Gmail **Asish Kumar Ankam <ankamasish21@gmail.com>**

💡 Do you know about call(), apply(), bind() methods in JS?

1 message

Akshay Saini <team@namastedev.com> Wed, Sep 18, 2024 at 11:37 AM
To: ankamasish21@gmail.com

🔥 call(), apply(), and bind() 🔥

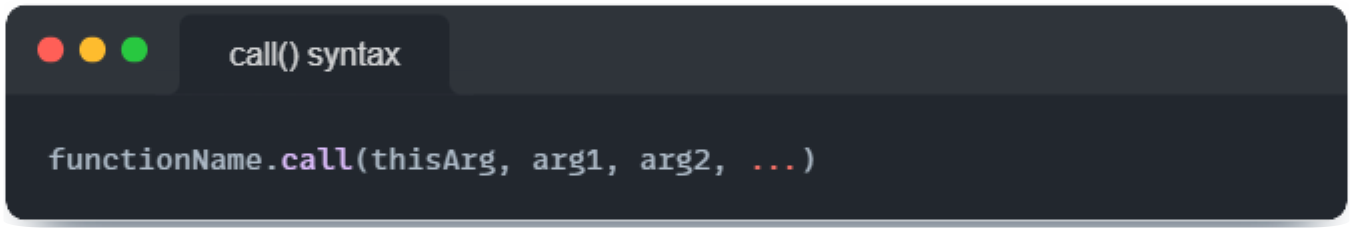
Namaste Asish Kumar

In JavaScript, call(), apply(), and bind() are methods used to control the context (i.e., the value of this) of a function. Let me explain each of them with code examples:

1. call()

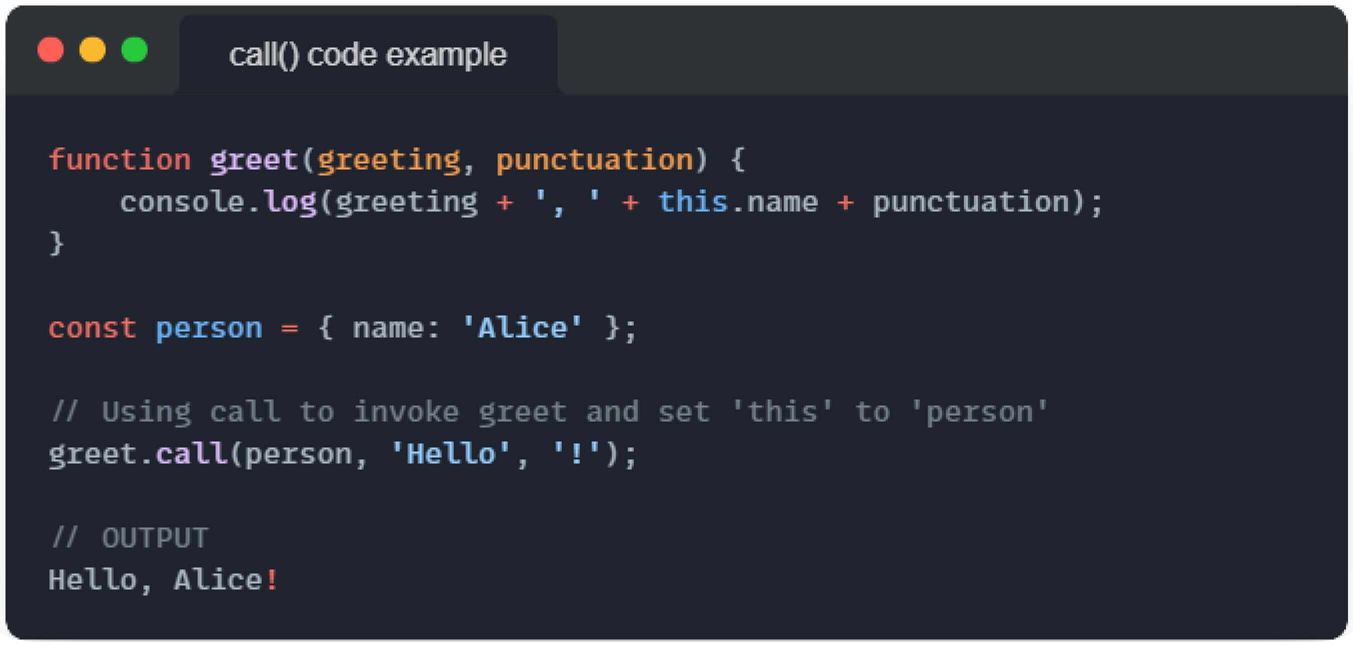
The call() method invokes a function with a given this value and arguments passed individually.

