



Métodos de los arreglos



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





Inicio

{desafío}
latam_



/ Crear una lista de elementos donde se puedan agregar elementos a través de un input y borrar elementos de la lista a través de un botón */*

/ Agregar elementos a un arreglo */*

/ Agregar elementos a un arreglo a través de un input */*

/ Eliminar elementos de un arreglo */*

/ Eliminar elementos de un arreglo a través de un botón */*

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.





Activación de conceptos



¿Qué muestra el siguiente código?

```
a = [1,2,3,4,5]  
console.log(a[0])
```



Activación de conceptos



¿Qué muestra el siguiente código?

```
a = [1,2,3,4,5]  
console.log(a[6])
```




Activación de conceptos



¿Qué muestra el siguiente código?

```
a = [1,2,3,4,5]  
console.log(a[a.length-1])
```



Activación de conceptos



¿Qué debemos reemplazar en xxx para obtener 1?

```
a = [{id: 1}, {id: 2},  
      {id:3}]  
console.log(a[0].xxx)
```



Activación de conceptos



¿Cuál es el problema de modificar `.innerHTML`
dentro de un ciclo?



Activación de conceptos



Primer lugar:



Segundo lugar:



Tercer lugar:



Desarrollo

{desafío}
latam_



/* Agregar un elemento a un arreglo */

Agregar elementos a un arreglo

Agregar al final con push

```
>> a = [1,2,3,4,5,6]  
a.push("hola")  
console.log(a)
```

```
► Array(7) [ 1, 2, 3, 4, 5, 6, "hola" ]
```



Con push podemos agregar un elemento al final de un arreglo.

Agregar elementos a un arreglo

Agregar al principio con unshift

```
>> a = [1,2,3,4,5,6]  
a.unshift("hola")  
console.log(a)
```

```
► Array(7) [ "hola", 1, 2, 3, 4, 5, 6 ]
```



Con unshift podemos agregar un elemento al principio de un arreglo.

Ejercicio

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

```
const superHeroes= [  
  "Ironman",  
  "Superman",  
  "Hawkeye"  
]
```

1. Agrega el nombre de un superhéroe al final del arreglo usando el método *push*
2. Agrega el nombre de un superhéroe al principio del arreglo usando el método *unshift*

{desafío}
latam_

Ejercicio

¡Manos al teclado!



/* Crear una lista de elementos donde se puedan agregar elementos a través de un input y borrar elementos de la lista a través de un botón */

/* Agregar elementos a un arreglo */ ✓

/* Agregar elementos a un arreglo a través de un input */

/* Eliminar elementos de un arreglo */

/* Eliminar elementos de un arreglo a través de un botón */

Objetivos

/* Agregar elementos a través de un input */



Crearemos una página web **arreglos-con-input.html** donde agregaremos elementos a una lista a través de un input, pero esta vez lo haremos utilizando arreglos.

Autocompleteemos la base de html en nuestro archivo.

Agregar elementos a través de un input

Motivación

- Para agregar elementos tomábamos el valor de un input y simplemente lo agregábamos al final de un string.
- Esta nueva forma utilizando arreglos que aprenderemos ahora es un poco más compleja, pero nos permitirá luego borrar los valores que agreguemos o modificarlos.

Invitados:

- Isabel
- Pedro
- José
- Javiera

Agregar elementos a través de un input

Paso 1: El HTML

Para esto ocuparemos el siguiente código HTML:

```
<input id="nuevoInvitado"> <!-- 1 -->
<button id="agregarInvitado">Agregar</button>
<!-- 2 -->
<h3>Invitados: </h3>
<ul id="invitados"></ul> <!-- 3 -->
```

1

Agregar2

Invitados:

- Isabel
- Pedro
- José
- Javiera

3



Hasta aquí no hay diferencias de cómo hemos trabajado hasta ahora. La lista inicialmente no tendrá contenido.

