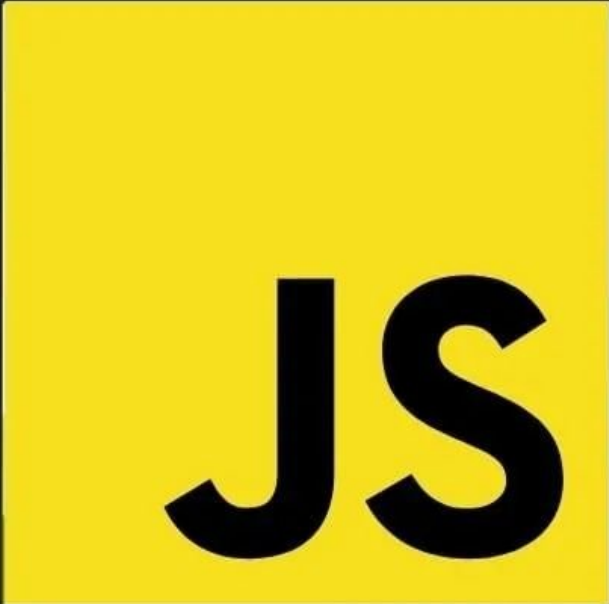


call(), apply() bind()

Functions | In Detail

A yellow square containing the letters 'JS' in a bold, black, sans-serif font, representing JavaScript.

JS

Mallikarjun | @CodeBustler



call() Function

- The **call** method is used to invoke a function with a specified "**this**" value and a list of arguments (parameters).
- **Syntax:**

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

The first argument (**thisArg**) specifies the value of "**this**" within the function when it is invoked

call() Function | Example

```
const person = {
  fullName: function (location, rollNo) {
    return `${this.firstName} ${this.lastName}
            from ${location}, RollNo ${rollNo}`;
  }
};

const person1 = {
  firstName: "Arjun",
  lastName: "Mavnoor"
};
const person2 = {
  firstName: "CodeBustler",
  lastName: "XYZ"
};

console.log(person.fullName.call(person1, "Bangalore", 32));
// Arjun Mavnoor from Bangalore, RollNo 32

console.log(person.fullName.call(person2, "Hyderabad", 54));
// CodeBustler XYZ from Hyderabad, RollNo 54
```

apply() Function

- The apply method is similar to call, but it takes an array-like object of arguments instead of individual arguments.
- **Syntax:**

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



array-like object
of arguments

apply() Function | Example

```
const person = {
  fullName: function (location, rollNo) {
    return `${this.firstName} ${this.lastName}
            from ${location}, RollNo ${rollNo}`;
  }
};

const person1 = {
  firstName: "Arjun",
  lastName: "Mavnoor"
};

const person2 = {
  firstName: "CodeBustler",
  lastName: "XYZ"
};

// APPLY
console.log(person.fullName.apply(person1, ["Bangalore", 32]));
// Arjun Mavnoor from Bangalore, RollNo 32

console.log(person.fullName.apply(person2, ["Hyderabad", 54]));
// CodeBustler XYZ from Hyderabad, RollNo 54
```

bind() Function

- The bind method is used to create a new function that, when invoked, has a specified this value and optional arguments.
- **Syntax:**

```
function.bind(thisArg, arg1, arg2, ...)
```



create a new
function

bind() Function | Example

```
const person = {
  fullName: function (location, rollNo) {
    return `${this.firstName} ${this.lastName}
      from ${location}, RollNo ${rollNo}`;
  }
};

const person1 = {
  firstName: "Arjun",
  lastName: "Mavnoor"
};

const person2 = {
  firstName: "CodeBustler",
  lastName: "XYZ"
};

// BIND | Returns Functions
let p1Detail = person.fullName.bind(person1, ["Bangalore", 32]);

let p2Detail = person.fullName.bind(person2, ["Hyderabad", 54]);

console.log(p1Detail());
// Arjun Mavnoor from Bangalore, RollNo 32

console.log(p2Detail());
// CodeBustler XYZ from Hyderabad, RollNo 54
```

Important bullet points

- **call** and **apply** set “**this**” and pass arguments to a function;
 - **call** uses individual args,
 - **apply** uses an array of args.
- **bind** creates a **new function** with a fixed this & optional args, without immediate execution.
- **call** and **apply** invoke functions **immediately**;
bind allows delayed execution.

Other Posts

JavaScript Event Loop

In Detail



Mallikarjun | @CodeBustler



JS Event Delegation

In Detail



Mallikarjun | @CodeBustler



To learn **HTML, CSS, JavaScript & React JS** or Front-End **Projects**, Visit my page all concepts & Projects codes are available 🔥

@CodeBustler

Did You Find it Useful?



Alamin CodePapa
@CodePapa360

Follow for more

Like | Comment | Repost

