

```

// Dino Crescimbeni
// github: https://github.com/dcrescimbeni/nivelatorio-bootcamp

// 1 - Loop de pares
function loopDePares(number) {
  for (let i = 0; i <= 100; i++) {
    console.log(i);
    if ((i + number) % 2 === 0) {
      console.log(`El número ${i + number} es par`);
    }
  }
}
// loopDePares(10);

// 2 - Loop de impares con palabra
function loopPalabra(number, word) {
  for (let index = 0; index <= 100; index++) {
    console.log(index);
    if ((index + number) % 2 === 1) {
      console.log(word);
    }
  }
}
// loopPalabra(1, 'dino');

// 3 - Sumatoria
function sumattion(number) {
  let sum = 0;
  for (let index = 0; index <= number; index++) {
    sum += index;
  }
  return sum;
}
// console.log(sumattion(3));
// console.log(sumattion(8));

// 4 - Callback
function callback(number, fun) {
  return fun(number);
}
// console.log(
//   callback(5, (num) => {
//     return num * 10;
//   })
// );
// console.log(
//   callback(25, (num) => {
//     return num / 5;
//   })
// );

// 5 - Nuevo arreglo
function nuevoArreglo(number) {
  let newArray = [];

```

```

    for (let i = 1; i <= number; i++) {
        newArray.push(i);
    }
    return newArray;
}
// console.log(nuevoArreglo(5));
// console.log(nuevoArreglo(10));

// 6 - Similar string.split()
function newSplit(string) {
    let stringSplit = [];
    for (let i = 0; i < string.length; i++) {
        stringSplit.push(string[i]);
    }
    return stringSplit;
}
// console.log(newSplit('hola'));
// console.log(newSplit('chau'));

// 7 - Caracter del medio
function middleCharacter(string) {
    let middlePos;
    if (string.length % 2 === 1) {
        // Impar
        middlePos = (string.length - 1) / 2;
        return string[middlePos];
    } else {
        middlePos = string.length / 2 - 1;
        return `${string[middlePos]}${string[middlePos + 1]}`;
    }
}
// console.log(middleCharacter('plataforma5'));
// console.log(middleCharacter('hola'));
// console.log(middleCharacter('cosas'));

// 8 - Mover ceros a lo último
function moveZeros(array) {
    const array1 = [];
    const arrayZero = [];
    array.forEach((element) => {
        if (element === 0) {
            arrayZero.push(0);
            return;
        }
        array1.push(element);
    });
    return array1.concat(arrayZero);
}
// console.log(moveZeros([false, 1, 0, 1, 2, 0, 1, 3, 'a']));
// console.log(moveZeros([1, 2, 0, 1, 0, 1, 0, 3, 0, 1]));

// 9 - Manejando dos arreglos
function arrayHandler(array1, array2) {
    if (array1.length !== array2.length) {

```

```

    console.log('Los arrays no son del mismo largo');
    return;
}
array1.forEach((element, index) => {
    console.log(`Soy ${element} y yo soy ${array2[index]}`);
});
}
// arrayHandler([1, 2, 3, 4], ['h', 'o', 'l', 'a']);

// 10 - Mezclando arreglos
function mezclarArreglos(array1, array2) {
    const newArray = [];
    let maxLength;
    // Determinar el array más largo
    if (array1.length > array2.length) {
        // El primer array es más largo
        maxLength = array1.length;
    } else {
        // El segundo array es igual o más largo que el primero
        maxLength = array2.length;
    }

    for (let i = 0; i < maxLength; i++) {
        array1[i] !== undefined && newArray.push(array1[i]);
        array2[i] !== undefined && newArray.push(array2[i]);
    }
    return newArray;
}
// console.log(mezclarArreglos([1, 2, 3, 4], ['h', 'o', 'l', 'a']));
// console.log(mezclarArreglos([1, 2, 3, 4], ['h', 'p']));
// console.log(mezclarArreglos(['h', 'p'], [1, 2, 3, 4]));

// 11 - Arreglos
// join
function newJoin(array, joiner) {
    let string = '';
    array.forEach((element) => {
        string += element;
        string += joiner;
    });
    return string;
}
// console.log(newJoin([1, 2, 3, 4, 5], ' '));

// pop
var transformArray = [1, 2, 3, 4, 5];
// console.log(transformArray);
function newPop(array) {
    const lastElement = array[array.length - 1]; // Get last element
    array.splice(array.length - 1, 1); // Delete last element
    return lastElement;
}
// console.log(newPop(transformArray));
// console.log(transformArray);

```

```

// filter
function newFilter(array, filterFun) {
  const newArray = [];
  array.forEach((element) => {
    if (filterFun(element)) {
      newArray.push(element);
    }
  });
  return newArray;
}
// console.log(
//   newFilter([1, 2, 3, 4, 5], (e) => {
//     return e % 2 === 0;
//   })
// );

// map
function newMap(array, mapFun) {
  const newArray = [];
  array.forEach((element) => {
    newArray.push(mapFun(element));
  });
  return newArray;
}
// console.log(
//   newMap([1, 2, 3, 4, 5], (e) => {
//     return e * 10;
//   })
// );

// 12
function minSum(array) {
  let min1;
  let min2;

  if (array[0] < array[1]) {
    min1 = array[0];
    min2 = array[1];
  } else {
    min1 = array[1];
    min2 = array[0];
  }

  for (let i = 2; i < array.length; i++) {
    if (array[i] < min2) {
      if (array[i] < min1) {
        min2 = min1;
        min1 = array[i];
      } else {
        min2 = array[i];
      }
    }
  }
}