Agregar elementos a través de un input

Paso 2: Obtener los elementos necesarios

```
<input id="nuevoInvitado">
<button id="agregarInvitado">Agregar</button>
<h3>Invitados: </h3>
<ul id="invitados"></ul>
<script>
  const listaDeInvitados = document.querySelector("#invitados")
  const invitadoInput = document.querySelector("#nuevoInvitado")
  const btnAgregar = document.querySelector("#agregarInvitado")
</script>
```

Agregar elementos a través de un input

Paso 3: Agregamos el elemento al arreglo



Creamos el arreglo de invitados y agregamos la interacción al botón para que al ser presionado, tome el valor del input y lo agregue como un nuevo elemento al arreglo

```
<input id="nuevoInvitado">
<button id="agregarInvitado">Agregar</button>
<h3>Invitados: </h3>
<ul id="invitados"></ul>
<script>
  const listaDeInvitados = document.querySelector("#invitados")
  const invitadoInput = document.querySelector("#nuevoInvitado")
  const btnAgregar = document.querySelector("#agregarInvitado")
  const invitados = []

  btnAgregar.addEventListener("click", () => {
    const nuevoInvitado = invitadoInput.value
    invitados.push(nuevoInvitado)
    invitadoInput.value = ""
  })
</script>
```

Agregar elementos a través de un input

Reseteamos el input después de agregar el valor



Asignamos un string vacío al value del input para vaciarlo.

```
btnAgregar.addEventListener("click", () => {  
  const nuevoInvitado = invitadoInput.value  
  invitados.push(nuevoInvitado)  
  invitadoInput.value = ""  
})  
</script>
```


Agregar elementos a través de un input

Paso 4: Iteramos sobre el arreglo y actualizamos el HTML



El paso final es
cada vez que se
agrega un elemento

```
const listaDeInvitados = document.querySelector("#invitados")
const invitadoInput = document.querySelector("#nuevoInvitado")
const btnAgregar = document.querySelector("#agregarInvitado")
const invitados = []

btnAgregar.addEventListener("click", () => {
  /* Agregamos el invitado al arreglo */
  const nombreInvitado = invitadoInput.value
  invitados.push(nombreInvitado)
  invitadoInput.value = ""

  /* Actualizamos la información en el HTML */
  let html = ""
  for (let invitado of invitados) {
    html += `<li>${invitado}</li>`;
  }
  listaDeInvitados.innerHTML = html;
})
```

Métodos de los arreglos

Agregar elementos

Y ahora nuestra lista ahora puede agregar nuevos invitados:

Invitados

- Sam
- Spencer
- Nataly
- Juanes

Ejercicio

- Practicaremos realizando los mismos pasos anteriores desde el comienzo, pero esta vez para construir una lista de tareas.



Ejercicio ¡Manos al teclado!



/* Crear una lista de elementos donde se puedan agregar elementos a través de un input y borrar elementos de la lista a través de un botón */

/* Agregar elementos a un arreglo */ ✓

/* Agregar elementos a un arreglo a través de un input */ ✓

/* Eliminar elementos de un arreglo */

/* Eliminar elementos de un arreglo a través de un botón */

Objetivos

/* Eliminando elementos de un arreglo */

Eliminar elementos de un arreglo

El método splice

- En JavaScript podemos eliminar elementos de un arreglo utilizando splice.

```
arreglo.splice(<índice>, <cantidad de elementos a eliminar>)
```

- Usualmente eliminaremos un solo elemento en un índice particular.

```
const cocteles = [ "Piñacolada", "Margarita", "Mojito", "Caipirinha" ]  
cocteles.splice(2, 1) /* Eliminamos el elemento en el índice 2, o sea Mojito */  
console.log(cocteles)
```



Eliminar un elemento de un arreglo

El método findIndex

En algunos casos vamos a necesitar borrar un elemento a partir de su valor, por ejemplo queremos borrar "Margarita" del arreglo anterior, en ese caso tendremos que buscar el índice primero con findIndex y luego borrar el elemento ocupando splice.

```
const cocteles = [ "Piñacolada", "Margarita", "Mojito", "Caipirinha" ]  
const index = cocteles.findIndex( (elemento) => elemento === "Margarita" )  
cocteles.splice(index, 1)  
console.log(cocteles)
```



Ejercicio

Realicemos el siguiente ejercicio en donde debemos eliminar la talla 44 del siguiente arreglo:

```
const tallasDisponibles = [ 41, 42, 43, 44, 45 ]
```

1. Utiliza el splice para borrar el elemento en el índice 3
2. Utiliza findindex y splice para borrar el elemento con el valor 43

Ejercicio

¡Manos al teclado!



