

2020

# ARRAYS: Introducción a la Informática

<JUAN CAMILO RODRIGUEZ LIBREROS>

NOVIEMBRE DE 2020

# 1 CONTENIDO

---

1	CONTENIDO.....	1
2	PRESENTACIÓN.....	2
3	ARRAYS PARTE 1.....	3
4	ARRAYS PARTE 2.....	17

## 2 PRESENTACIÓN

---

La siguiente monografía tiene como fin presentar el tema de arrays dictado en la clase de introducción a la informática.

AUTOR: <JUAN CAMILO RODRIGUEZ LIBREROS >

<CÓDIGO 1005894155>

<CORREO juancamilo.rodriguez1@utp.edu.co>

<GITHUB <https://github.com/JuanCRL1>>

### 3 ARRAYS PARTE 1

---

Los arrays o vectores son zonas de almacenamiento de datos pueden ser de cualquier tipo números, cadenas, símbolos etc.

En esta primer parte se va a mostrar un taller desarrollado en el entorno repl.it donde vamos a profundizar un poco los arrays y algunas funciones que tienen estos.

```
// Definiendo un array vacío
var numeros = [];

// Asignando valores a cada posición del array, mediante un
ciclo
for (var i = 0; i < 100; ++i) {
  numeros[i] = i+1;
}

// Mostrando el array en la pantalla
texto("(:) Array con los números desde el 1 hasta el 100",
2)
for (var i = 0; i < 100; i++) {
  texto(numeros[i] + "-");
}
```

Como vemos para definir un array creamos una variable con el nombre que sea en este caso es números y se le pone un igual y unos corchetes dentro de este puede ir el tamaño del array (cantidad de datos que va a tener) o puede ser vacío como es el caso de este.

para llenarlo se crea un ciclo for con una variable i que empieza desde 0 hasta llegar a 100, se le asigna a los corchetes vacíos esta variable i y se le suma 1 para que el vector quede lleno automáticamente con los números del 1 al 100; para mostrar el array se crea otro ciclo for que recorra las posiciones del array en este caso hasta 100 y se muestra con una salida de texto.

(::) Array con los números desde el 1 hasta el 100

1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-  
28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-  
51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-  
74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-  
97-98-99-100-

Al finalizar esta sería la salida del array lleno.

En esta otra parte vamos a ver como se pueden sumar los datos de un array.

Para ello se puede hacer manualmente como se ve acá:

```
var numeros = [1,2,3,4,5];

var suma = numeros[0] + numeros[1] + numeros[2] + numeros[3]
+ numeros[4];

texto("", 2);
texto("::) Suma manual de los elementos del array: " +
numeros, 2)
texto("Suma = " + suma, 2);
```

Se crea una variable suma, se pone el nombre del array con cada posición en de cada elemento sumándolos.

Al final esta sería la salida:

(::) Suma manual de los elementos del array: 1,2,3,4,5

Suma = 15

Este proceso puede ser mas simple utilizando ciclos pues si hay un array con 1000 elementos o más, poner cada posición sería más difícil o agotador a continuación se mostrará el código utilizando ciclos:

```

var numeros = [1,2,3,5,8,13,21];

var suma = 0;

for (var i = 0; i < numeros.length; ++i)
{
    suma += numeros[i];
}

texto("(:) Suma de elementos del array: " + numeros + "
utilizando ciclos", 2);
texto("Suma = " + suma, 2);

```

Como se aprecia se crea el array y una variable suma que empieza desde 0, después se hace un ciclo for con una variable i que recorra el array desde la posición 0 hasta la última posición luego se pone la instrucción que la variable suma sea igual a la suma de cada elemento del array en la posición i (de 0 hasta la última posición), esta sería la salida:

(:) Suma de elementos del array: 1,2,3,5,8,13,21 utilizando ciclos

Suma = 53

Ahora vamos a ver como se separa una frase por palabras:

```

var frase = "Buen dia colombia el reporte de hoy va a ser
sobre economia";

var palabras = frase.split(" ");

texto("(:) Se divide una frase en palabras. La frase es =
'" + frase + "'", 2);
for (var i = 0; i < palabras.length; ++i)
{
    texto("Palabra: " + i + ": " + palabras[i], 1);
}

texto("", 1);