Syntax of call() method:



```
functionName.call(thisArg, arg1, arg2, ...)
```

Example of call() method:



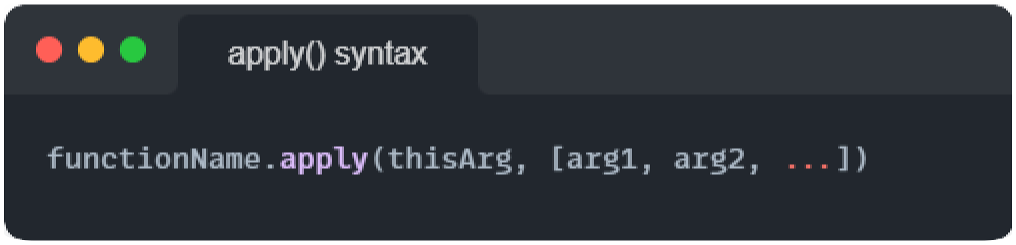
```
function greet(greeting, punctuation) {
    console.log(greeting + ', ' + this.name + punctuation);
}

const person = { name: 'Alice' };

// Using call to invoke greet and set 'this' to 'person'
greet.call(person, 'Hello', '!');

// OUTPUT
Hello, Alice!
```

In this example, this inside the greet function refers to person.

2. apply()

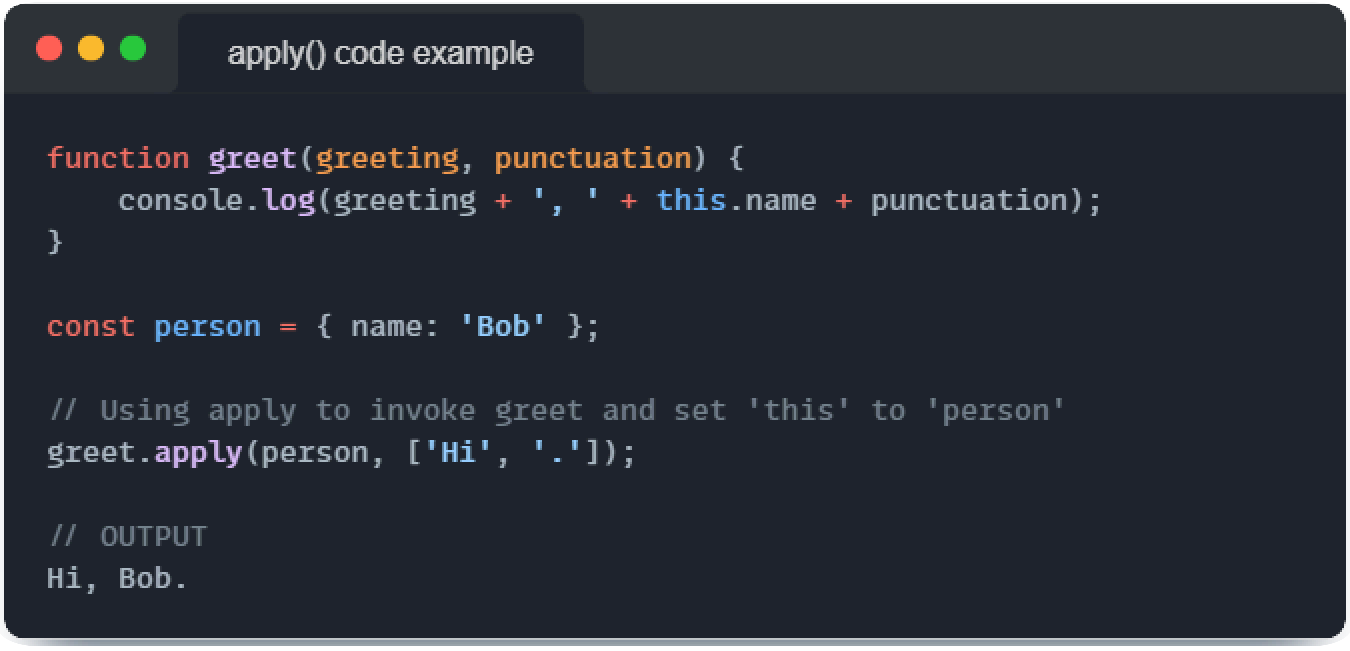
The apply() method is similar to call(), but it accepts an array of arguments rather than listing them one by one.

Syntax of apply() method:



```
functionName.apply(thisArg, [arg1, arg2, ...])
```

Example of apply() method:



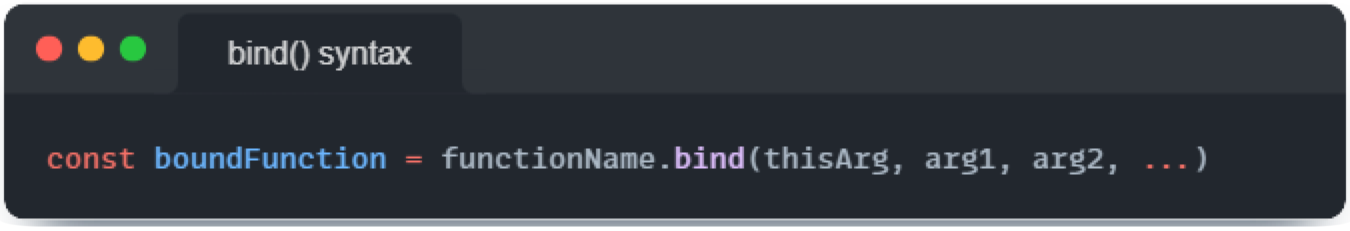
```
function greet(greeting, punctuation) {
    console.log(greeting + ', ' + this.name + punctuation);
}

const person = { name: 'Bob' };

// Using apply to invoke greet and set 'this' to 'person'
greet.apply(person, ['Hi', '.']);

// OUTPUT
Hi, Bob.
```

Here, arguments are passed as an array, unlike call().

