

JavaScript

Anonymous Functions



1. Basic Anonymous Function:

```
const greet = function(name) {  
  return `Hello, ${name}!`;  
};  
  
console.log(greet("Alice")); // Output: Hello, Alice!
```

2. Anonymous Arrow Function:

```
const greet = (name) => `Hello, ${name}!`;  
  
console.log(greet("Bob")); // Output: Hello, Bob!
```



3. Anonymous Function as a Callback:

Anonymous functions are often used as callbacks, for example in event handlers or array methods.

- **Event Handler:**

```
document.getElementById("myButton").addEventListener("click", function() {
  alert("Button clicked!");
});
```

- **Array Method:**

```
const numbers = [1, 2, 3, 4, 5];
const doubled = numbers.map(function(num) {
  return num * 2;
});

console.log(doubled); // Output: [2, 4, 6, 8, 10]
```



4. Immediate Invocation (IIFE - Immediately Invoked Function Expression): Anonymous

functions can be used to create IIFEs, which run as soon as they are defined.

```
const numbers = [1, 2, 3, 4, 5];
const doubled = numbers.map(function(num) {
    return num * 2;
});

console.log(doubled); // Output: [2, 4, 6, 8, 10]
```



Characteristics

- **No Name:** Anonymous functions do not have a name, making them useful for one-time use or short snippets of code.
- **Assigned to Variables:** They can be assigned to variables, passed as arguments, or returned from other functions.
- **Lexical this (Arrow Functions):** Anonymous arrow functions inherit this from their surrounding context, unlike traditional function expressions.





FOLLOW