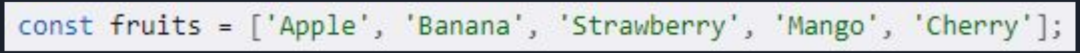
PF Objects and Arrays

Presentation A

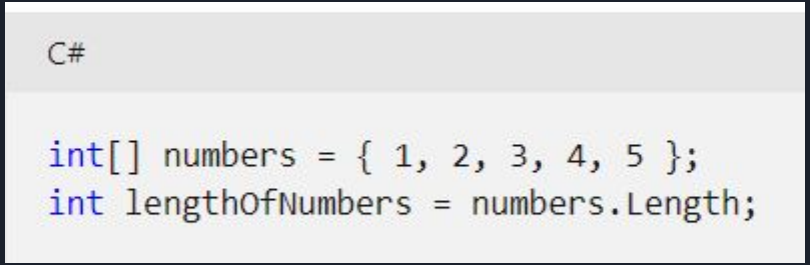
Which data type is used specifically to store sequences of values? Show an example in the following programming languages: Javascript, C#, Java

The Array data type enables storing a collection of multiple items under a single variable name, and has members for performing common array operations.



```
const fruits = ['Apple', 'Banana', 'Strawberry', 'Mango', 'Cherry'];
```

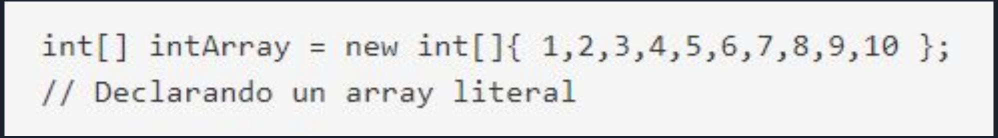
Js



C#

```
int[] numbers = { 1, 2, 3, 4, 5 };
int lengthOfNumbers = numbers.Length;
```

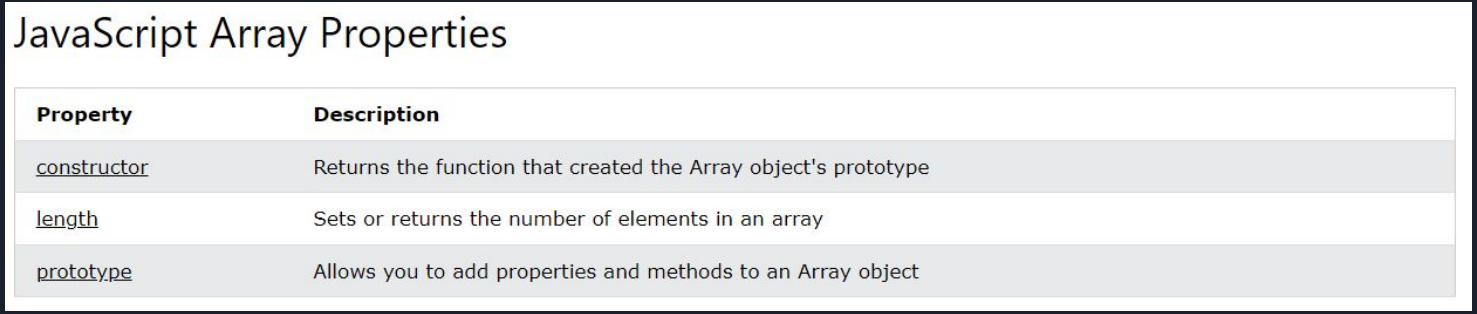
C#



```
int[] intArray = new int[]{ 1,2,3,4,5,6,7,8,9,10 };
// Declarando un array literal
```

