



Arreglos y Objetos



**Activen las cámaras los que puedan y
pasemos asistencia**

***Utilizar arreglos y objetos
como fuente de información
para poblar dinámicamente
los datos de una página web.***

- Unidad 1: Introducción a JavaScript.
- Unidad 2: Condiciones.
- Unidad 3: Funciones.
- Unidad 4: Arreglos y objetos.
- Unidad 5: Métodos de arreglos.
- Unidad 6: APIs



Te encuentras aquí





Inicio

{desafío}
latam_



/ Acceder a valores almacenados en un arreglo */*

/ Mostrar los elementos de un arreglo por consola utilizando un ciclo */*

/ Agregar elementos a una lista HTML a partir de un arreglo de Strings */*

/ Crear un objeto con propiedades a partir de la interpretación de un texto */*

/ Crear el template de una tarjeta que contenga propiedades de un objeto */*

/ Agregar los elementos de una lista HTML a partir de un arreglo de objetos */*

Objetivos

Activación de conceptos

Contesta la pregunta correctamente y gana un punto

Instrucciones:

- Se realizará una pregunta, el primero en escribir “YO” por el chat, dará su respuesta al resto de la clase.
- El docente validará la respuesta.
- En caso de que no sea correcta, dará la oportunidad a la segunda persona que dijo “Yo”.
- Cada estudiante podrá participar un máximo de 2 veces.
- Al final, el/la docente indicará el 1º, 2º y 3º lugar.
- Esta actividad no es calificada, es solo una dinámica para recordar los conceptos clave para abordar esta sesión.



Pregunta 1

¿Cuántos parámetros tiene esta función?

```
function agregarBordes(elementId, color = 'red') {  
    elemento = document.querySelector('#'  
+ elementId);  
    elemento.style.border="dashed 3px " +  
color  
}
```

Pregunta 2

¿Si a la función agregarBordes la llamamos como agregarBordes('p-1', 'black') de qué color se mostrará el borde?

```
function agregarBordes(elementId,  
color = 'red') {  
    elemento =  
document.querySelector('#' + elementId);  
    elemento.style.border="dashed 3px "  
+ color  
}
```


Pregunta 3

¿Cómo se llama a la siguiente función?

```
suma = (a, b) => a + b
```

Pregunta 4

¿Qué alcance tiene la variable b?

```
a = 5  
if (a == 5) {  
  var b = 8  
}
```

Activación de conceptos



Primer lugar:



Segundo lugar:



Tercer lugar:



Desarrollo

{desafío}
latam_



/* Arreglos */

¿Qué son los arreglos?

- Los arreglos nos ayudan a almacenar un conjunto de datos en una misma variable.

Por ejemplo:

```
const sucursales = ["Rebeca Mate 18", "Libertad 5", "Av manquehue Sur 31"]
```

```
const codigosDeDescuento = ["ABC", "FE1", "8IA"]
```

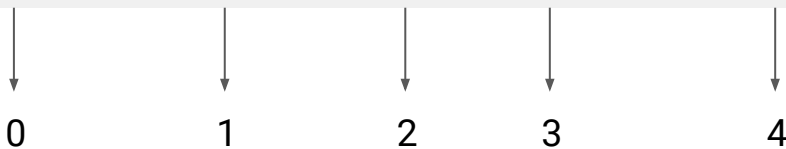
```
const clientesPremium = ["Andrea", "Fernando", "Mariel"]
```

- Para crear un arreglo se deben ocupar los corchetes [], y posteriormente declarar los elementos separados por comas.

Acceder a los elementos de un arreglo

- Cada elemento almacenado en un arreglo es accesible por un índice

```
const bancos = ["Estado", "Santander", "BCI", "ITAU", "Scotiabank"]
```



- Para acceder al **primer** elemento, escribimos lo siguiente: `bancos[0]`
- Mientras que para acceder al **último** elemento, escribimos lo siguiente: `bancos[4]`
- Probemos guardando el arreglo en la consola del inspector de elementos y luego escribamos `console.log(bancos[0])`

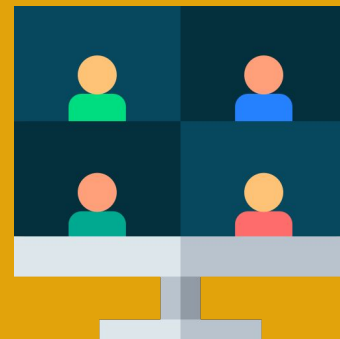
Ejercicio

1. Realicemos el siguiente ejercicio en donde a partir del siguiente arreglo:

```
const nombres = [  
  "Juan",  
  "Luisa",  
  "Fabian",  
  "Jorge",  
]
```

