



Programación Imperativa

¡Más prácticas!

Como ya observaron esta semana cerramos con los contenidos que veremos en esta materia: *variables, tipos de datos, condicionales y bucles*, que son herramientas esenciales para cualquier tipo de lenguaje de programación, pero para poder realizar un correcto uso de éstas necesitamos ejercitar nuestro pensamiento computacional, entonces vamos a hacerlo resolviendo los siguientes enunciados de la forma que creamos más conveniente.

Ejercicios

1. A partir de el siguiente array de edades nos solicitan realizar lo siguiente:

```
const edades = [11, 12, 15, 18, 25, 22, 10, 33, 18, 5];
```

- a. Obtener en un nuevo array las edades menores a 18.



```
function arrayDeEdadesMenores(array){  
    let arrayDeEdadesMenores = [];  
  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i]<18){  
            arrayDeEdadesMenores.push(array[i]);  
        }  
    }  
  
    return arrayDeEdadesMenores;  
}  
  
let arrayDeMenores = arrayDeEdadesMenores(edades);  
console.log(arrayDeMenores);
```

- b. Obtener en un nuevo array las edades mayor o igual a 18.

```
function arrayDeEdadesMayores(array){  
    let arrayDeEdadesMayores = [];  
  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i]>=18){  
            arrayDeEdadesMayores.push(array[i]);  
        }  
    }  
  
    return arrayDeEdadesMayores;  
}  
  
let arrayDeMayores = arrayDeEdadesMayores(edades);  
console.log(arrayDeMayores);
```

- c. Obtener en un nuevo array las edades igual a 18.



```
function arrayDeEdadesIgualesA18(array){  
    let arrayDeEdadesIguales = [];  
  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i]==18){  
            arrayDeEdadesIguales.push(array[i]);  
        }  
    }  
  
    return arrayDeEdadesIguales;  
}  
  
let arrayDeEdades18 = arrayDeEdadesIgualesA18(edades);  
console.log(arrayDeEdades18);
```

d. Obtener el menor.

```
function obtenerElMenorValor(array){  
    let valorMenor = array[0];  
  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i]<valorMenor){  
            valorMenor = array[i];  
        }  
    }  
  
    return valorMenor;  
}  
  
let menor = obtenerElMenorValor(edades);  
console.log(menor);
```

e. Obtener el mayor.



```
function obtenerElMayorValor(array){  
    let valorMenor = array[0];  
  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i]>valorMenor){  
            valorMenor = array[i];  
        }  
    }  
  
    return valorMenor;  
}
```

```
let mayor = obtenerElMayorValor(edades);  
console.log(mayor);
```

- f. Obtener el promedio de edades.

```
function promedioDeEdades(array){  
    let suma = 0;  
    let promedio = 0;  
    let i=0;  
  
    while(i<array.length){  
        suma = suma + array[i];  
        i++;  
    }  
    promedio = suma/i;  
  
    return promedio;  
}
```

```
let promedioObtenido = promedioDeEdades(edades);  
console.log("El promedio de edades es: " + promedioObtenido);
```

- g. Incrementar en 1 todas las edades.



```
function incrementarEdades(array){  
    for(let i=0 ; i<array.length ; i++){  
        array[i] = array[i] + 1;  
    }  
  
    return array;  
}  
  
let arrayEdadesIncrementadas = incrementarEdades(edades);  
console.log(arrayEdadesIncrementadas);
```

Recordemos que hay muchas formas de resolver, nuestra tarea será pensar cuál es la mejor para poder reutilizar nuestro código con diferentes array de edades.

```
const edadesGrupo1 = [21, 12, 15, 18, 25];  
const edadesGrupo2 = [2, 11, 54, 63, 24];
```

Excelente, ahora, como ya trabajamos un array, sabemos que este puede tener como elemento cualquier tipo de dato —Number, String, Objetos Literales, Array, etc—. Para continuar con la Mesa de Trabajo trabajaremos con un array de objetos literales, para ello invitamos a repasar la clase de **Objetos Literales** para tener en claro su estructura y sintaxis. Podemos decir que un objeto literal es una colección de propiedades y cada propiedad es una asociación entre una clave y un valor.

```
/*  
Observamos que es un array por su sintaxis que comienza y termina con corchetes []  
y sus elementos tienen la sintaxis de objetos literales {}  
Entonces, podemos decir que tenemos una array de objetos literales,  
o una colección de objetos literales
```



```
*/  
const arrayDeObjetosLiterales =  
[  
  {},  
  {},  
  {},  
  {}  
];
```

2. Tenemos como base un array de cuentas bancarias, donde a cada una la representamos con un objeto literal. A partir de este array trabajaremos sobre los siguientes enunciados, resolviendo de la forma que nos parezca más adecuada

```
const arrayCuentas =  
[  
  {  
    titular: "Arlene Barr",  
    estaHabilitada: false,  
    saldo: "$3,253.40",  
    edadTitular: 33,  
    tipoCuenta: "sueldo"  
  },  
  {  
    titular: "Roslyn Torres",  
    estaHabilitada: false,  
    saldo: "$3,229.45",  
    edadTitular: 27,  
    tipoCuenta: "sueldo"  
  },  
  {  
    titular: "Cleo Lopez",  
    estaHabilitada: true,  
    saldo: "$1,360.19",  
    edadTitular: 34,  
    tipoCuenta: "corriente"  
  },  
  {  
    titular: "Daniel Malone",
```



```
    estaHabilitada: false,  
    saldo: "$3,627.12",  
    edadTitular: 30,  
    tipoCuenta: "sueldo"  
  },  
  {  
    titular: "Ethel Leon",  
    estaHabilitada: true,  
    saldo: "$1,616.52",  
    edadTitular: 34,  
    tipoCuenta: "sueldo"  
  },  
  {  
    titular: "Harding Mitchell",  
    estaHabilitada: true,  
    saldo: "$1,408.68",  
    edadTitular: 25,  
    tipoCuenta: "corriente"  
  }  
]
```