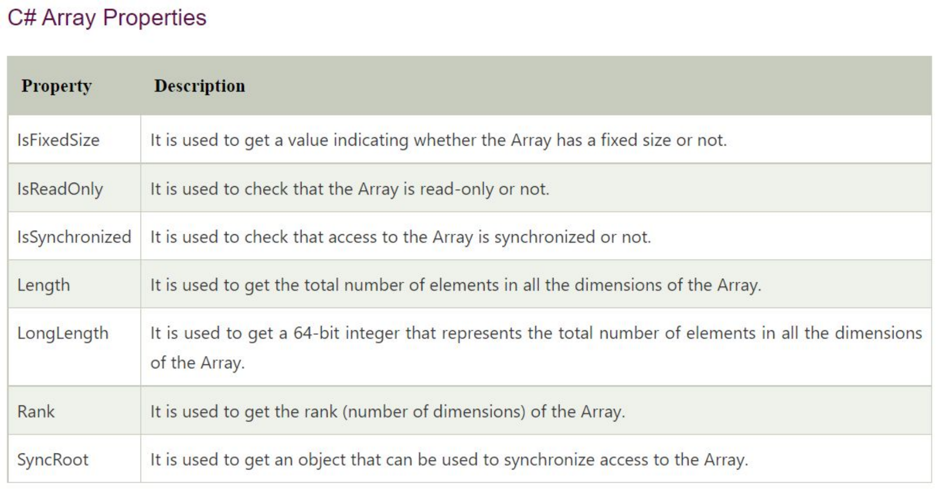
Java

Which are the main properties of arrays? Search for the properties that arrays have in the following programming languages: Javascript, C#, Java



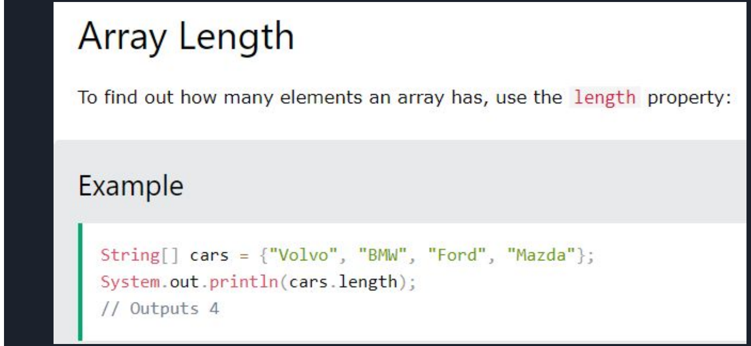
JavaScript Array Properties

Property	Description
constructor	Returns the function that created the Array object's prototype
length	Sets or returns the number of elements in an array
prototype	Allows you to add properties and methods to an Array object



C# Array Properties

Property	Description
IsFixedSize	It is used to get a value indicating whether the Array has a fixed size or not.
IsReadOnly	It is used to check that the Array is read-only or not.
IsSynchronized	It is used to check that access to the Array is synchronized or not.
Length	It is used to get the total number of elements in all the dimensions of the Array.
LongLength	It is used to get a 64-bit integer that represents the total number of elements in all the dimensions of the Array.
Rank	It is used to get the rank (number of dimensions) of the Array.
SyncRoot	It is used to get an object that can be used to synchronize access to the Array.



Array Length

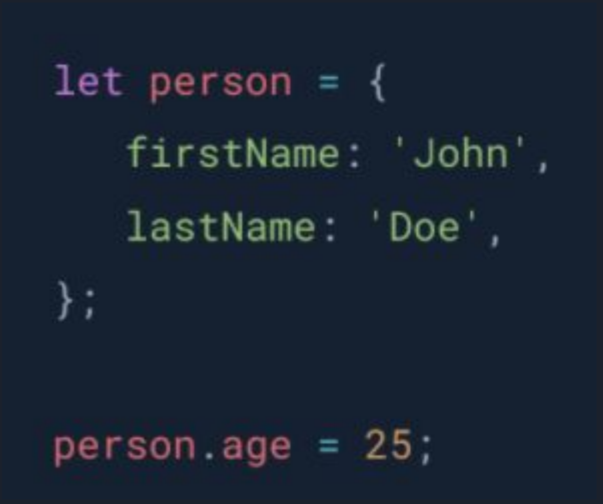
To find out how many elements an array has, use the `length` property:

