/\*

call() method :

The call() method in JavaScript is used to call a function with a

specified this value and arguments provided individually. It allows

you to explicitly set the context (this value) for a function

invocation, which is particularly useful when working with object

oriented programming and when borrowing methods from other

objects.

 \*/

// Syntac functionname.cal/(thisArd, arg1 , arg2)

*const* person = {

  fullName: *function* (*city*, *country*) {

    return this.firstName + " " + this.lastNaame + ", " + *city* + ", " + *country*;

  },

};

*const* person1 = {

  firstName: "John",

  lastNaame: "Dou",

};

*const* person2 = {

  firstName: "Alice",

  lastNaame: "Wallker",

};

// calling ullName fuction with different context using call()

console.log(person.fullName.call(person1, "New York", "USA"));

console.log(person.fullName.call(person2, "London", "UK"));

//exampe basic usage

*function* greet() {

  return `hello, ${this.name}`;

}

*const* per = { name: "Alica" };

console.log(greet.call(per));

//by using call(), we invoke the great function with the person

//  object is context, it allowing this greet to refer to person.

//Example 2 : passing arguments

*function* introduce(*age*, *gender*) {

  return `I am ${this.name}, ${*age*} year old, ${*gender*}.`;

}

*const* obj = { name: "Bob" };

console.log(introduce.call(obj, 30, "male"));

//Here call( ) allows us topass arguments directly to the

//  fuction being called with specified name the context.

//Example 3 : Borrowing Methods

*const* dog = {

  speak: *function* () {

    return `Woof, my name is ${this.name}`;

  },

};

*const* cat = { name: "Fluffy" };

console.log(dog.speak.call(cat));

// in this example demonstartes the how to borrow a method

// (speak) from one object

// (dog) and use it with in the context of another object(cat).

//Example 4 Inheriting constructor

*function* PersonN(*name*, *age*) {

  this.name = *name*;

  this.age = *age*;

}

*function* Student(*name*, *age*, *grade*) {

  PersonN.call(this, *name*, *age*);

  this.grade = *grade*;

}

*const* studentobj = new Student("Rahul", 30, "A");

console.log(studentobj);

// here , call() is used to invoke the PersonN constructor

// with in the Student constructor to set properties inherited

// from PersonN.

//Example 5 : Function Currying

*function* greeting(*greet*, *punctuation*) {

  return `${*greet*}, ${this.name} ${*punctuation*}`;

}

*const* obj2 = { name: "Ramesh" };

*const* greetingFuction = greeting.bind(obj2);

console.log(greetingFuction("hi", "!"));