```

```

    return min1 + min2;
}
// console.log(minSum([7, 6, 5, 4, 3, 2, 1]));
// console.log(minSum([1, 10, 43, 900, 20, 8]));

// 13
function arregloDeObjetos(number) {
    const newArray = [];
    for (let i = 1; i <= number; i++) {
        newArray.push({ valor: i });
    }
    return newArray;
}
// console.log(arregloDeObjetos(5));
// console.log(arregloDeObjetos(3));

// 14 - Arreglo de objetos
function arregloDeObjetos(number, string) {
    const newArray = [];
    for (let i = 1; i <= number; i++) {
        newArray.push({ [string]: i });
    }
    return newArray;
}
// console.log(arregloDeObjetos(5, 'hola'));
// console.log(arregloDeObjetos(3, 'chau'));

// 15 - Unica propiedad
arreglo = [
    { name: 'lucas', edad: 20 },
    { name: 'santi', edad: 22 },
];
function oneProperty(arreglo, property) {
    const newArray = [];
    arreglo.forEach((element) => {
        newArray.push({ [property]: element[property] });
    });
    return newArray;
}
// console.log(oneProperty(arreglo, 'edad'));
// console.log(oneProperty(arreglo, 'name'));

// 16 - Guerra de palabras
const abc = {
    a: 1,
    b: 2,
    c: 3,
    d: 4,
    e: 5,
    f: 6,
    g: 7,
    h: 8,
    i: 9,

```

```
j: 10,  
k: 11,  
l: 12,  
m: 13,  
n: 14,  
o: 15,  
p: 16,  
q: 17,  
r: 18,  
s: 19,  
t: 20,  
u: 21,  
v: 22,  
w: 23,  
x: 24,  
y: 25,  
z: 26,  
};
```

```
function warWord(string1, string2) {  
  let sumString1 = 0;  
  let sumString2 = 0;  
  
  for (let index = 0; index < string1.length; index++) {  
    sumString1 += abc[string1[index]];  
  }  
  for (let index = 0; index < string2.length; index++) {  
    sumString2 += abc[string2[index]];  
  }  
  
  return sumString1 > sumString2 ? string1 : string2;  
}
```

```
// console.log(warWord('hola', 'chau'));  
// console.log(warWord('love', 'friendship'));
```

```
// 17 - Prefijos telefonicos  
const prefijos = [54, 55, 56, 57, 58];  
const paises = ['argentina', 'brasil', 'chile', 'colombia', 'venezuela'];
```

```
function validarPrefijo(number) {  
  // Crear nuevo objeto  
  const prefijoPais = {};  
  prefijos.forEach((element, index) => {  
    prefijoPais[element] = paises[index];  
  });  
  
  let stringNumber = number.toString();  
  let prefixNumber = `${stringNumber[0]}${stringNumber[1]}`;  
  
  return prefijoPais[prefixNumber] !== undefined  
    ? `Este número pertenece a ${prefijoPais[prefixNumber]}`  
    : `Este número no pertenece a nuestros países`;  
}  
// console.log(validarPrefijo(5412345678));
```

```

// console.log(validarPrefijo(5712345678));
// console.log(validarPrefijo(8012345678));

// 18 - {value:key} kelue: vay
var prueba = {
  nombre: 'santi',
  edad: 22,
  nacionalidad: 'de otro planeta',
  documento: 44444444,
};

function reverseKeys(object) {
  if (typeof object !== 'object') {
    return `error, input can't be a ${typeof object}`;
  } else if (Array.isArray(object)) {
    return `error, input can't be an Array`;
  }

  const keysArray = Object.keys(object);

  const newObject = {};
  keysArray.forEach((element) => {
    if (typeof object[element] === 'string') {
      newObject[object[element]] = element;
    } else {
      newObject[element] = object[element];
    }
  });
  return newObject;
}

// console.log(reverseKeys(344));
// console.log(reverseKeys([]));
// console.log(reverseKeys('hola'));
// console.log(reverseKeys(prueba));

// 19 - Palindromo
function palindromo(string) {
  let isPalyndrome = true;
  let middle;
  if (string.length % 2 === 0) {
    // Palabra de largo par
    middle = string.length / 2 - 1;
  } else {
    // Palabra impar
    middle = (string.length - 1) / 2 - 1;
  }

  for (let i = 0; i <= middle; i++) {
    if (
      string[i].toLowerCase() !== string[string.length - 1 - i].toLowerCase()
    ) {
      isPalyndrome = false;
    }
  }
}

```

```

    return isPalyndrome;
}
// console.log(palindromo('anilina'));
// console.log(palindromo('Ana'));
// console.log(palindromo('Enrique'));

// 20 - Rotación de arreglo a la derecha
function rotar(array, number) {
    const newArray = [];
    const firstRotationElement = array.length - number;

    for (let i = firstRotationElement; i < array.length; i++) {
        newArray.push(array[i]);
    }

    for (let j = 0; j < array.length - number; j++) {
        newArray.push(array[j]);
    }

    return newArray;
}
// console.log(rotar([1, 2, 3, 4, 5], 2));
// console.log(rotar([1, 2, 3, 4, 5], 3));

// 21 - Consecutivos similares
function consecutivosSimilares(string) {
    let sumConsecutives = 0;

    for (let i = 1; i < string.length; i++) {
        if (string[i] === string[i - 1]) {
            sumConsecutives++;
        }
    }

    return sumConsecutives;
}

// console.log(consecutivosSimilares('AAAA'));
// console.log(consecutivosSimilares('BBBBB'));
// console.log(consecutivosSimilares('ABABABAB'));
// console.log(consecutivosSimilares('BABABA'));
// console.log(consecutivosSimilares('AAABBB'));

```