

Apply Method :- اول استخدام في عمل call لل function وبتاخذ 2 arguments وهم this العائدة علي ال object وال array of function arguments زي في المثال :-

```
1  const person = {
2    firstName: "John",
3    LastName: "doe",
4  };
5
6  function greet(greeting, message) {
7    return `${greeting} ${this.firstName}. ${message}`;
8  }
9  let result = greet.apply(person, ["Hello", "How are you?"]);
10 console.log(result);
11
12 // output : Hello John. How are you?
13
```

تاني استخدام ليها انها بتخلي object يستعير function من object تاني زي كذا

```
1  const computer = {
2    name: "MacBook",
3    isOn: false,
4
5    turnOn() {
6      this.isOn = true;
7      return `The ${this.name} is On`;
8    },
9    turnOff() {
10     this.isOn = false;
11     return `The ${this.name} is Off`;
12   },
13 };
14
15 const server = {
16   name: "Dell PowerEdge T30",
17   isOn: false,
18 };
19
20 let result2 = computer.turnOn.apply(server);
21 // output: The Dell PowerEdge T30 is On
22 let result3 = computer.turnOff.apply(server);
23 // output: The Dell PowerEdge T30 is Off
```

تالت استخدام لل apply method انها بتضيف 2 array لبعض زي concat

```
1 let arr = [1, 2, 3];
2 let numbers = [4, 5, 6];
3 arr.push.apply(arr, numbers);
4 console.log(arr);
5 //output: [1,2,3,4,5,6]
```

Call Method :- اول استخدام في عمل call لل function وبتاخذ 2 arguments وهم this العائدة علي ال object وال function arguments زي في المثال :

```
1 const person = {
2   firstName: "John",
3   LastName: "doe",
4 };
5
6 function greet(greeting, message) {
7   return `${greeting} ${this.firstName}. ${message}`;
8 }
9 let result1 = greet.call(person, "Hello", "How are you?");
10 //output: Hello John. How are you?
```

تاني استخدام ليها انها بتخلي object يستعير function من object تاني زي كذا

```
1  const car = {
2    name: "car",
3    start() {
4      console.log("Start the " + this.name);
5    },
6    speedUp() {
7      console.log("Speed up the " + this.name);
8    },
9    stop() {
10     console.log("Stop the " + this.name);
11   },
12 };
13
14 const aircraft = {
15   name: "aircraft",
16   fly() {
17     console.log("Fly");
18   },
19 };
20
21 car.start.call(aircraft);
22 car.speedUp.call(aircraft);
23
24 //output: Start the aircraft
25 //output: speed up the aircraft
```

Bind Method :- بنستخدمها علشان ننشئ function من function جوا object زي كذا:

```
1  let runner = {
2    name: "Runner",
3    run: function (speed) {
4      console.log(this.name + " runs at " + speed + " mph.");
5    },
6  };
7  let flyer = {
8    name: "Flyer",
9    fly: function (speed) {
10     console.log(this.name + " flies at " + speed + " mph.");
11   },
12 };
13
14 let run = runner.run.bind(flyer, 20);
15 run();
16
17 //output: Flyer runs at 20 mph.
```

Let and const :- بيتعملهم hoisting لكن بدون default initialization علشان كذا هيطلع error زي في المثال
لكن var بيتعملها hoisting لكن ب default initialization علشان كذا بيطلع undefined output



```
1 console.log(number1);
2 var number1 = 10;
3 //output: undefined
4
5 console.log(number);
6 let number = 10; //or const
7 //output: Uncaught ReferenceError: Cannot access 'number' before initialization
```