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars.length);
// Outputs 4
```

What is an object? Show a real life example to better understand this concept

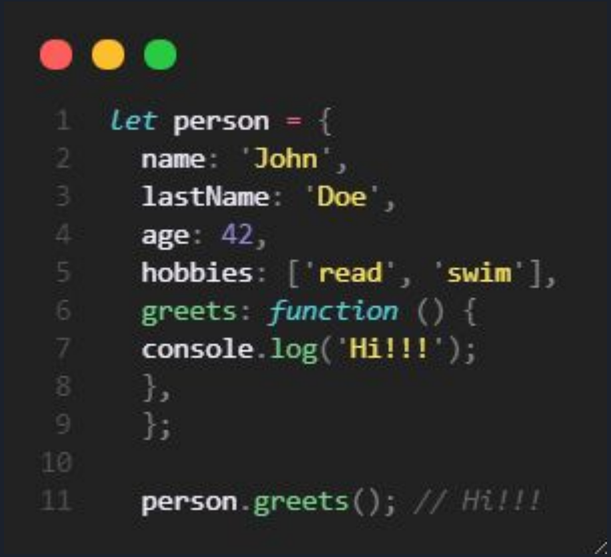
An object is an unordered collection of properties, each of which has a name and a value. Property names are strings, so objects map strings to values. Javascript objects are dynamic properties can usually be added and deleted. Objects are mutable and are manipulated by reference rather than by value.



```
let person = {
   firstName: 'John',
   lastName: 'Doe',
};

person.age = 25;
```

What data types can you assign as values of objects in Javascript? Show an example of an object with the different data types assigned to it

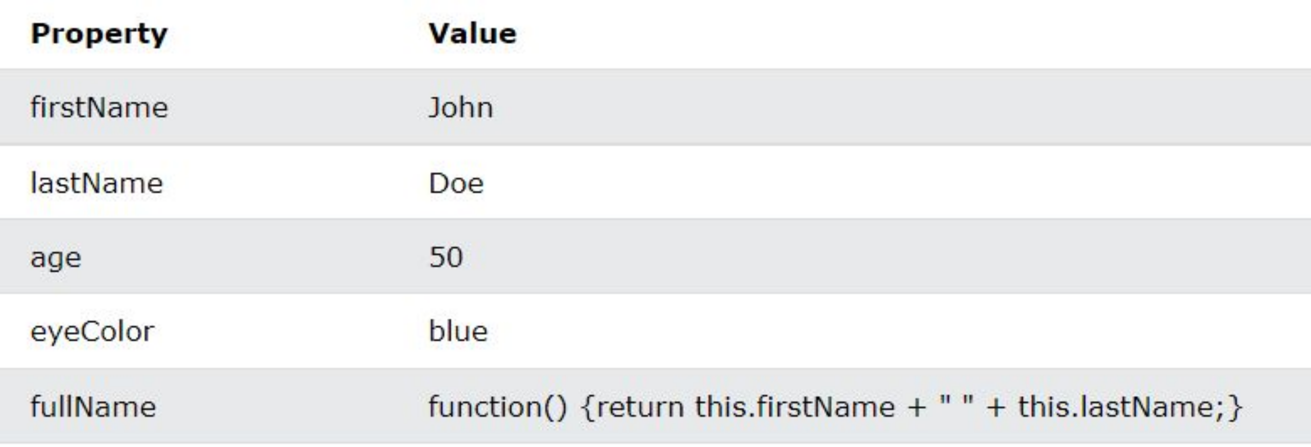
Objects can store any type of data such as arrays, booleans, other objects, etc.



```
let person = {
  name: 'John',
  lastName: 'Doe',
  age: 42,
  hobbies: ['read', 'swim'],
  greets: function () {
  console.log('Hi!!!');
  },
  };

