Understanding call, apply, and bind methods in JavaScript

What is call?

call invokes a function with a given this value and arguments provided individually.

Syntax: functionName.call(thisArg, arg1, arg2, ...)

```
function greet(greeting, punctuation) {
  console.log(greeting + ', ' + this.name + punctuation);
}
const person = { name: 'Alice' };
greet.call(person, 'Hello', '!');
// Output: Hello, Alice!
```

What is apply?

apply is similar to call, but arguments are provided as an array.

Syntax: functionName.apply(thisArg, [argsArray])

```
function greet(greeting, punctuation) {
  console.log(greeting + ', ' + this.name + punctuation);
}
const person = { name: 'Bob' };
greet.apply(person, ['Hi', '...']);
// Output: Hi, Bob...
```

What is bind?

bind creates a new function that, when called, has its this keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called.

Syntax: const newFunction = functionName.bind(thisArg, arg1, arg2, ...)

```
function greet(greeting, punctuation) {
  console.log(greeting + ', ' + this.name + punctuation);
}
const person = { name: 'Charlie' };
const greetPerson = greet.bind(person, 'Hey');
greetPerson('!!');
// Output: Hey, Charlie!!
```

Will you give this a try?

Leave a comment below