3. bind()

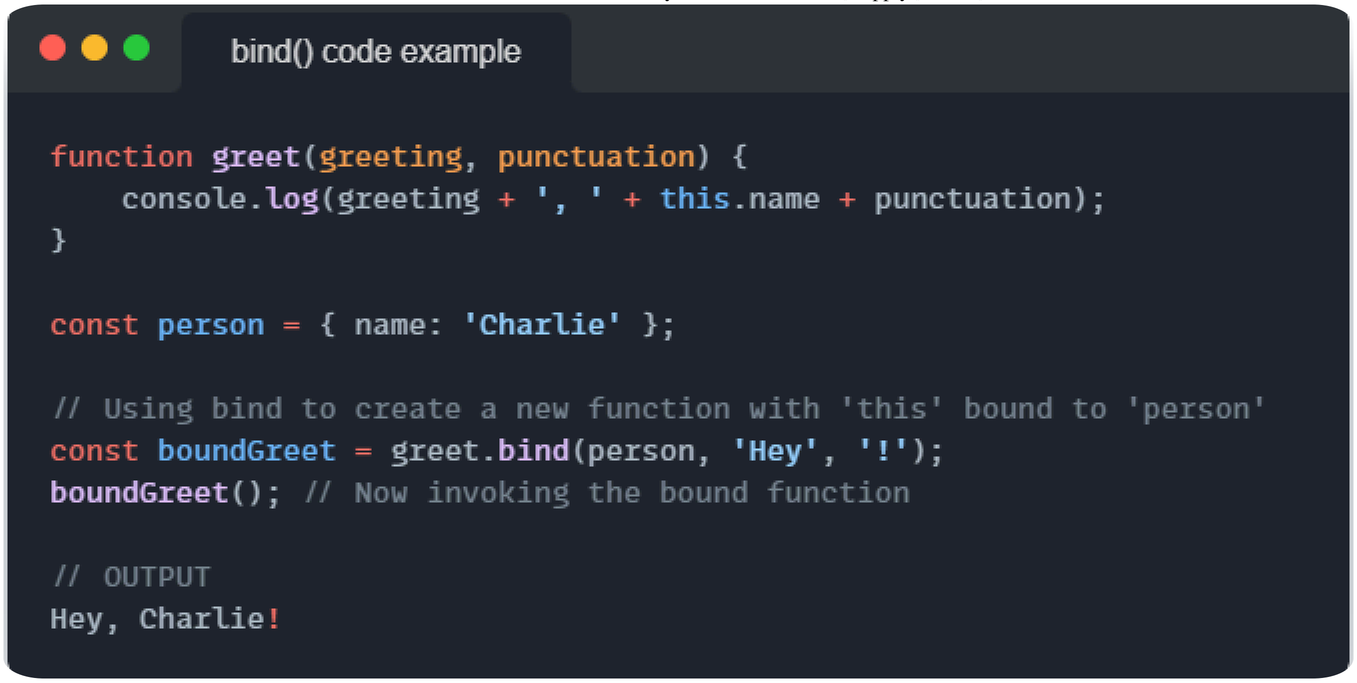
The bind() method returns a new function with a specific this value, allowing you to pass arguments to the function without immediately invoking it.

Syntax of bind() method:



```
const boundFunction = functionName.bind(thisArg, arg1, arg2, ...)
```

Example of bind() method:


bind() code example

```
function greet(greeting, punctuation) {
    console.log(greeting + ', ' + this.name + punctuation);
}

const person = { name: 'Charlie' };

// Using bind to create a new function with 'this' bound to 'person'
const boundGreet = greet.bind(person, 'Hey', '!');
boundGreet(); // Now invoking the bound function

// OUTPUT
Hey, Charlie!
```

With bind(), the function greet is not immediately executed. Instead, a new function is created with this bound to person, and it's invoked later.

SUMMARY

- **call()**: Immediately invokes a function with the this value and arguments passed individually.

- **apply()**: Immediately invokes a function with the this value, but arguments are passed as an array.

- **bind()**: Returns a new function where this is bound, and it can be invoked later.

These methods are particularly useful when borrowing methods from other objects or setting the context dynamically.

If you want to explore more such interview topics in detail, then checkout the **FREE COURSE** offered by NamasteDev below!

🔥 Enroll in Free Course