  person.greets(); // Hi!!!
```

What are methods?

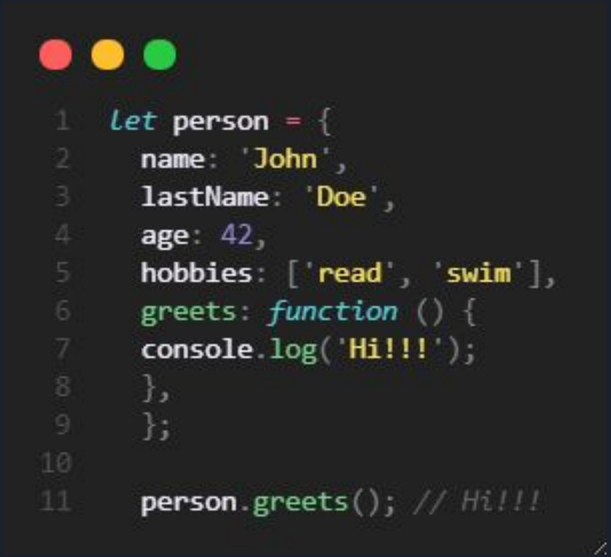
JavaScript methods are actions that can be performed on objects. A JavaScript method is a property containing a function definition. Methods are functions stored as object properties.



Property	Value
firstName	John
lastName	Doe
age	50
eyeColor	blue
fullName	function() {return this.firstName + " " + this.lastName;}

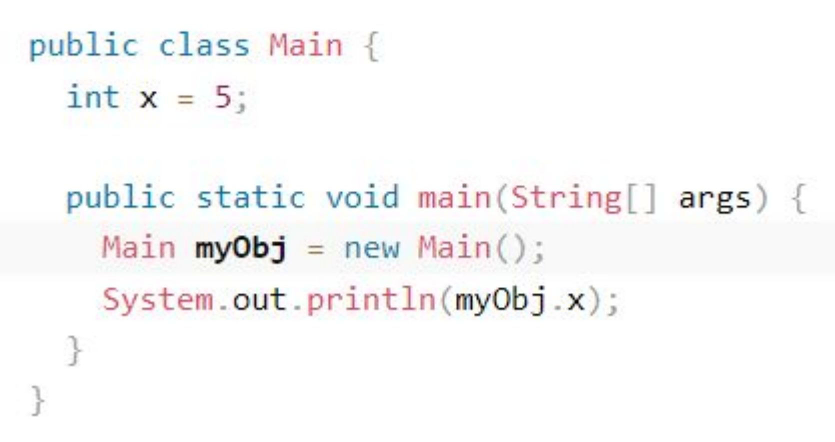
How can you create an object in the following languages? Javascript, C# and Java

Js



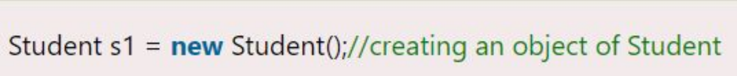
```javascript
let person = {
  name: 'John',
  lastName: 'Doe',
  age: 42,
  hobbies: ['read', 'swim'],
  greets: function () {
  console.log('Hi!!!');
  },
  };

  person.greets(); // Hi!!!
```

Java



```java
public class Main {
  int x = 5;

  public static void main(String[] args) {
    Main myObj = new Main();
    System.out.println(myObj.x);
  }
}
```

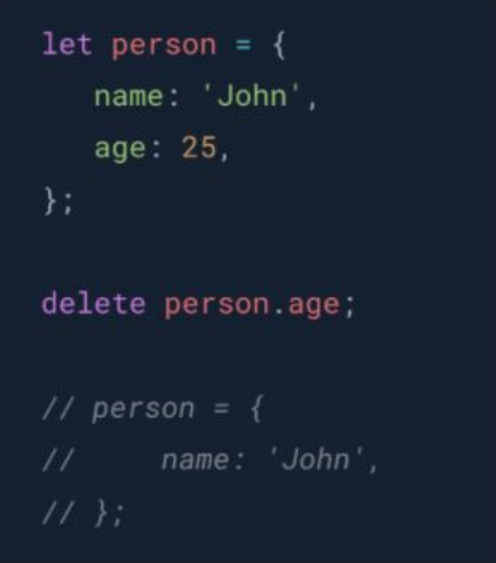
C#



```csharp
Student s1 = new Student();//creating an object of Student
```

What is the delete operator of objects in Javascript? Show an example

We can use the delete operator to remove properties from objects.



