Page No.
Date

* Call, Apply and Bind methods in javascript

we know that functions are also objects in javascript, so these 3 methods call, apply and bind are used to control the invocation of the function.
call and apply were introduced in ECMAScript 3 while bind() was added as part of ECMAScript 5.

Uses :-

you can use call/apply to invoke function immediately.
bind() returns a bound function that, when executed later, will have the correct context ("this") for calling the original function.
so bind() can be used when the function needs to be called later in certain events when it's useful.

① call() Method

The call() method is used to call a function with a given "this" and arguments provided to it individually.
This means that we can call any function, explicitly specifying the reference that "this" should reference in the calling function.

example :-

```
let name = {
    firstName: "Sonu",
    lastName: "Tekane",
};
let printName = function (hometown, state) {
    console.log( this.firstName + " " +
                 this.lastName + " from " +
                 hometown + ", " + state);

printName.call (name, "Bhenda", "Maharashtra");
```

⇓

Sonu Tekane from Bhenda, Maharashtra

The first parameter in call() method sets the "this" value, which is the object, on which the function is invoked upon.
In our example above,
it's the "name" object
The rest of the parameters are the arguments to the actual function.

② apply() Method

The apply method is an important method of the function prototype and is used to call other functions with a provided "this" keyword value and arguments provided in the

form of array or an array like object.
Array-Like objects provided may refer to NodeList or the arguments object inside a function.
This means that we can call any function and explicitly specify what "this" should reference in the calling function.

example :-

```
let name = {
    first : "Sonu",
    last : "Tekane",
};
let printName = function (hometown, state) {
    console.log( this.first + " " + this.last
                + " from " + hometown + " , " +
                state);
printName.apply (name, [" Bhenda", "Maharashtra"]);
```

⇓

Sonu Tekane from Bhenda, Maharashtra

similarily to call method the first parameter in apply() method sets the "this" value which is the object upon which the function is invoked.
In our example above,
it's the "name" object

The only difference of apply() with the call() method is that the second parameters of the apply() method accepts the arguments to the actual functions as an array.

imp

③ bind() Method

The bind() method 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.

The bind function is much like the call function, with the main difference being that bind returns a new function whereas call does not.

According to ECMAScript 5 specifications, the function returned by bind() is a special type of exotic function object (as they call it) called the Bound Function. (BF)

The BF wraps the original function object. Calling a BF runs the wrapped function in it.

[simply we can say that, it gives a copy which can be invoked later as per user's need.]

example :-

```
let name = {
    first : "sonu",
    last : "Tekane",
};
let printName = function ( hometown , state) {
    console.log ( this.first + " " + this.last
                + " from " + hometown + ", "
                + state);
let printMyName = printName.bind (name, "Bhenda",
                                  "Maharashtra");
console.log (printMyName);
printMyName ();
```

gives function

we can call like this whenever we want

```
f (hometown, state) {
    conso
}
```

Sonu Tekane from Bhenda, Maharashtra

Summary :- ① call method is used to invoke a function directly.

② apply is same as call, only difference is, it takes second argument as array list.

③ bind method does not directly invoke the method but gives you copy of the exactly same method which can be invoked later.