

LÓGICA DE PROGRAMACIÓN

Arrays

Profesor: Jonathan Cruz

Email: jonathan.cruz@davinci.edu.ar

Arrays: introducción

¿Qué es una variable?

- Es un espacio reservado de la memoria que tiene un nombre de referencia que sirve para guardar y manipular una determinada información.
 - Ejemplo:
let nombre = 'John';
let apellido = 'Do';

| Memoria RAM | |
|-------------|----------|
| nombre | apellido |
| 'John' | 'Do' |

Arrays: introducción

¿Qué es un array o matriz?

- Es un espacio de la memoria con un nombre de referencia para identificarlo, que se **divide** en sub-espacios.
- Cada **sub-espacio** se identifica por su **índice numérico**, que contendrá un valor.
- El **primer índice** siempre será el número **0**.
 - Ejemplo:
let nombre = 'John', apellido = 'Do';
let materias = ['Lógica de programación', 'Programación I'];

| Memoria RAM | | | |
|-------------|----------|--------------------------|------------------|
| nombre | apellido | materias | |
| 'John' | 'Do' | 0 | 1 |
| | | 'Lógica de programación' | 'Programación I' |

Arrays: introducción

¿Para qué sirven los arrays o matrices?

- Los arrays **reducen** la cantidad de variables a utilizar en un programa.
- Los arrays son **dinámicos**, durante el programa se le pueden agregar o quitar elementos (con las variables comunes no se puede).
- Los arrays se pueden **recorrer** de principio a fin, o como se lo desee (con las variables comunes no se puede).
- Los arrays son una manera de tener en un solo elemento una **fuentes de múltiples datos**.

Arrays: creación

¿Cómo se crea un array o matriz?

- Existen dos formas de crear un array:
 - Usando la versión constructora:
`let datos = new Array();`
 - Usando la versión literal:
`let datos = [];`
- Generalmente utilizaremos la versión literal a la hora de crearlos.
- **ATENCIÓN:**
 - ***¡Si no se le indica a la variable que será un array, estaremos trabajando con una variable común... Por lo tanto, toda operación del tipo array a realizar posteriormente, causará errores en la ejecución del código!***

Arrays: contenido

¿Cómo se le guardan valores a un array o matriz?

- Existen tres formas de agregar contenido a un array, sin importar si fue creado con la versión constructora o literal:
 - Separando con coma los valores entre los corchetes o paréntesis.
let datos = new Array('Cadena', 21, true, 'ABC', [], null);
let datos = ['Cadena', 21, true, 'ABC', [], null];
- Como se puede apreciar, un array permite guardar distintos tipos de datos en la misma variable.

Arrays: contenido

¿Cómo se le guardan valores a un array o matriz?

- Existen tres formas de agregar contenido a un array, sin importar si fue creado con la versión constructora o literal:
 - Creando un array vacío e ir guardando valores en cada índice indicado:

```
let datos = [];  
datos[0]   = 'Cadena';  
datos[1]   = 21;  
datos[2]   = true;  
datos[3]   = 'ABC';  
datos[4]   = [];  
datos[5]   = null;
```
- Es indistinto si se crea con la versión constructora o literal.
- Se podrían saltar índices si se deseara. Los índices saltados se crean automáticamente.
- Por ser sub-espacios declarados sin inicializar, su valor será ***undefined***.

Arrays: contenido

¿Cómo se le guardan valores a un array o matriz?

- Existen tres formas de agregar contenido a un array, sin importar si fue creado con la versión constructora o literal:
 - Creando un array vacío y cargando su contenido mediante el método **push**:

```
let datos = [];  
datos.push('Cadena');  
datos.push(21);  
datos.push(true);  
datos.push('ABC');  
datos.push([]);  
datos.push(null);
```
- Es indistinto si se crea con la versión constructora o literal.
- El método crea automáticamente un nuevo índice al final de array y guarda el valor pasado entre paréntesis (más adelante veremos este método en detalle).

Arrays: contenido

¿Cómo se puede mostrar el contenido de un array o matriz?

- Se puede mostrar todo el contenido de un array directamente mediante algún método de "impresión", como ser alert, document.write, console, etcétera (esta no sería la forma más "elegante").
- Se puede mostrar el contenido particular, seleccionando cada índice que se desea "imprimir".

```
let datos = ['ABC', 1985, false];  
console.info(datos);           // Saldrá: ABC,1985,false  
console.info(datos[0]);        // Saldrá: ABC  
console.info(datos[2]);        // Saldrá: false  
console.info(datos[3]);        // Saldrá: undefined
```

- Veamos un ejemplo para mayor comprensión

Arrays: diferencia entre constructor y literal

¿Cuál es la diferencia entre el literal y el constructor?

- Se sabe que si se colocan valores separados por coma entre los paréntesis o corchetes, cada uno se almacenará en un índice, comenzando desde el 0.
- Pero si al **crear un array de la manera constructora**, solamente se le coloca **un número entre sus paréntesis**, se estará definiendo la **longitud inicial** del array (es decir, cuántos índices ya debe tener reservados).
- Hasta que esos espacios no se inicializan, cada índice tendrá como valor **undefined**.
- Esto se suele utilizar en los casos que se sabe la longitud a la hora de trabajar con un array para evitar consumo de memoria.
- La realidad es que hoy en día esto no es necesario.

Arrays: equivocaciones más comunes al comenzar a trabajar con arrays

¿En qué nos solemos equivocar ni bien comenzamos a trabajar con arrays?

- A la hora de crearlos:

- `let datos[];`

Falta el igual:

- `let datos = Array();`

Falta el new:

- `let datos[] = [];`

Sobran los primeros corchetes:

¿Qué está mal?

`let datos = [];`

¿Qué está mal?

`let datos = new Array();`

¿Qué está mal?

`let datos [] = [];`

- A la hora de guardar contenido:

- `let datos; datos[0] = 'valor';`

Nunca inicializamos la variable como un array:

¿Qué está mal?

`let datos = [];`

Arrays: recorrido

Ciclo for:

- Sirve para recorrer un array numérico, utilizando una variable para recorrer cada índice.
- Generalmente, se debe ir desde el índice 0 hasta la longitud (**length**) del array exclusive.

Ciclo for in:

- Recorre cada índice, que será lo que se guarda en la variable declarada entre paréntesis. Su sintaxis es:

```
for (let indice in array) {  
    console.info(indice, array[indice]);  
}
```

Ciclo for of:

- A diferencia del anterior, recorre cada valor, que será lo que se guarda en la variable declarada entre paréntesis. Sirve para recorrer valores en lugar de índices. Su sintaxis es:

```
for (let valor of array) {  
    console.info(valor);  
}
```

- Veamos un ejemplo para mayor comprensión.

Arrays

Practiquemos un poco:

- **Ejercicio de práctica para hacer en conjunto:**

Crear un array precargado con 10 colores HTML válidos.

Con un ciclo de repetición for recorrer cada índice:

- Por cada índice el valor leído será utilizado como "background-color" de un span; el índice será el contenido del span.
- Cada span se irá agregando a la variable html.

El contenido de html sera mostrado dentro del div.galeria.

Arrays

Practiquen un poco:

- **Ejercicio de práctica para que realicen en clase, luego lo vemos en conjunto:**

Partiendo del array discos, que contiene en sus índices los nombres de archivos de imagen de discos, se debe recorrer con un ciclo de repetición for, generando por cada índice:

- Una etiqueta div.
- Dentro de la etiqueta div, crear una etiqueta img cuyo src y alt sean el nombre del archivo (ver la ruta a colocar en el src).
- Luego de la etiqueta img, colocar una etiqueta p que muestre el nombre del archivo.
- Cada bloque debe irse guardando en la variable html. El contenido de html será mostrado dentro del div.galeria.

Arrays

Practiquen un poco más:

- **Ejercicio de práctica para que realicen en clase, luego lo vemos en conjunto:**

Ahora la información de los discos se repartirá entre 3 arrays:

- imagenes: contiene el nombre de las imágenes.
- nombres: contiene los nombres de los discos.
- lanzamientos: contiene el año de lanzamiento.

Con un ciclo de repetición for, se deberá generar por cada índice:

- Una etiqueta div, que contendrá una etiqueta a cuyo href sea la ruta del archivo.
- Dentro de la etiqueta a, crear una etiqueta img cuyo src sea el nombre del archivo (ver la ruta a colocar en el src) y el alt el nombre del disco.
- Luego de la etiqueta a, colocar una etiqueta h3 que muestre el nombre del disco.
- Luego de la etiqueta h3, colocar una etiqueta p que muestre el año de lanzamiento del disco.
- Cada bloque debe irse guardando en la variable html. El contenido de html será mostrado dentro del div.galeria.