```
let person = {
   name: 'John',
   age: 25,
};

delete person.age;

// person = {
//     name: 'John',
// };
```

What is the **in operator** of objects in Javascript? Show a practical example of using the **in operator**

The JavaScript in operator is **used to check if a specified property exists in an object or in its inherited properties** (in other words, its prototype chain). The in operator returns true if the specified property exists.

```
const car = {
  make: 'Toyota',
  model:'Camry',
  year: '2018',
  start: function() {
    console.log(`Starting ${this.make} ${this.model}, ${this.year}`);
  }
}
'make' in car // Returns true.
'start' in car // Returns true.
'Toyota' in car // Returns false.
```

What is the difference between the **in operator** and the **hasOwnProperty method** in Javascript?

The key difference is that **in will return true for inherited properties, whereas hasOwnProperty() will return false for inherited properties**

What **built-in array methods** exist in the following programming languages?Javascript,Java,C#

Array.push: allows to add a value to the end of an array.(javascript)

Array.pop: allows to remove the last element of an array.(javascript)

Array.add: allows to add a value to the end or in the position specified of an array.(java)

Array.remove: allows to delete the specified element in the index position of an array(java)

Array.Add: allows to add a value in the array(c#)

remove in c# doesnt exists, a form to make this, is to create another array with de new elements without the element you want to remove.

Find the **built-in methods** that **objects** have in Javascript and show an example of each one of them

Object.keys(obj)
Returns an array of the keys that the object contains.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/keys
Object.values(obj)
Returns an array of the values of each of the elements in the object.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/values
Object.entries(obj)
Returns an array containing nested arrays for each key-value pair. The first element in each nested array is the key, and the second is the value.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/entries
Object.fromEntries(arr)
Creates a new object from an array of key-value pairs passed as the argument. Returns the new object.
https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Object/fromEntries

What is the difference between an **array** and an **object** in Javascript?

Explain when it is recommended to use each one of them.

Array is a list of values and object is a complex of key and values pairs and defines an structure.

You can use Array to save a list of passengers of a flight.

You can use an object to define the information of each passenger.

Explain the difference between **mutability** and **immutability**

A mutable object is an object whose state can be modified or changed over time. An immutable object, on the other hand, is an object whose state cannot be modified after it is created.

Find an example of a **mutable** and an **immutable** data type in Javascript and show a practical example

In JavaScript, string and numbers are immutable data types

```
var x =7;
                                                                                2
x += 1;
In JavaScripts, objects are mutable.

let employee1 = { name: "Joe Clarke", age: 35, id: 123, city: 'London' };

let employee2 = employee1;

let employee3 = { name: "Joe Denly", age: 25, id: 121, city: 'Birmingham'};

console.log("check Whether employee1=employee2", employee1===employee2);

console.log("check Whether employee1=employee3", employee1===employee3);
```

Write a function that returns an array with all the keys of the following object:

```
product = { name: “Sony WH-1000XM3”, price: ”300.00”, itemsInStock: 23 };
```

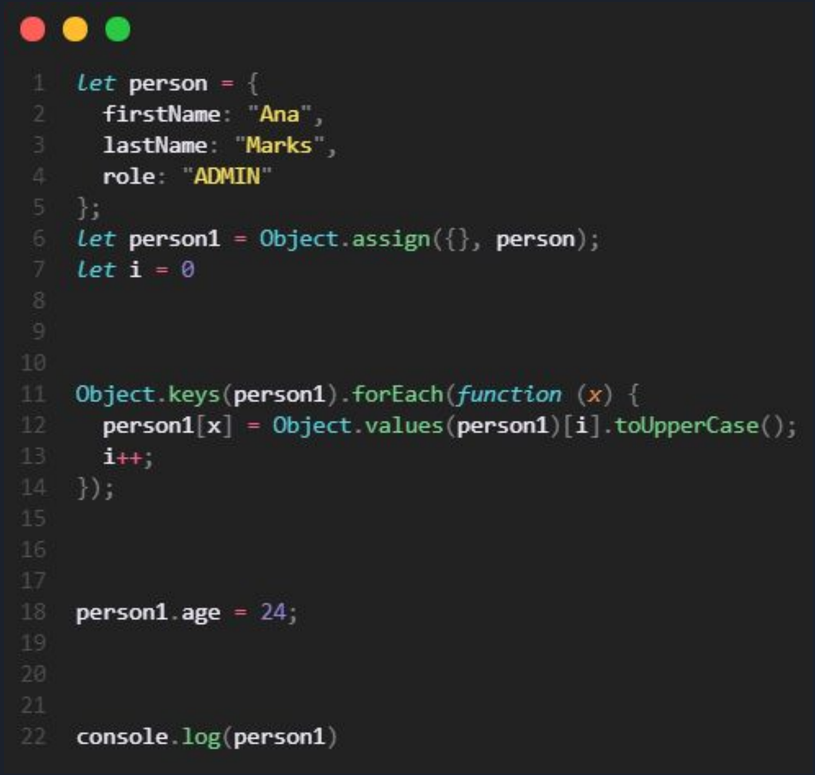
```
return Object.keys(product);
```

Write a function that returns an array with all the keys and values of the following object:

```
product: { name: “Sony WH-1000XM3”, price: ”300.00”, itemsInStock: 23 };
```

```
return Object.entries(product);
```

Write a function that receives the person object as a parameter and that changes all the values of the object to upper case and adds a new property of the object named age with the value of 24 (age: 24). The function should return a new object with the values changed and the new property without mutating the original object.



```
let person = {
  firstName: "Ana",
  lastName: "Marks",
  role: "ADMIN"
};
let person1 = Object.assign({}, person);
let i = 0



Object.keys(person1).forEach(function (x) {
  person1[x] = Object.values(person1)[i].toUpperCase();
  i++;
});



person1.age = 24;



console.log(person1)
```

PF Objects and Arrays

Test B

What are the main properties of arrays? Find the properties that arrays have in the following programming languages

JavaScript:

- Constructor: Returns the function that created the Array object’s prototype.
- length: Sets or returns the number of element in an array.
- prototype: Allows you to add properties and methods to an Array object.

Python:

What built-in array methods are there in the following languages?

Python: append(), clear(), copy(), count(), extend(), index(), insert(), pop(), remove(), reverse(), sort()

PHP: array_unshift, asort, extract, ksort, key, pos, sizeof,....

Go: len(array), append

What is an **object**? Show a real life example to better understand this concept

In JavaScript, an object is **an unordered collection of key-value pairs**. Each key-value pair is called a property. The key of a property can be a string. And the value of a property can be any value, e.g., a string, a number, an array, and even a function. JavaScript provides you with many ways to create an object.

```
person{

nombre: juan,

edad: 24,

role: admin,

}
```

What are **object keys** and **object values**?

The keys of an object is the list of property names. The values of an object is the list of property values.

How can you create an **object** in the following languages?Python,PHP,Go

```python
class Objeto():
    color = "verde"
    tamanio = "grande"
    aspecto = "feo"
    antenas = Antena()
    ojos = Ojo()

      pelos = Pelo()
```

```php
<?php
class foo
{
    function hacer_algo()
    {
        echo "Haciendo algo.";
    }
}

$bar = new foo;
$bar->hacer_algo();
?>
```

```go
type Mascota struct{
edad int
nombre,raza string
}

func main(){
mascota := Mascota{1,
“Maggie”,”no sé”
}
```

How can you create an object without the need to create a class previously in PHP? Show an example

Yes

```
 $object = new stdClass();

   $object->property = 'Here we go';
```

What is **dot notation** in objects? Show practical examples about using dot notation with objects

Dot notation allows you to access to the values of the properties of an object.

```
person = {
name: “Roberto”
}
console.log(person.name);
```

What is the **hasOwnProperty** method used for in Javascript? Show an example to better understand this method

The hasOwnProperty () method returns a boolean value that indicates whether the object has the specified property.

```
let person = {

name: 'Alex',

age: 25,

};

console.log(person.hasOwnProperty("name")); // true

console.log(person.hasOwnProperty("salary")); // false
```

What is **mutability**? Show an example to better understand this concept

Objects and arrays are accessed by reference which can cause the values to be changed in unexpected ways if we don’t make a deep copy of the objects.