2. Muestra por consola los nombres Juan, Fabián y Jorge.

Demostración



/* Acceder a valores almacenados en un arreglo */ ✓

/* Mostrar los elementos de un arreglo por consola utilizando un ciclo */

/* Agregar elementos a una lista HTML a partir de un arreglo de Strings */

/* Crear un objeto con propiedades a partir de la interpretación de un texto */

/* Crear el template de una tarjeta que contenga propiedades de un objeto */

/* Agregar los elementos de una lista HTML a partir de un arreglo de objetos */

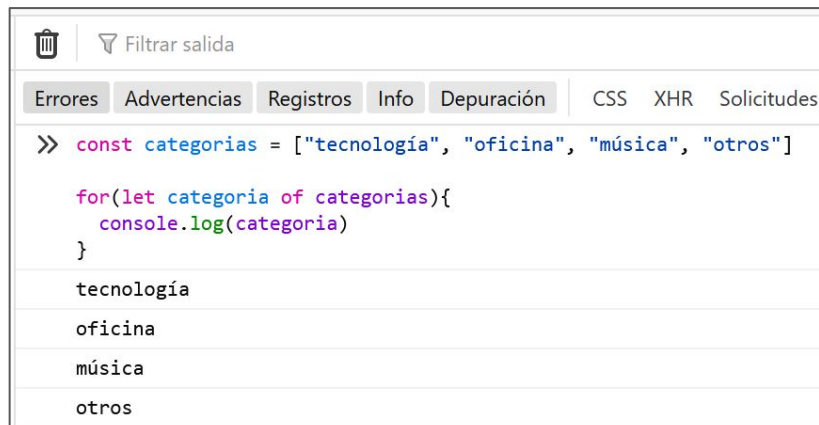
Objetivos

/* Ciclos */

Arreglos y ciclos

- Los ciclos permiten repetir un bloque de código varias veces. Los podemos utilizar para mostrar los datos de un arreglo uno en uno.
- Hay varios tipos de ciclo, veamos un ejemplo con for.

```
const categorias =  
["tecnología", "oficina", "música",  
"otros"]  
  
for(let categoria of categorias){  
  console.log(categoria)  
}
```



The screenshot shows a web browser's developer console with the 'Filtrar salida' (Filter output) search bar at the top. Below the search bar are tabs for 'Errores', 'Advertencias', 'Registros', 'Info', 'Depuración', 'CSS', 'XHR', and 'Solicitudes'. The 'Registros' (Logs) tab is selected, displaying the output of the JavaScript code. The code defines an array 'categorias' with the values 'tecnología', 'oficina', 'música', and 'otros'. A 'for-of' loop iterates over this array, logging each element to the console. The console output shows the following sequence of logs: 'tecnología', 'oficina', 'música', and 'otros'.

```
>> const categorias = ["tecnología", "oficina", "música", "otros"]  
  
for(let categoria of categorias){  
  console.log(categoria)  
}
```

tecnología
oficina
música
otros

Anatomía básica de un ciclo

En donde:

```
const categorias = ["tecnología", "oficina", "música", "otros"]
```

```
for(let categoria of categorias ){  
  console.log(categoria)  
}
```

Palabra
reservada para
crear un for

El arreglo que
será iterado

La variable que
contendrá el
elemento en
cada iteración

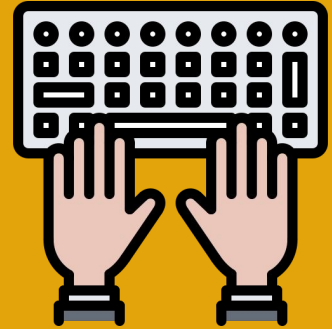
Ejercicio

1. Realicemos el siguiente ejercicio en donde debas mostrar por consola todos los elementos del siguiente arreglo utilizando el ciclo for.

```
const lenguajes = [  
  "JavaScript",  
  "Php",  
  "Ruby",  
  "Python",  
  "Scala",  
  "Go"  
]
```

Ejercicio

¡Manos al teclado!



