@thekrprince

Call, Apply & Bind method in JS

Call, **Apply** and **Bind**.
These are the predefined methods in JavaScript which is used to **call/borrow** the function using the different **"this" reference** and **argument**.

```
------------------------------Code 1------------------------------
let person = {
      firstName: "Kumar",
      lastname: "Prince",
      personName: function(){console.log(
          this.firstName + " " + this.lastName);
      }
}
person.personName();    //op - Kumar Prince
--------------------------------------------------------------------
```

As you can see, I created a object with 2 variable and 1 function and then calling it but what if we have another object which needs the same function? We could simply copy paste the function but would that be a ideal way? No. So that's where **call**, **apply** and **bind** methods come in the picture.

Call method

Using call() method we can call the another function with given **this** value and arguments.

```
------------------------------Code 2------------------------------
let person2 = {
      firstName: 'John',
      lastName: 'Snow',
};
person.fullName.call(person2);  //op - John Snow
----------------------------------------------------------------------
```

What this last line of code is doing that rather than pointing to **person's** variable in **Code 1**, it's pointing to **person2 reference** and printing the John Snow as output.

Generally, when we want to reuse the function, we don't keep it inside the object rather than we create it separately and then call it. **Eg. below**

-------------------------------Code 3-------------------------------

```
let personWork = function (jobTitle, place) {
      console.log(this.firstName + ' ' +  this.lastName + '
works as a ' + jobTitle + ' in ' + place);
};
let person3 = {
      firstName: 'J.K',
      lastName: 'Rowling',
};
personWork.call(person3, 'Writer', 'UK');     //op: J.K
Rowling works as a Writer in UK
```

--

The first arguments will always be the reference to "this" reference as "person3" in above code example and then the arguments a/c to the function parameter.

Apply method

apply() method also functions same as **call()** method, the only difference is that while using apply method we have to pass the arguments in an array rather than individually.

------------------------------Code 4------------------------------

```
let personWork = function (jobTitle, place) {
      console.log(this.firstName + ' ' + this.lastName + '
works as a ' + jobTitle + ' in ' + place);
};
let person4 = {
      firstName: 'J.K',
      lastName: 'Rowling',
};
personWork.call(person3, ['Writer', 'UK']); //op: J.K
Rowling works as a Writer in UK
```

--

Bind method

bind() method is same as **call()** method, the only difference here is that rather than calling the function, it returns the function which we can store in a variable and then call that function.

-----------------------------Code 5-----------------------------

```
let person5 = {
      firstName: 'Harry',
      lastName: 'Potter',
};

let res = personWork.bind(person5, 'Wizard', 'Hogwarts');
res();

//op: Harry Potter works as a Wizard in Hogwarts.
```

--

Thank you!

Hope you find it helpful.

Follow **@thekrprince** for more
Web Dev related content