```
let object1 = { value: 10 };
let object2 = object1;
let object3 = { value: 10 };

console.log(object1 === object2);
// → true
console.log(object1 === object3);
// → false

object1.value = 15;
console.log(object2.value);
// → 15
console.log(object3.value);
```

Explain the difference between **mutability** and **immutability**

the diference is that mutability makes changes in state of the object, while inmutability not makes changes in numbers or strings state

Find an example of a **mutable** and an **immutable** data type in Javascript and show a practical example

In JavaScript, string and numbers are immutable data types

```
var x =7;

x += 1;
```

2

In JavaScripts, objects are mutable.

```
let employee1 = { name: "Joe Clarke", age: 35, id: 123, city: 'London' };

let employee2 = employee1;

let employee3 = { name: "Joe Denly", age: 25, id: 121, city: 'Birmingham'};

console.log("check Whether employee1=employee2", employee1===employee2);

console.log("check Whether employee1=employee3", employee1===employee3);
```

Exercise 1.

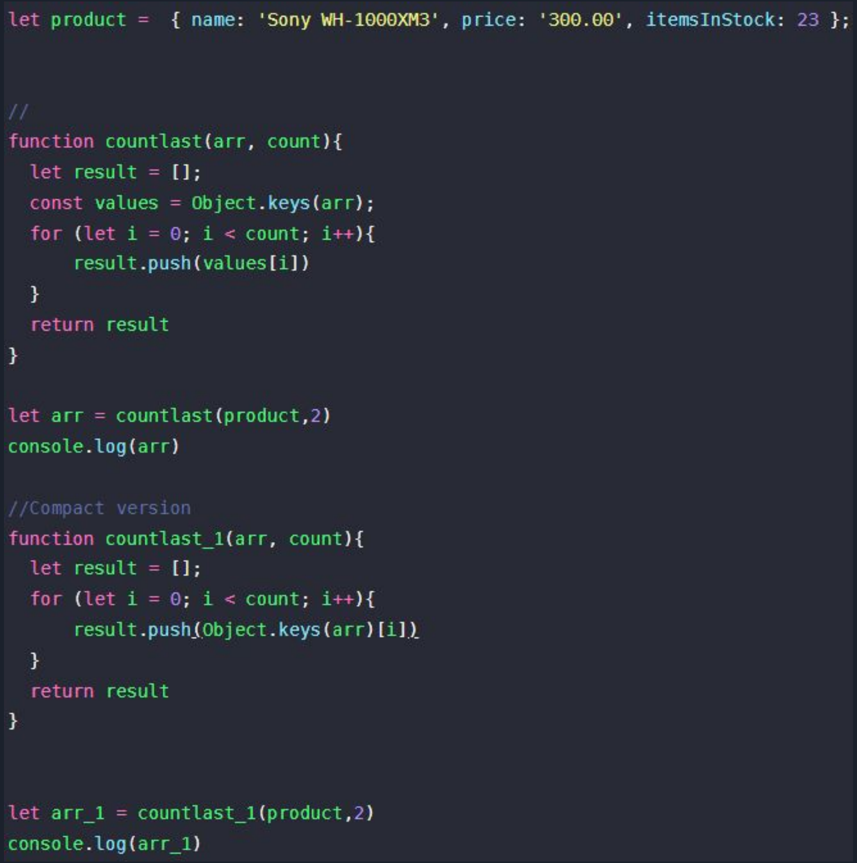
1. **Write a function that receives the student object as a parameter and that appends to all the values of the object the value of ‘Edited’ (example: firstName: ‘Ana Edited’), adds a new property of the object named grade with the value of 9.5 (grade: 9.5) and that removes the role property. The function should return a new object with the values changed and the new property without mutating the original object.**

```
let student = { firstName: "Ana", lastName: "Marks", role:
"STUDENT" };
var student2 = Object.assign({}, student);
function stu(student){
  student2 = Object.assign({}, student);
  let i = 0;