```

Vemos que se crea una variable frase donde se ve una cadena ingresada, después se crea una variable palabras que excluye los espacios en blanco para poder contar las palabras,

después se hace un ciclo for con una variable i que recorra la cantidad de palabras para poder así dividir las, Esta es la salida del programa:

(::) Se divide una frase en palabras. La frase es = 'Buen día colombia el reporte de hoy va a ser sobre economia'

Palabra: 0: Buen  
Palabra: 1: dia  
Palabra: 2: colombia  
Palabra: 3: el  
Palabra: 4: reporte  
Palabra: 5: de  
Palabra: 6: hoy  
Palabra: 7: va  
Palabra: 8: a  
Palabra: 9: ser  
Palabra: 10: sobre  
Palabra: 11: economia

Para buscar elementos en un array, creamos uno, en este caso se crea uno llamado nombres, lleno obviamente con nombres de personas, se ingresa el nombre que queremos encontrar, creamos un variable posición, se le da una instrucción nombres.indexOf(nombre), para que recorra el array y mire si el nombre ingresado anteriormente se encuentra dentro de este si el nombre esta se pone una salida de texto que dice nombre encontrado en el array y si no, no fue encontrado, a continuación se mostrará el código y su salida:

```

var nombres = ["Santiago", "Sofía", "Juan", "Carlos",
"María"];
texto("(::) Lista de nombres = ", 1);
texto("(-) " + nombres, 2);

nombre = "Ramón";

texto("(-) Se está buscando el nombre: Ramón", 1);

var posicion = nombres.indexOf(nombre);

if (posicion >= 0) {
    texto("(-) Encontrado " + nombre + " en la posición " +
posicion, 2);
}
else {
    texto("(-) " + nombre + " no encontrado en el array.", 2);
}

```

```

(::) Lista de nombres =
(-) Santiago,Sofía,Juan,Carlos,María

(-) Se está buscando el nombre: Ramón
(-) Ramón no encontrado en el array.

```

Este sería otro código el cual nos dice en la primera y última posición donde fue encontrado.



```

var nombres = ["Diego", "Miguel", "Sofía", "Juan", "Ramón",
"Miguel", "Ramón", "Carlos", "Miguel", "María"];

texto("(:) Se busca un nombre en la lista = ", 1);
texto("(-) " + nombres, 2);

var nombre = "Juan";

texto("(-) El nombre a buscar es: " + nombre, 1);

var primeraPosicion = nombres.indexOf(nombre);

texto("(-) Encontrado " + nombre + " en la primera posición
= " + primeraPosicion, 1);

var ultimaPosicion = nombres.lastIndexOf(nombre);

texto("(-) Encontrado " + nombre + " en la última posición
= " + ultimaPosicion, 2);

```

Esta es su salida:

```

(:) Se busca un nombre en la lista =
(-) Diego,Miguel,Sofia,Juan,Ramón,Miguel,Ramón,Carlos,Miguel,María

(-) El nombre a buscar es: Juan
(-) Encontrado Juan en la primera posición = 3
(-) Encontrado Juan en la última posición = 3

```

Como vemos Juan solo estaba una sola vez entonces la primer y última posición es igual.

Para unir un array en una sola cadena se utiliza la función join(), se crea el array y en otra variable se guarda la cadena que va dar del resultado de la unión de todos los elementos del array para entender esto se mostrara el código y su salida:

```
var nombres = ["David", "Sofía", "Ramón", "Carlos",
"Miguel", "Valentina"];

var nombres_union = nombres.join();

texto("(:) Unión de un array utilizando join. Todos sus
elementos se unen en una sola cadena: ", 2)
texto(nombres_union, 2)
```

(:) Unión de un array utilizando join. Todos sus elementos se unen en una sola cadena:

David,Sofía,Ramón,Carlos,Miguel,Valentina

También se puede utilizar el `.toString` así como en este código:

```
nombres_union = nombres.toString();
texto("(:) Utilización del método toString para obtener el
string equivalente de un array", 2);
texto(nombres_union, 2);
```

