

POLYFILL FOR BIND METHOD

```
Function.prototype.mybind = function(...args) {  
  let obj = this  
  params = args.slice(1);  
  return function(...args2) {  
    obj.apply(obj, [...params, ...args2]);  
  }  
}
```

The above is the polyfill for bind method.

Polyfill:

To make older browsers understand our new code, the new code is converted into an older code which browsers can understand called polyfill.

Explanation

Our task is to create own implementation of **bind** method.

★ Traditional bind function :-

- let **name** = {
 firstname : "Ashraya",
 lastname : "Devan"
}
- let **printName** = function () {
 console.log (this.firstname + " " + this.lastname);
}
- let **printmyName** = **printName**.**bind**(**name**);
- **printmyName**();

★ Our task is to create our own implementation of **bind** method.

Something like : **mybind**() { }
which behaves exactly like **bind** method.

- let **printmyName2** = **printName**.**mybind**(**name**);
printmyName2();

★ Observe **bind** method carefully once :-

① Every function in javascript has access to this **bind** method.

So, every function should also have access to mybind method.

The way to do this is "to put mybind function in Function.prototype

NOTE:-

If we write any method in 'Function.prototype' then every methods we write have access to those methods.

② We get a function in return that can be later invoked using **bind** method.

③ If we call **printMyName2**, then **printName** method should be executed.

We can get this using this variable.

Store this keyword in an object

Call this object by passing args[0] as the reference, which is the name.

So, basic implementation of 'mybind' method:-

```
Function.prototype.mybind = function(...args) {  
  let obj = this  
  return function() {  
    obj.call(args[0]);  
  }  
}
```

④ But there is problem with the arguments

If we have another parameter inside the printName function, we have to extract it

params = args.slice(1)

→ This will eliminate first element & return the later all elements

⑤ Use apply method to pass params as array.

obj.apply(args[0], params);