/* Crear una lista de elementos donde se puedan agregar elementos a través de un input y borrar elementos de la lista a través de un botón */

/* Agregar elementos a un arreglo */ ✓

/* Agregar elementos a un arreglo a través de un input */ ✓

/* Eliminar elementos de un arreglo */ ✓

/* Eliminar elementos de un arreglo a través de un botón */

Objetivos

Trabajando con objetos

En lugar de solo guardar strings guardaremos objetos con un identificador

```
const invitados = [  
  { id: 431, nombre: "Jhonny Depp" },  
  { id: 124, nombre: "Brad Pitt" },  
  { id: 541, nombre: "Leonardo DiCaprio" },  
  { id: 664, nombre: "Morgan Freeman" }  
];
```



El identificador tiene que ser único, usualmente este valor lo entrega el backend cuando insertamos el valor por ahora generaremos estos valores localmente.

Trabajando con objetos

Modificamos nuestro código inicial

Insertamos un objeto en lugar de un string, el id lo generaremos automáticamente, podríamos ocupar un contador pero para este caso es más sencillo obtener el tiempo en segundos desde 1970, de esta forma es un identificador único para el contexto del ejercicio.

```
btnAgregar.addEventListener("click", () => {  
  /* Agregamos el invitado al arreglo */  
  nuevoInvitado = {id: Date.now(), nombre: invitadoInput.value}  
  invitados.push(nuevoInvitado)  
  invitadoInput.value = ""  
  
  /* Actualizamos la información en el HTML */  
  let html = ""  
  for (invitado of invitados) {  
    html += `<li>${invitado.nombre}</li>`  
  }  
  listaDeInvitados.innerHTML = html;  
})
```

{desafío}
latam_



Tenemos que mostrar el nombre del invitado, no el objeto.

Trabajando con objetos

Implementando el borrado

```
btnAgregar.addEventListener("click", () => {  
  /* Agregamos el invitado al arreglo */  
  nuevoInvitado = {id: Date.now(), nombre: invitadoInput.value}  
  invitados.push(nuevoInvitado)  
  invitadoInput.value = ""  
  
  /* Actualizamos la información en el HTML */  
  let html = ""  
  for (invitado of invitados) {  
    html += `- ${invitado.nombre} <button onclick="borrar(${invitado.id})"> x </button> </li>`;  
  }  
  listaDeInvitados.innerHTML = html;  
})

```



Trabajando con objetos

Implementando el borrado

```
function borrar(id){  
  const index = invitados.findIndex((ele) => ele.id == id)  
  invitados.splice(index, 1)  
  
  /* Actualizamos la información en el HTML */  
  let html = ""  
  for (invitado of invitados) {  
    html += `- ${invitado.nombre} <button onclick="borrar(${invitado.id})"> x </button> </li>`;  
  }  
  listaDeInvitados.innerHTML = html;  
}

```



Creamos la función borrar, donde buscamos el elemento por su id y luego lo borramos ocupando splice, finalmente actualizamos la página.

Reutilizando código

Creando la función render



Como será frecuente que tengamos que actualizar la lista de invitados, por ejemplo al agregar un invitado nuevo, al borrar uno, al modificar uno, al agregar los valores iniciales es mejor crear una función y llamar cuando se necesiten actualizar.

```
/* Actualizamos la información en el HTML */
function renderInvitados(){
  let html = ""
  for (invitado of invitados) {
    html += `- ${invitado.nombre} <button
onclick="borrar(${invitado.id})"> x </button> </li>`;
  }
  listaDeInvitados.innerHTML = html;
}

```

```
function borrar(id){
  const index = invitados.findIndex((ele) => ele.id == id)
  invitados.splice(index, 1)
  renderInvitados()
}
```

Ejercicio

Realicemos el siguiente ejercicio en donde debamos ofrecer una lista de servicios básicos pendientes por pagar que puedan eliminarse por parte del usuario. Solo el borrado, no hay que agregarlos

Para esto ocupa el siguiente HTML:

```
<h4>Servicios pendiente por pagar:</h4>  
<ul class="servicios"></ul>
```

Y el siguiente arreglo:

```
const servicios = [  
  { id: 1, nombre: "Agua" },  
  { id: 2, nombre: "Electricidad" },  
  { id: 3, nombre: "Gas" },  
  { id: 4, nombre: "Internet" },  
  { id: 5, nombre: "Cable Tv" },  
];
```

Ejercicio ¡Manos al teclado!



/* Crear una lista de elementos donde se puedan agregar elementos a través de un input y borrar elementos de la lista a través de un botón */ ✓

/* Agregar elementos a un arreglo */ ✓

/* Agregar elementos a un arreglo a través de un input */ ✓

/* Eliminar elementos de un arreglo */ ✓

/* Eliminar elementos de un arreglo a través de un botón */ ✓

Objetivos



**¡¡ Las actividades de la guía adelantan
gran parte del desafío !!**



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