Посилання:

https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global\_Objects/Function/bind

<https://getinstance.info/articles/javascript/call-apply-and-bind-functions/>

<https://www.w3schools.com/js/js_function_call.asp>

https://www.w3schools.com/js/js\_function\_apply.asp

<https://learn.javascript.ru/call-apply>

Задача. ПРИКЛАД З ЛЯМБДА ВИРАЗОМ, ЯКИЙ ЗБЕРІГАЄ КОНТЕКСТ

<https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Functions/Arrow_functions>

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

function myfunction() {

this.a = 2;

this.f = function () {

alert(this.a);

}

setTimeout(()=>this.f(), 1000);

}

var ob = new myfunction();

</script>

</head>

<body>

</body>

</html>

Приклад з позичанням методів

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

var ob1 = {

a: 1,

b: 2,

show: function () {

document.write(` a=${this.a} b=${this.b}`);

}

};

var ob2 = {

a: 11,

b: 22,

c: 1234

};

var ob3 = {

a: "Hello",

b: 3.5,

s: "dfdhfgjk"

};

//------------ Позичання методів (якщо використовується метод багато разів) ------

//ob2.show = ob1.show; //Позичання метода

//ob2.show(); //Виведе a=11 b=22 бо this == ob2 (this–це об’єкт, який зліва від крапки)

//----------------------------

//function myfunction() {

// arguments.filter = [].filter; //Позичання метода

// return arguments.filter(x=>x<7); //Використання позиченого метода

//}

//myfunction(2, 1, 3, 4, 5, 2, 56, 6, 4, 6, 7, 3);

//============= Використання одноразового виклику метода в іншому контексті =====

//об"єкт\_власник\_метода.потрібний\_метод. call (цільовий\_об"єкт, список\_параметрів (якщо потрібно))

ob1.show.call(ob2);

ob1.show.call(ob3);

</script>

</head>

<body>

</body>

</html>

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

var ob1 = {

a: 1,

b: 2,

show: function (startMsg, endMsg) {

document.write(`${startMsg} : a=${this.a} b=${this.b} : ${endMsg} <br>`);

},

};

ob1.show = ob1.show.bind(ob1); // Зафіксували this = ob1

var ob2 = {

a: 11,

b: 22,

c: 1234

};

//------------

ob2.show = ob1.show; //Позичання метода

ob2.show("Hello", "Bye"); //Виведе Hello: a=1 b=2 :Bye бо this==ob1 (хоч і викликали через ob2)

//----------------------------

</script>

</head>

<body>

</body>

</html>

var person = {

fullName: function (city, country) {

return this.firstName + " " + this.lastName + "," + city + "," + country;

}

}

var person1 = {

firstName: "John",

lastName: "Doe",

}

//об"єкт\_власник. потрібний\_метод. apply (цільовий\_об"єкт, МАСИВ\_параметрів (якщо потрібно))

person.fullName.apply(person1, ["Oslo", "Norway"]);

Таким чином між call і apply різниця у тому, що при виклику call вказуємо список параметрів, а при виклику apply – масив аргументі

//об"єкт\_власник\_метода.потрібний\_метод. call (цільовий\_об"єкт, список\_параметрів (якщо потрібно))

//об"єкт\_власник\_метода.потрібний\_метод. apply(цільовий\_об"єкт, МАСИВ\_параметрів (якщо потрібно))