- a. Obtener un nuevo array de cuentas cuyas edades sean menores a 30.



```
function arrayConEdadesMenores(array){  
    let arrayConMenores = [];  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i].edadTitular < 30){  
            arrayConMenores.push(array[i]);  
        }  
    }  
    return arrayConMenores;  
}  
  
let arrayDeMenores = arrayConEdadesMenores(arrayCuentas);  
console.log(arrayDeMenores);
```

- b. Obtener un nuevo array de cuentas cuyas edades sean mayor o igual a 30.

```
function arrayConEdadesMayoresOIguales(array){  
    let arrayConMayores = [];  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i].edadTitular >= 30){  
            arrayConMayores.push(array[i]);  
        }  
    }  
    return arrayConMayores;  
}  
  
let arrayDeMayores = arrayConEdadesMayoresOIguales(arrayCuentas);  
console.log(arrayDeMayores);
```

- c. Obtener un nuevo array de cuentas cuyas edades sean igual a 30.



```
function arrayConEdadesIgualesA30(array){  
    let arrayConEdad30 = [];  
  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i].edadTitular == 30){  
            arrayConEdad30.push(array[i]);  
        }  
    }  
  
    return arrayConEdad30;  
}  
  
let arrayDeEdadTreinta = arrayConEdadesIgualesA30(arrayCuentas);  
console.log(arrayDeEdadTreinta);
```

- d. Obtener la cuenta con el titular de la misma más joven.

```
function obtenerTitularMasJoven(array){  
    let titularJoven = {};  
    let edadMasJoven = array[0].edadTitular;  
  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i].edadTitular < edadMasJoven){  
            titularJoven = array[i];  
        }  
    }  
  
    return titularJoven;  
}  
  
let titularMasJoven = obtenerTitularMasJoven(arrayCuentas);  
console.log(titularMasJoven);
```

- e. Obtener un nuevo array por cada tipo de cuenta.



```
function arrayPorTipoDeCuenta(array){

    let arrayCuentaSueldo = [];
    let arrayCuentaCorriente = [];
    let arrayContenedor = [];

    for(let i=0 ; i<array.length ; i++){
        if(array[i].tipoCuenta == "sueldo"){
            arrayCuentaSueldo.push(array[i]);
        }
        else{
            arrayCuentaCorriente.push(array[i]);
        }
    }

    arrayContenedor.push(arrayCuentaSueldo);
    arrayContenedor.push(arrayCuentaCorriente);

    return arrayContenedor;
}

let arrayOrdenado = arrayPorTipoDeCuenta(arrayCuentas);
console.log(arrayOrdenado);
```

- f. Obtener un nuevo array con las cuentas habilitadas.

```
function arrayDeCuentasHabilitadas(array){

    let arrayCuentasHabilitadas = [];

    for(let i=0 ; i<array.length ; i++){

        if(array[i].estaHabilitada === true){
            arrayCuentasHabilitadas.push(array[i]);
        }
    }

    return arrayCuentasHabilitadas;
}

let arraySoloCuentasHabilitadas = arrayDeCuentasHabilitadas(arrayCuentas);
console.log(arraySoloCuentasHabilitadas);
```

- g. Obtener un nuevo array con las cuentas deshabilitadas.



```
function arrayDeCuentasDeshabilitadas(array){  
  
    let arrayCuentasdeshabilitadas = [];  
  
    for(let i=0 ; i<array.length ; i++){  
        if(array[i].estaHabilitada === false){  
            arrayCuentasdeshabilitadas.push(array[i]);  
        }  
    }  
  
    return arrayCuentasdeshabilitadas;  
}  
  
let arraySoloCuentasDeshabilitadas = arrayDeCuentasDeshabilitadas(arrayCuentas);  
console.log(arraySoloCuentasDeshabilitadas);
```

- h. Obtener la cuenta con el menor saldo.

```
function obtenerCuentaDeMenorSaldo(array){  
  
    let cuentaDeMenorSaldo = {};  
    let menorSaldo = array[0].saldo;  
    parseFloat(menorSaldo);  
  
    for(let i=0 ; i<array.length ; i++){  
  
        let saldoAComparar = array[i].saldo;  
        parseFloat(saldoAComparar);  
  
        if(saldoAComparar < menorSaldo){  
            cuentaDeMenorSaldo = array[i];  
        }  
    }  
  
    return cuentaDeMenorSaldo;  
}  
  
let cuentaConSaldoMenor = obtenerCuentaDeMenorSaldo(arrayCuentas);  
console.log(cuentaConSaldoMenor);
```

- i. Obtener la cuenta con el mayor saldo.



```
function obtenerCuentaDeMayorSaldo(array){

    let cuentaDeMayorSaldo = {};
    let mayorSaldo = array[0].saldo;
    parseFloat(mayorSaldo);

    for(let i=0 ; i<array.length ; i++){

        let saldoAComparar = array[i].saldo;
        parseFloat(saldoAComparar);

        if(saldoAComparar > mayorSaldo){
            cuentaDeMayorSaldo = array[i];
        }
    }

    return cuentaDeMayorSaldo;
}

let cuentaConSaldoMayor = obtenerCuentaDeMayorSaldo(arrayCuentas);
console.log(cuentaConSaldoMayor);
```