

# 1. Call()

The `call()` method binds the `this` value to the function and executes the function. It takes the `this` value and a list of arguments as parameters. It returns the value returned by the function, which is called using the call method.



```
1  function greetings() {
2    console.log(`Hello, ${this.name}! How are you doing today?`);
3  }
4
5  const person1 = { name: 'John Smith' };
6  const person2 = { name: 'Mark Rober' };
7
8  greetings.call(person1);
9  // Output: Hello, John Smith! How are you doing today?
10
11 greetings.call(person2);
12 // Output: Hello, Mark Rober! How are you doing today?
```

## 2. Apply()

The `apply()` method binds the `this` value to the function and executes the function. It takes the `this` value and a single array-like object as its second argument, and it returns the value returned by the function, which is called using the `apply` method.



```
1  function greetings(...greetMessage) {
2      console.log(`${greetMessage[0]}, ${this.name}! ${greetMessage[1]}
3      ${greetMessage[2]} ${greetMessage[3]} ${greetMessage[4]}?`);
4  }
5
6  const person1 = { name: 'John Smith' };
7  const person2 = { name: 'Mark Rober' };
8
9  greetings.apply(person1, ['Hello', 'How', 'are', 'you', 'doing']);
10 // Output: Hello, John Smith! How are you doing today?
11
12 greetings.apply(person2, ['Hello', 'How', 'are', 'you', 'doing']);
13 // Output: Hello, Mark Rober! How are you doing today?
```

# 3. Bind()

The `bind()` method is used to create a new function with a specified `this` value, without invoking the original function immediately. This is useful for creating functions with a fixed `this` value. However, we still need to separately invoke the returned function.



```
1  function greetings() {  
2    console.log(`Hello, ${this.name}! How are you doing today?`);  
3  }  
4  
5  const person1 = { name: 'John Smith' };  
6  const person2 = { name: 'Mark Rober' };  
7  
8  const introducePerson1= greetings.bind(person1);  
9  introducePerson1() // Output: Hello, John Smith! How are you doing today?  
10  
11 const introducePerson2= greetings.bind(person2);  
12 introducePerson2() // Output: Hello, Mark Rober! How are you doing today?
```

# Advantages

`Call()` and `Apply()` are useful when you want to call a function with a specific ``this`` context for a single invocation. `Bind()` is more commonly used when you need to create a new function with a fixed ``this`` context, which can be called multiple times.

These methods are powerful tools for managing the ``this`` context and passing arguments when working with JavaScript functions and are commonly used in scenarios like event handling, function composition, and object-oriented programming.