# Exercise: Prototypal Chain and Inheritance in JavaScript

This document defines an in-class exercise from the ["Advanced JavaScript" Course @ Software University](https://softuni.bg/courses/advanced-javascript).

## Don’t Rely on Chance

In JavaScript everything is public and everything can be overwritten. To prove this, make the following function work without throwing an error:

function getRandomNum() {  
 var randomNum = Math.floor(Math.random() \* 10);  
 return randomNum;  
}

var mysteryNum = getRandomNum();  
for(var i = 0; i < 10; i++) {  
 var currentMysteryNum = getRandomNum();  
 if(currentMysteryNum != mysteryNum) {  
 throw new Error("No chance for you today!");  
 }  
}

## Extension Methods

Write a function getRandom() which works on arrays and returns random elements from the array it’s been called on. The function should work like this:

* Arrays - should return a random element in the array

var arr = [1, 3, 5, 10];

arr.getRandom(); // 5 (or some other random item)

* Strings – should return a random character in the string

var str = "This is an example string";

str.getRandom(); // "s" (or some other random item)

* Objects – should return a random property (key-value pair) from the object

var obj = {

name: "Gosho",

age: 25,

grade: 5.95,

isActive: true,

languages: ["English", "French"]

};

obj.getRandom(); // { grade: 5.95 } (or some other random property)

## call() and apply()

Pretend that **Function.prototype.call()** does not exist. Write JavaScript’s call() function using apply().

## Sorted List

Create a class that maintains a list of numbers in ascending order.

The class will have two methods:

* **add(num)** will add **num** to the list
* **get(index)** will get the value at the **ith** index in the list

You should also provide a **length** property (**not** method) that gives the length of the list.