Para concatenar dos arrays se utiliza la función `concat`, se crean los dos arrays y en otra variable se guarda la concatenación de los dos así como se muestra en el código:

```
var cadena_1 = ["Juan", "Carlos", "Oscar", "Daniel", "Ana"];
var cadena_2 = ["Ramón", "Sofía", "Victor"];

texto("(:) Concatenación de arrays utilizando concat", 2);

texto("(-) Primer array = " + cadena_1, 1);
texto("(-) Segundo array = " + cadena_2, 2);

texto("(-) Primera forma de concatenacion. cadena_2 se
concatena a cadena_1", 1);
var concatenacion = cadena_1.concat(cadena_2);
texto("(-) " + concatenacion, 2);

concatenacion = cadena_2.concat(cadena_1);

texto("(-) Segunda forma de concatenacion. cadena_1 se
concatena a cadena_2", 1);
texto("(-) " + concatenacion, 2);
```

Esta es la salida:

(::) Concatenación de arrays utilizando concat

(-) Primer array = Juan,Carlos,Oscar,Daniel,Ana

(-) Segundo array = Ramón,Sofía,Victor

(-) Primera forma de concatenacion. cadena\_2 se concatena a cadena\_1

(-) Juan,Carlos,Oscar,Daniel,Ana,Ramón,Sofía,Victor

(-) Segunda forma de concatenacion. cadena\_1 se concatena a cadena\_2

(-) Ramón,Sofía,Victor,Juan,Carlos,Oscar,Daniel,Ana

Para dividir un array por partes se utiliza la función .splice, se le pone hasta donde dividir y se muestra por pantalla a continuación se muestra el código y la salida:

```
var concatenacion = ["Miguel","Oscar","Antonio","Ramón",
"Valentina","David","María"];

texto("(::) Ilustración del método de división por partes
utilizando splice", 2)
texto("(-) El array original = " + concatenacion, 1);

// Extrae una parte del array y la coloca en 'unaParte'
var unaParte = concatenacion.splice(3,3);

// En concatenacion queda el excedente, al cual se le
denomina 'otraParte'
var otraParte = concatenacion;

texto("(-) Una parte splice(3,3) = " + unaParte, 1); //
Ramón,Sofía,Daniel
texto("(-) Otra parte (lo que sobra) = " + otraParte, 2);
```

(::) Ilustración del método de división por partes utilizando splice

(-) El array original = Miguel,Oscar,Antonio,Ramón,Valentina,David,María

(-) Una parte splice(3,3) = Ramón,Valentina,David

(-) Otra parte (lo que sobra) = Miguel,Oscar,Antonio,María

Para agregar mas elementos a un array se utiliza la función `.push` y el elemento a poner, como se muestra en el siguiente código:

```
var numeros = [1,2,3,4,5];
texto("(:) Agregando elementos a un array (al final)", 2)
texto("(-) Array antes de agregar un elemento al final: " +
numeros, 1); // 1,2,3,4,5
numeros.push(6);
texto("(-) Array después de agregar el elemento: " +
numeros, 2); // 1,2,3,4,5,6
```

Esta sería la salida:

```
(:) Agregando elementos a un array (al final)
(-) Array antes de agregar un elemento al final: 1,2,3,4,5
(-) Array después de agregar el elemento: 1,2,3,4,5,6
```

Con la función `unshift` también se agregan más elementos la diferencia entre este y el anterior es que pone al principio el dato y el otro lo pone al final a continuación se muestra el código y el resultado:

```
texto("(:) Agregando elementos al comienzo de un array.
Utilización de unshift", 2);

var numeros = [2,4,6,8];
texto("(-) Números inicial = " + numeros, 1); // 2,3,4,5

var nuevo_numero = 1;
numeros.unshift(nuevo_numero);
texto("(-) Agregado el 1: " + numeros, 2); // 1,2,3,4,5

numeros = [3,5,7];

texto("(-) Números inicial = " + numeros, 1); // 3,4,5

numeros.unshift(1,2);
texto("(-) Agregados dos valores, el 1 y el 2. Nuevo array
= " + numeros, 2); // 1,2,3,4,5
```

(::) Agregando elementos al comienzo de un array. Utilización de unshift

(-) Números inicial = 2,4,6,8

(-) Agregado el 1: 1,2,4,6,8

(-) Números inicial = 3,5,7

(-) Agregados dos valores, el 1 y el 2. Nuevo array = 1,2,3,5,7

También si se quiere poner elementos de un array se utiliza el .splice:

```
var numeros = [1,2,3,7,8,9];
```

```
texto("(::) Agregando elementos en la mitad de un array. ",
2);
```

```
texto("(-) Array Números (antes) = " + numeros, 1); // 1,2,
3,7,8,9
```

```
var nuevos_elementos = [4,5,6];
numeros.splice(3,0,nuevos_elementos);
```

```
texto("(-) Números (después). Se agrega [4,5,6] a partir de
la tercera posición = " + numeros, 2); // 1,2,3,4,5,6,7,8,9
```

Salida:

(::) Agregando elementos en la mitad de un array.

