developer

FOLLOW ME

I need Support