Object.keys(student).forEach(function(x){
  student2[x] = Object.values(student2)[i] + " EDITED";
  i++;
});
  student2.age = 9.5;
  delete student2.role;
  console.log(student2)
}

stu(student);
```

Exercise 2

1. **Write a function that returns an array with the first 2 keys of the following object:**

- product: { name: “Sony WH-1000XM3”, price: ”300.00”, itemsInStock: 23 };

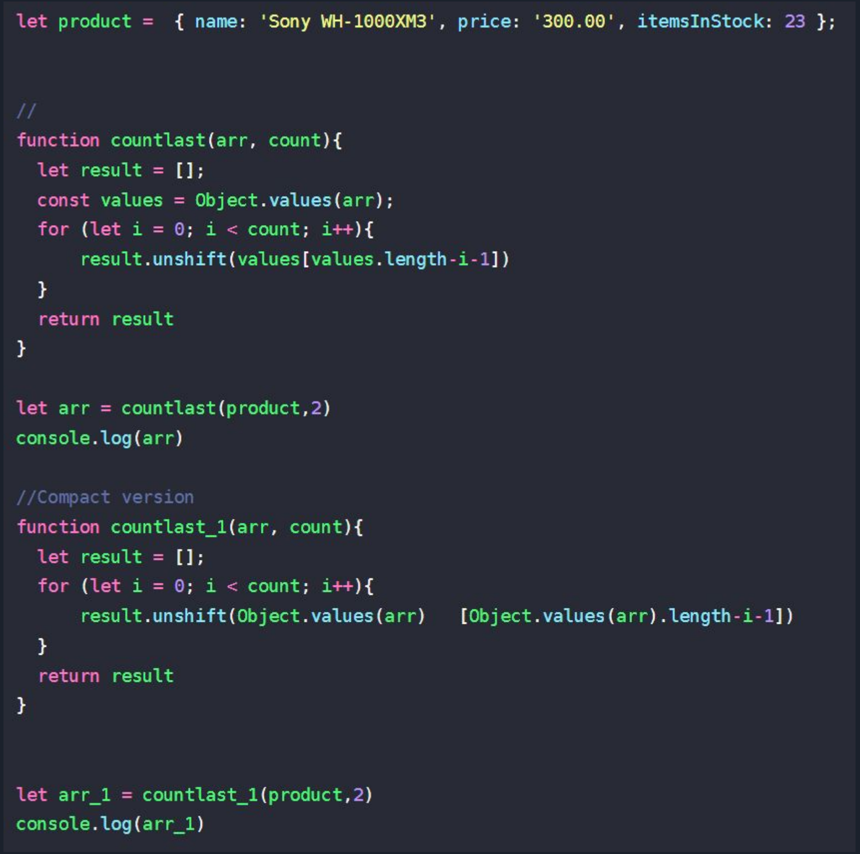


```
let product =  { name: 'Sony WH-1000XM3', price: '300.00', itemsInStock: 23 };


//
function countlast(arr, count){
  let result = [];
  const values = Object.keys(arr);
  for (let i = 0; i < count; i++){
      result.push(values[i])
  }
  return result
}

let arr = countlast(product,2)
console.log(arr)

//Compact version
function countlast_1(arr, count){
  let result = [];
  for (let i = 0; i < count; i++){
      result.push(Object.keys(arr)[i])
  }
  return result
}


let arr_1 = countlast_1(product,2)
console.log(arr_1)
```

Exercise 3

1. **Write a function that returns an array with the last 2 values of the following object:**

- product: { name: “Sony WH-1000XM3”, price: ”300.00”, itemsInStock: 23 };



```
let product =  { name: 'Sony WH-1000XM3', price: '300.00', itemsInStock: 23 };


//
function countlast(arr, count){
  let result = [];
  const values = Object.values(arr);
  for (let i = 0; i < count; i++){
      result.unshift(values[values.length-i-1])
  }
  return result
}

let arr = countlast(product,2)
console.log(arr)

//Compact version
function countlast_1(arr, count){
  let result = [];
  for (let i = 0; i < count; i++){
      result.unshift(Object.values(arr)   [Object.values(arr).length-i-1])
  }
  return result
}


let arr_1 = countlast_1(product,2)
console.log(arr_1)
```