(-) Array Números (antes) = 1,2,3,7,8,9

(-) Números (después). Se agrega [4,5,6] a partir de la tercera posición =  
1,2,3,4,5,6,7,8,9

Para eliminar el ultimo elemento de un array se utiliza .pop, para hacerlo con el primer dato se utiliza la función unshift y para eliminar el elemento en la mitad .splice a continuación se mostraran los códigos y su salida:

(Con .pop)

```

var numeros = [1,2,3,4,5,9];
texto("(:) Removiendo un dato al final del array", 2);

texto("(-) Array original = " + numeros, 1);

numeros.pop();
texto("(-) Array sin el último elemento: " + numeros, 2);
// 1,2,3,4,5

```

(:) Removiendo un dato al final del array

(-) Array original = 1,2,3,4,5,9

(-) Array sin el último elemento: 1,2,3,4,5

(Con unshift)

```

var numeros = [9,1,2,3,4,5];
texto("(:) Removiendo elementos del comienzo de un array.
Se utiliza 'unshift'", 2);

texto("(-) Valor original de Números = " + numeros, 1);
numeros.shift();
texto("(-) Se elimina el primer elemento del array: " +
numeros, 2); // 1,2,3,4,5

```

(:) Removiendo elementos del comienzo de un array. Se utiliza 'unshift'

(-) Valor original de Números = 9,1,2,3,4,5

(-) Se elimina el primer elemento del array: 1,2,3,4,5

(con .splice)

```

var numeros = [1,2,3,100,200,300,400,4,5];

texto("(:) Eliminando elementos en la mitad de un array.
Se utiliza splice, según se aprecia en la instrucción", 2);
texto("(-) Números (antes) = " + numeros, 1); // 1,2,3,100,
200,300,400,4,5

numeros.splice(3,4);

texto("(-) Números (después) = " + numeros, 2); // 1,2,3,4,5

```

(::) Eliminando elementos en la mitad de un array. Se utiliza splice, según se aprecia en la instrucción

(-) Números (antes) = 1,2,3,100,200,300,400,4,5

(-) Números (después) = 1,2,3,4,5

Para invertir un array se utiliza la función .reverse como se muestra a continuación:

```
var numeros = [1,2,3,4,5];
texto("(::) Invertiendo arrays: Número = " + numeros, 2);

numeros.reverse();

texto("(-) Array invertido: " + numeros, 2); // 5,4,3,2,1
```

Salida:

(::) Invertiendo arrays: Número = 1,2,3,4,5

(-) Array invertido: 5,4,3,2,1

Para ordenar un array se utiliza la función .sort a continuación se muestra el código y la salida:

```
var nombres = ["David","Miguel","Carlos","María","Bernardo",
"Ramón"];
texto("(::) Ordenando el array = " + nombres, 2);

nombres.sort();
texto("(-) Array ordenado = " + nombres, 2);
// Bernardo, Carlos, David, María, Miguel, Ramón
```

(::) Ordenando el array = David,Miguel,Carlos,María,Bernardo,Ramón

(-) Array ordenado = Bernardo,Carlos,David,María,Miguel,Ramón

Si se utiliza para ordenar números quedaría así:

```

texto("(:) Ordenamiento de números. En su forma básica,
los números se ordenan mal", 2);

var numeros = [3,1,2,100,4,200];
texto("(-) Array original = " + numeros, 1);

numeros.sort();
texto("(-) Array ordenado = " + numeros, 2); // 1,100,2,200,
3,4

```

(:) Ordenamiento de números. En su forma básica, los números se ordenan mal

(-) Array original = 3,1,2,100,4,200

(-) Array ordenado = 1,100,2,200,3,4

Como vemos queda mal organizado lo que se tiene que hacer es comparar cada numero para que quede bien se crea una función como se muestra a continuación:

```

function comparar(num1, num2) {
return num1 - num2;
}

texto("(:) Ordenamiento numérico corregido", 2);

var numeros = [3,1,2,100,4,200];

texto("(-) Array original = " + numeros, 1);

numeros.sort(comparar);

texto("(-) Array ordenado = " + numeros, 2); // 1,2,3,4,
100,200

```

Salida:

(:) Ordenamiento numérico corregido

(-) Array original = 3,1,2,100,4,200

(-) Array ordenado = 1,2,3,4,100,200



Si se quiere mostrar el cuadrado de cada número en un array se utiliza `foreach` entonces se crea una función que eleve al cuadrado un número y se pone dentro del `foreach`(cuadrado) así como se muestra a continuación:

```
function cuadrado(numero) {
  |   texto(numero.toString() + " --- " + (numero * numero),
  |   1);
}
var numeros = [1,2,3,4,5,6,7,8,9,10];

texto("(::) Cuadrados, utilizando forEach", 2);

numeros.forEach(cuadrado);
```

Salida:

(::) Cuadrados, utilizando forEach

```
1 --- 1
2 --- 4
3 --- 9
4 --- 16
5 --- 25
6 --- 36
7 --- 49
8 --- 64
9 --- 81
10 --- 100
```

Eso es casi todo lo que se puede hacer utilizando arrays y funciones en la próxima parte se explicaran más funciones.

## 4 ARRAYS PARTE 2

---

En esta parte vamos a ver un poco sobre arrays de 2 dimensiones o matrices.

En JavaScript no se pueden crear matrices como en C++ o Java donde se pone el nombre de la matriz y dos corchetes que indican las filas y columnas, para ello se crean arrays con una longitud determinada y se agrega en otro array que va a contener los arrays creados.

A continuación se muestra el código de como hacerlo y su salida:

```
var arreglo1 = ["Juan", 24, 18000]; // Fila 0
var arreglo2 = ["Ana", 30, 30000]; // Fila 1
var arreglo3 = ["Carla", 28, 41000]; // Fila 2
var arreglo4 = ["Carlos", 31, 28000]; // Fila 3
var arreglo5 = ["Pedro", 29, 35000]; // Fila 4

var salario = [arreglo1, arreglo2, arreglo3, arreglo4,
arreglo5];

var notas = [
    ['Juan', 4],
    ['Sofía', 5],
    ['Carlos', 3],
    ['Alberto', 5],
    ['María', 5]
];
```

### (::) IMPRIMIENDO UN ARREGLO EN DOS DIMENSIONES

```
Juan --- 24 --- 18000 ---
Ana --- 30 --- 30000 ---
Carla --- 28 --- 41000 ---
Carlos --- 31 --- 28000 ---
Pedro --- 29 --- 35000 ---
```

Para imprimirlo se crea un ciclo for con una variable que recorra las filas y otro for que recorra las columnas para poder mostrarlo como se ve en este código:

```

texto("(:) IMPRIMIENDO UN ARREGLO EN DOS DIMENSIONES", 2);

// Este ciclo es para las filas
for (var fila = 0; fila < salario.length; fila++) {
    // Este ciclo es para las columnas
    for (var columna = 0; columna < salario[fila].length;
        columna++) {
        // Aquí se accede a cada elemento del array
        texto( salario[fila][columna] + " --- " );
    }
    texto("", 1);
}

```

La salida seria la imagen anterior.

Para agregar un elemento a una matriz se crea esta con los datos y se utiliza la función .push (mencionada en la parte 1 de arrays) con el dato a ingresar, a continuación se mostrará como:

```

texto("(:) AGREGANDO ELEMENTOS", 2);

texto("(-) Agregando un elemento al final:", 1);
texto("(-) salario.push(['Julio', 32, 35000]); ", 2);

salario.push(['Julio', 32, 35000]);

```

### (:) AGREGANDO ELEMENTOS

(-) Agregando un elemento al final:  
 (-) salario.push(['Julio', 32, 35000]);

Para eliminar elementos se utiliza la función mencionada anteriormente en arrays parte 1 .pop a continuación se mostrará el ejemplo:

```
texto("(:) ELIMINANDO ELEMENTOS", 2);  
texto("(-) Eliminando el elemento final:", 1);  
texto("(-) salario.pop();", 2);
```

```
salario.pop();
```

```
(:) ELIMINANDO ELEMENTOS
```

```
(-) Eliminando el elemento final:
```

```
(-) salario.pop();
```