`/* Acceder a valores almacenados en un arreglo */` ✓

`/* Mostrar los elementos de un arreglo por consola utilizando un ciclo */` ✓

`/* Agregar elementos a una lista HTML a partir de un arreglo de Strings */`

`/* Crear un objeto con propiedades a partir de la interpretación de un texto */`

`/* Crear el template de una tarjeta que contenga propiedades de un objeto */`

`/* Agregar los elementos de una lista HTML a partir de un arreglo de objetos */`

Objetivos

/* Interpolación */

Interpolación vs Concatenación

La interpolación es otra forma de juntar texto con variables.

```
const nombre = "Luis"  
const fecha = "2022-03-25"  
  
const interpolacion = `Hola, mi nombre es ${nombre}, hoy ${fecha} quiero emitir una  
solicitud`  
const concatenacion = 'Hola, mi nombre es ' + nombre + ', hoy ' + fecha + ' quiero emitir una  
solicitud'  
  
console.log(interpolacion)  
console.log(concatenacion)
```

Hola, mi nombre es Luis, hoy 2022-03-25 quiero emitir una solicitud

Sintaxis de la interpolación

La sintaxis que debemos seguir para interpolar es crear un string ocupando las comillas invertidas (```). Dentro de estas podemos escribir texto y variables.

Para utilizar variables debemos escribirlas entre los símbolos de pesos y las llaves `${}`, como se muestra a continuación.

Dentro de las comillas invertidas podemos hacer saltos de línea sin ningún problema.

```
let template = `  
  <h1>${titulo}</h1>  
  <p>${parrafo}</p>  
`
```


**/* Agregar contenido dinámicamente con
arreglos, ciclos e interpolación */**

Agregando elementos dinámicamente

```
<ul id="dynamic-content"> </ul>
<script>
  const data = ['Javiera', 'Camila', 'Francisco', 'Jorge', 'Daniela']
  const d = document.querySelector("#dynamic-content")
  for (let item of data){
    d.innerHTML+= `<li> ${item} </li>`
  }
</script>
```

Se lee como: Por cada uno de los datos en el arreglo actualizamos el DOM del documento agregando el valor anterior + `` ${item} ``

Ejercicio

1. Realicemos el siguiente ejercicio en donde agregarás las notas musicales como elementos en la siguiente lista UL

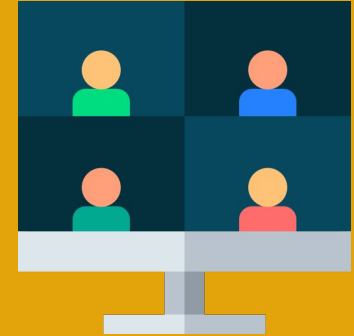
```
<div>
  <h2>Escala mayor de Do:</h2>
  <ul></ul>
</div>
```

Utiliza el siguiente arreglo:

```
const escalaMayorDeDo= [
  "Do",
  "Re",
  "Mi",
  "Fa",
  "Sol",
  "La",
  "Si",
]
```

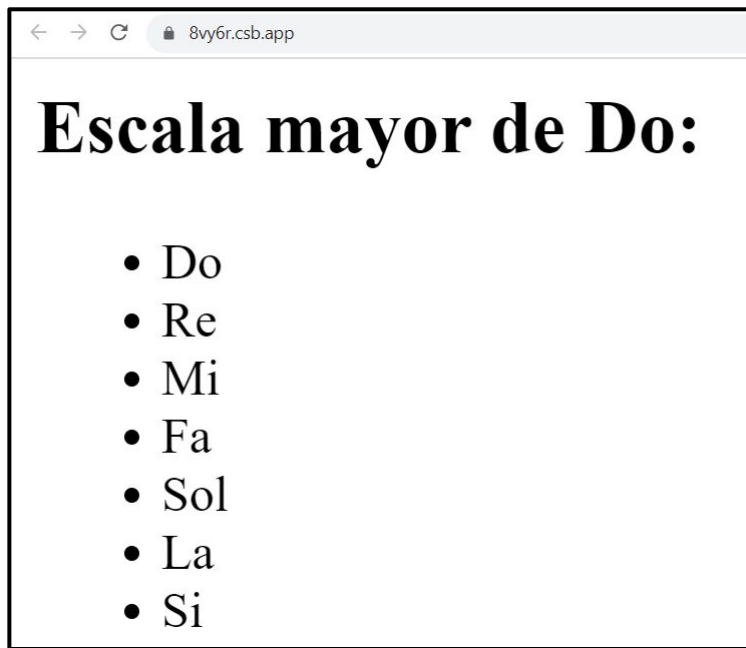
{desafío}
latam_

Demostración

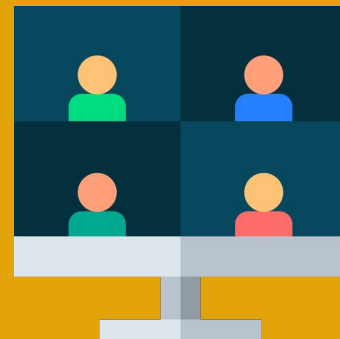


Ejercicio

Debemos obtener el siguiente resultado:



Demostración



Actualizaciones innecesarias de DOM

- Tenemos que tener cuidado con esta línea, cada vez que modificamos con `.innerHTML` se fuerza la modificación del DOM.
- Es mejor modificarlo una sola vez, en lugar de las múltiples iteraciones que tenga el ciclo.

```
d.innerHTML+= `- ${item} </li>`

```

Actualizando una sola vez el DOM

Por cada uno de los datos del arreglo los agregamos a un string. Luego utilizamos ese string para actualizar el DOM una única vez.

```
let data = ['Javiera', 'Camila', 'Francisco', 'Jorge', 'Daniela']
let html = ""
let d = document.querySelector("#dynamic-content")
for (let item of data){
  html += `- ${item} </li>`
}
d.innerHTML = html

```

Ejercicio

1. Actualiza el ejercicio de las notas musicales para evitar renderizar el DOM múltiples veces.

Ejercicio ¡Manos al teclado!



`/* Acceder a valores almacenados en un arreglo */` ✓

`/* Mostrar los elementos de un arreglo por consola utilizando un ciclo */` ✓

`/* Agregar elementos a una lista HTML a partir de un arreglo de Strings */` ✓

`/* Crear un objeto con propiedades a partir de la interpretación de un texto */`

`/* Crear el template de una tarjeta que contenga propiedades de un objeto */`

`/* Agregar los elementos de una lista HTML a partir de un arreglo de objetos */`

Objetivos

/* Objetos */

Creación de objetos

- Los objetos nos permiten agrupar datos en torno a una clave y valor. Por ejemplo, podemos agrupar todas las cualidades de un auto en una misma variable ocupando los objetos, veamos un ejemplo.
- Imaginemos que tenemos un **objeto auto**. Este auto tiene **atributos** como marca, modelo, año y kilometraje.

```
let auto = {  
  marca: "JAC",  
  modelo : "S4",  
  año: 2018,  
  km: 60304  
}
```

Los atributos se separan por coma,
el último no lleva coma.

Accediendo a datos del objeto

Existen 2 maneras de acceder a los datos de un objeto:

- Usando el punto (.)
 - `auto.marca`
- Usando los corchetes []
 - `auto["marca"]`

```
let auto = {  
  marca: "JAC",  
  modelo: "S4",  
  año: 2018,  
  km: 60304  
};  
  
console.log(auto.marca);  
console.log(auto["marca"]);
```

Ejercicio

Según el siguiente objeto:

```
const persona = {  
  nombre: "Pedro",  
  apellido: "Perez",  
  profesion: "Frontend Developer",  
  hobby: "Trekking"  
}
```

- Utilizando console.log muestra el hobby

Ejercicio

¡Manos al teclado!



Creación de templates con datos de un objeto

1. Teniendo el siguiente HTML:

```
<section class="articulos"></section>
```

2. Y el siguiente objeto:

```
const articulo = {  
  id: 31,  
  titulo: "Autos nuevos en Chile",  
  subtitulo: "El mercado de autos se normaliza",  
  descripcion: "No es novedad que los precios  
de los autos usados se han disparado debido  
a la falta en stock de autos nuevos, sin embargo  
puede que esto esté llegando a su fin..."  
};
```

Creación de templates con datos de un objeto

3. Modificamos el HTML

Este string suele llamarse template o plantilla, puesto que es una base para escribir un texto o un HTML de mayor complejidad



```
const articulosSection =  
document.querySelector(".articulos")  
  
articulosSection.innerHTML = `  
  <article class="articulo">  
    <h4>${articulo.titulo}</h4>  
    <h6>${articulo.subtitulo}</h6>  
    <p>${articulo.descripcion}</p>  
    <a  
href="/articulo/${articulo.id}"><button>Ver  
más</button></a>  
  </article>  
`;  
;
```

Creación de templates con datos de un objeto

Con un poco de estilos CSS:

```
.articulo {  
  width: 250px;  
  padding: 10px;  
  background: lightgray;  
}
```

Obtendremos un artículo con contenido dinámico y generado a partir de los datos de un objeto.

Autos nuevos en Chile

El mercado de autos se normaliza

No es novedad que los precios de los autos usados se han disparado debido a la falta en stock de autos nuevos, sin embargo puede que esto esté llegando a su fin...

[Ver más](#)

Ejercicio

Crea un producto para un e-commerce a partir del siguiente objeto:

```
const producto = {  
  id: 43,  
  titulo: "Cafetera magnética",  
  precio: 23990,  
  color: "rojo",  
  src: "...", // ingresa aquí la URL de la imagen,  
  descripción: `Calienta tu café matutino  
    con la nueva tecnología magnética`  
};
```

Ejercicio

¡Manos al teclado!



Ejercicio

1. Crea el HTML con una sección para el producto.
2. Crea una variable para contener la sección del HTML y selecciónala utilizando `document.querySelector`.
3. Crea una variable con el objeto producto mostrado.
4. Crea un template usando la interpolación y los datos del producto.
5. Utiliza el `innerHTML` para agregar los datos al HTML.

Ejercicio

¡Manos al teclado!



`/* Acceder a valores almacenados en un arreglo */` ✓

`/* Mostrar los elementos de un arreglo por consola utilizando un ciclo */` ✓

`/* Agregar elementos a una lista HTML a partir de un arreglo de Strings */` ✓

`/* Crear un objeto con propiedades a partir de la interpretación de un texto */` ✓

`/* Crear el template de una tarjeta que contenga propiedades de un objeto */` ✓

`/* Agregar los elementos de una lista HTML a partir de un arreglo de objetos */`

Objetivos

/* Arreglo de objetos */

Arreglo de objetos

Dentro de un arreglo guardamos la información de varios objetos

Esto se conoce como
arreglo de objetos

```
const productos = [  
  {  
    id: 1,  
    nombre: "Item 1",  
    precio: "12000"  
  },  
  {  
    id: 2,  
    titulo: "Item 2",  
    precio: "14000",  
  }  
]
```

Arreglo de objetos

Agregando productos a la página web a partir de un arreglo de productos

```
const productos = [  
  {  
    id: 1,  
    nombre: "Item 1",  
    precio: "12000"  
  },  
  {  
    id: 2,  
    nombre: "Item 2",  
    precio: "14000"  
  }  
]
```

```
let html = ""  
for (let producto of productos){  
  html += `<div id="${producto.id}" class="producto">  
    <h2> ${producto.nombre} </h2>  
    <p> ${producto.precio} </p>  
  </div> `  
}  
  
const ele = document.querySelector("#productos")  
ele.innerHTML = html
```

Ejercicio

Crea una galería de iconos utilizando el siguiente arreglo de objetos:

```
const iconos = [  
  { icono: "🚗", descripcion: "Auto" },  
  { icono: "🤖", descripcion: "Robot" },  
  { icono: "😬", descripcion: "Fantasma" },  
  { icono: "👽", descripcion: "Alien" },  
  { icono: "🦷", descripcion: "Diente" },  
  { icono: "🎮", descripcion: "Joystick" }  
];
```

En la siguiente lámina están los pasos a realizar para este ejercicio.

{desafío}
latam_

Ejercicio

¡Manos al teclado!



Ejercicio

1. Crea una sección HTML con la clase `.iconos`.
2. Genera el arreglo de objetos con los iconos.
3. Crea una variable para contener la sección del HTML y selecciónala utilizando `document.querySelector`.
4. Recorre el arreglo de iconos y genera el template que desees para mostrar el ícono.
5. Agrega en cada iteración el template generado en la sección de iconos del HTML.

Ejercicio ¡Manos al teclado!



Ejercicio

Resultado de ejecución.



Iconos divertidos 🤪



Ejercicio

¡Manos al teclado!



/* Acceder a valores almacenados en un arreglo */ ✓

/* Mostrar los elementos de un arreglo por consola utilizando un ciclo */ ✓

/* Agregar elementos a una lista HTML a partir de un arreglo de Strings */ ✓

/* Crear un objeto con propiedades a partir de la interpretación de un texto */ ✓

/* Crear el template de una tarjeta que contenga propiedades de un objeto */ ✓

/* Agregar los elementos de una lista HTML a partir de un arreglo de objetos */ ✓

Objetivos



Cierre

{desafío}
latam_



¿Existe algún concepto que no
hayas comprendido?

Reflexionemos

- Revisar la guía que trabajarán de forma autónoma.
- Revisar en conjunto el desafío.

¿Qué sigue?



*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam