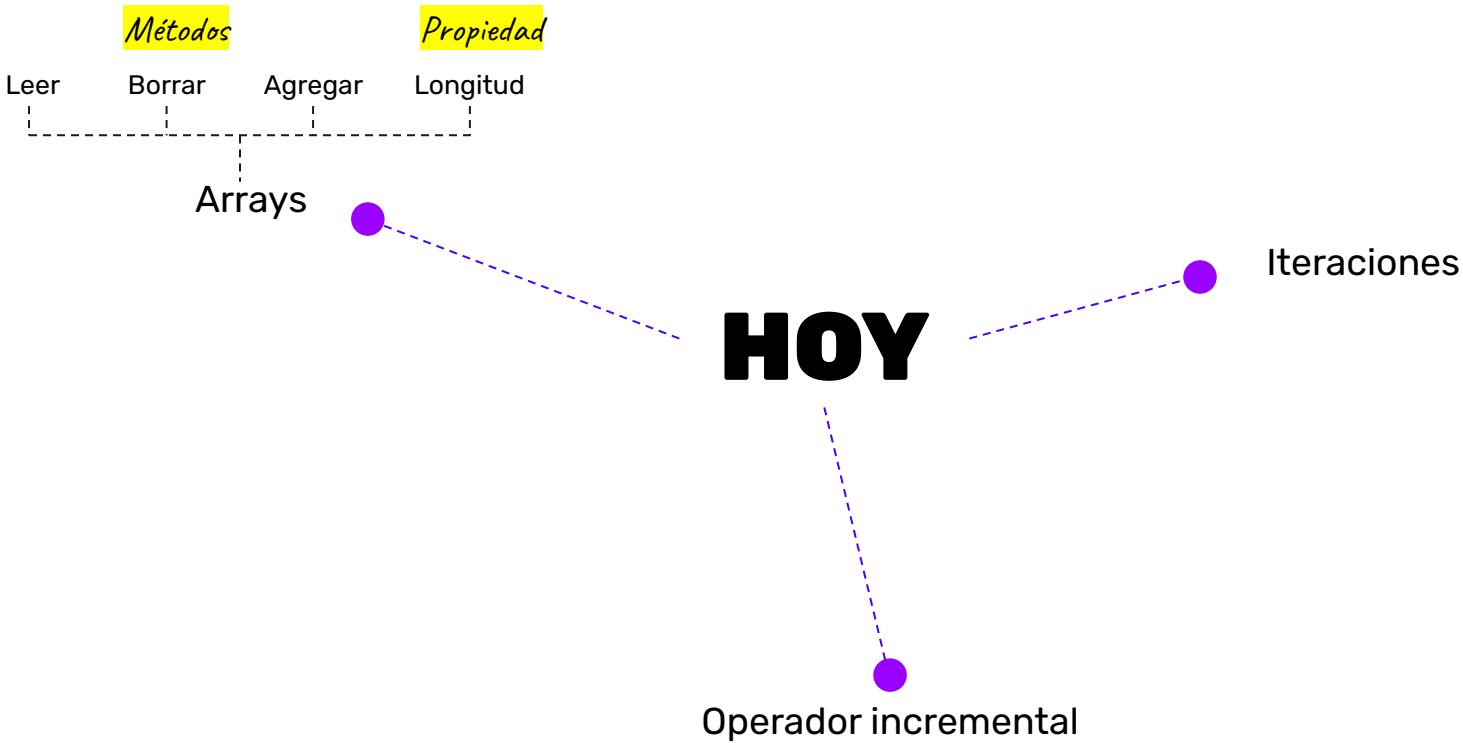
ACÁMICA


Métodos
Propiedad
Leer
Borrar
Agregar
Longitud
Arrays
HOY
Iteraciones
Operador incremental

Agenda

Daily

Operaciones con arrays

Ciclos

Break

Actividades

Cierre

Daily

Daily

Sincronizando... 20min.

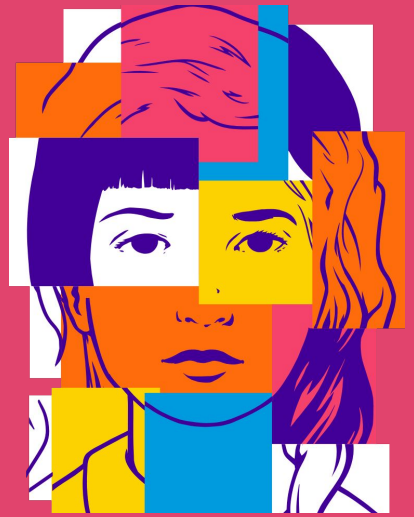
Bitácora

¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Challenge

¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Iteraciones



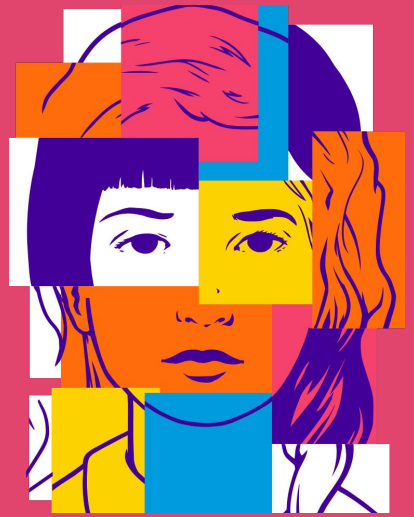
Iteraciones

Con las iteraciones podemos repetir una porción de código una N cantidad de veces.

Hay varias formas de iterar en Javascript, una de ellas es a través del ciclo “for”.

```
// ¿Qué muestra esto por consola?
for(let iteracion=0; iteracion <= 10; iteracion++) {
  console.log(iteracion);
}
```

Operador incremental



Operador incremental

El operador incremental (++) suma una unidad al operando numérico.

La sintaxis consiste en agregar dos signos de adición (+) antes de la variable.

```
x++;
```

Esto hará que la variable se incremente en un valor. Este operador solo funciona con números.

```
let x = 0;
x = x + 1;
x = x + 1;
console.log(x); // ¿Cuánto vale x?
```

¡Programamos!

Misma dinámica

Justo como lo hicimos en otros encuentros, estaremos construyendo un ejercicio mientras practicamos el concepto de iteraciones.

Instrucciones para el **squad lead**:

- Compartir esta presentación como PDF por el canal de slack. Todxs lxs trainees deberán ir siguiendo la presentación, paso a paso.
- Abrir un Code Sandbox nuevo con el template “Vanilla JS”, ir al archivo index.js y borrar todo el contenido. Dejar el archivo en blanco.
- Escoger una persona voluntaria al azar, para que ejecute el paso 1, listado en las próximas láminas.

Paso 1, Escribir el ciclo

La sintaxis del ciclo for puede ser un poco enredada. No hay problema si en algún momento queremos revisar nuestros apuntes o buscar en Google.

Para este paso, escribe el ciclo que te permita mostrar los números del 1 al 50 por la consola.

```
for(let numero=1; numero <= 50; numero++) {
  console.log(numero);
}
```

Ahora, ¡escoge a otra persona para que haga el paso 2!

Paso 2, Iteraciones + funciones

Vamos a sacarle el provecho a Javascript, combinando dos de sus conceptos.
Crearemos una función que se encargue de mostrar por consola la secuencia entre dos números, por ejemplo, la invocación:

```
mostrarNumeros(4,30);
```

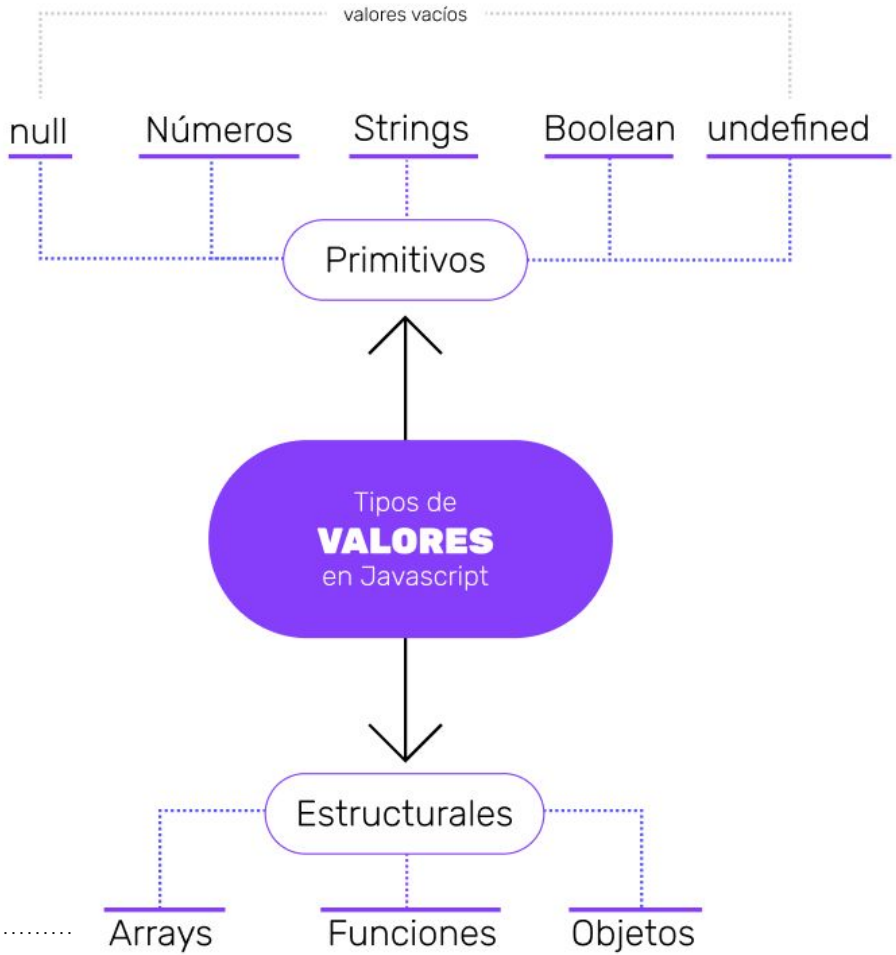
Deberá mostrar por consola toda la secuencia numérica desde el 4, hasta el número 30.

Si te sientes con ganas de hacer el reto sin código, ¡adelante!, pero si te bloqueas, te dejamos la respuesta acá.

Arrays como estructura de datos

Estos son los tipos de valores en Javascript.

El array forma parte de la categoría “estructural”.



```
"un string"
8
false
```

El array es como un pastillero.

Un contenedor lineal que tiene subdivisiones ordenadas, lineales y consecutivas, que nos permiten meter, en vez de pastillas, valores de Javascript.

El array es un valor estructural, o estructura de datos.

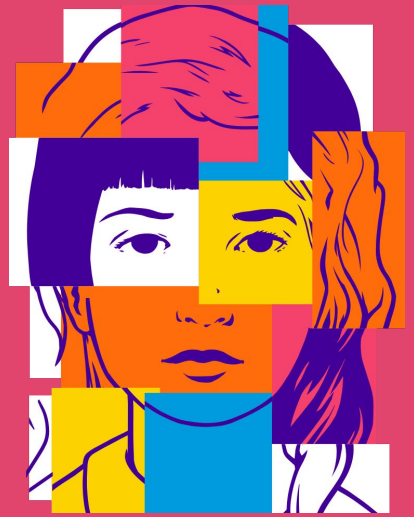
Los datos primitivos nos permiten guardar un tipo de valor por variable.

```
let decibelioP1 = 24;
let decibelioP2 = 30;
let decibelioP3 = 15;
```

Mientras que un array, nos permite guardar varios tipos de valores en una variable.

```
let decibelios = [24,30,15];
```

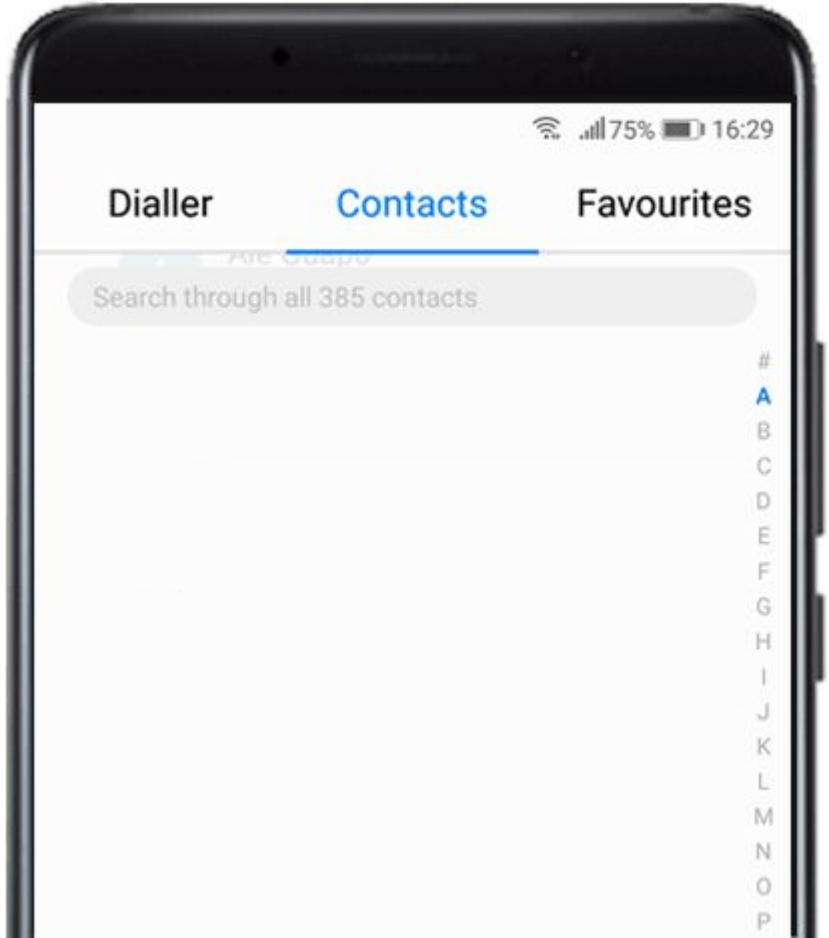
Arrays en UI’s

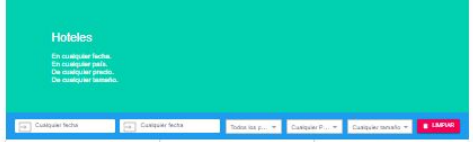


¿Donde encontramos los arrays y las iteraciones en el desarrollo web?

```
let contactos = [
  "Aleix",
  "Alejandro Moreno",
  "Alex Aller",
  "Alex",
  "Alex Nauth-Misir",
  "..."
];


for(i=0; i<=contactos.length; i++) {
  // mostrar los contactos
};
```





Proyecto de reserva de hoteles

El array será una estructura de datos clave para el proyecto que estaremos construyendo en este segundo sprint.

¡BREAK!

Ph. Credit: Drew Coffmann

Métodos de arrays

Métodos de arrays

Javascript nos da una serie de métodos para operar con arrays. Estos, hacen distintas cosas por nosotros.

Los métodos que vamos a conocer hoy, nos ayudan a lo siguiente:

- Cómo agregar elementos a un array definido
- Como borrar un elemento
- Como saber si un elemento está en el array

Vamos a utilizar el siguiente array para nuestros ejemplos:

```
let frutas = ["manzana", "pera", "uva", "banana"];
```

Buscar elementos

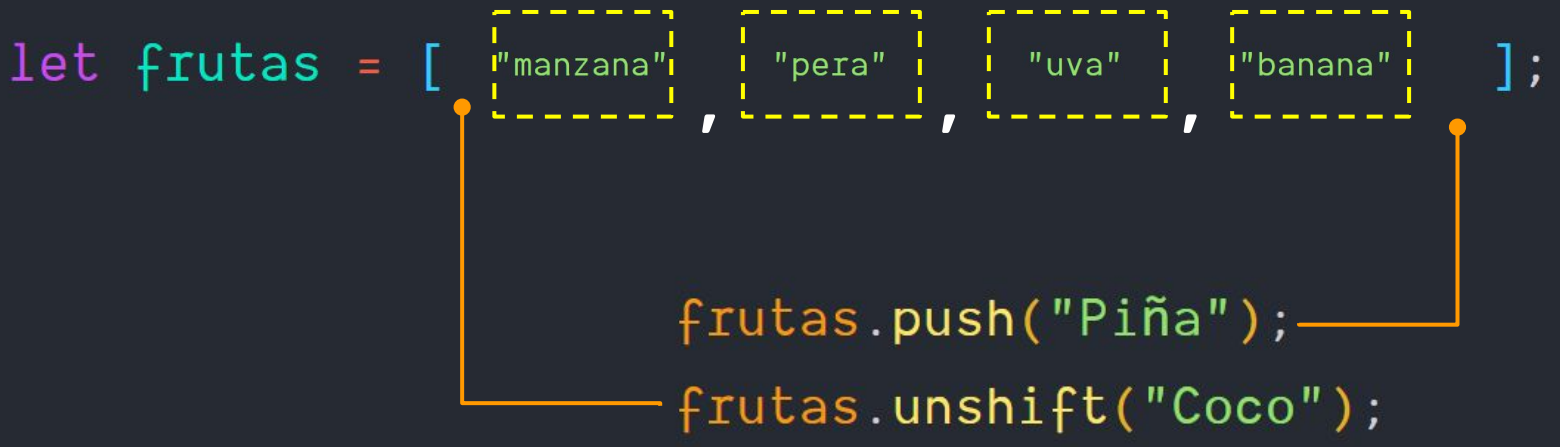
El método indexOf recorre el array hasta encontrar la primera coincidencia del valor que se especifique. De encontrarlo, retorna el index, o posición. Si no encuentra el elemento, el método retorna -1

```
let frutas = ["manzana", "pera", "uva", "banana"];

frutas.indexOf("uva");
```

Agregar elementos

El método unshift agrega elementos al principio del array, mientras que el push, los agrega al final.



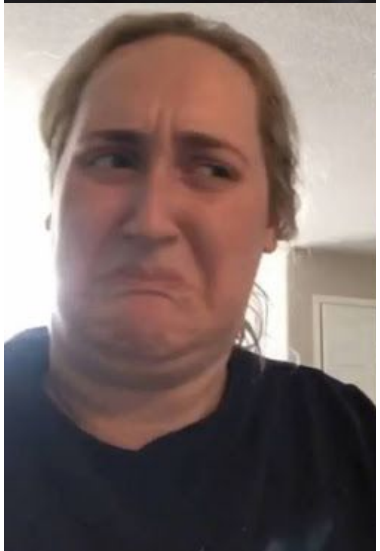
Borrar elementos

El método splice espera que le pasemos el index del elemento a borrar, así como la cantidad de elementos que se deseen borrar a partir de ese index.

```
let frutas = ["manzana", "pera", "uva", "banana"];
let remover = frutas.indexOf("uva"); // 2
frutas.splice(remover,1);
// ["manzana","pera","banana"]
```

¡Hora de hacer un smoothie!

...con conceptos de Javascript.



Arrays

Ciclos

Funciones

```
let frutas = ["manzana","pera","uva","banana"];
// quisiera mostrar las frutas por consola...
```

```
let frutas = ["manzana","pera","uva","banana"];
// quisiera mostrar las frutas por consola...
for(let i = 0; i <= frutas.length; i++) {
  console.log(frutas[i]);
};
let personas = ["Joan", "Carla", "Miranda", "Nico"];
// quisiera mostrar las personas por consola...
for(let i = 0; i <= personas.length; i++) {
  console.log(personas[i]);
};
let notas = [10, 5, 7, 9, 5, 8.8, 9, 10, 10, 10];
// quisiera mostrar las notas por consola...
for(let i = 0; i <= notas.length; i++) {
  console.log(notas[i]);
};
let resumenTDC = [22563.40, 1145, 109, 14223.12,12500];
// quisiera mostrar las compras por consola...
for(let i = 0; i <= resumenTDC.length; i++) {
  console.log(resumenTDC[i]);
};
```

```
// quisiera mostrar          por consola...
// quisiera mostrar            por consola...
// quisiera mostrar    algo      consola...
// quisiera mostrar           por consola...
```

```
for(let i = 0; i <=  lista .length; i++) {
  console.log(lista [i]);
};
```

```
let frutas = ["manzana","pera","uva","banana"];
let personas = ["Joan", "Carla", "Miranda", "Nico"];
let notas = [10, 5, 7, 9, 5, 8.8, 9, 10, 10, 10];
let resumenTDC = [22563.40, 1145, 109, 14223.12,12500];

function mostrarPorConsola(lista) {
  for(let i = 0; i <= lista.length; i++) {
    console.log(lista[i]);
  };
}

mostrarPorConsola(frutas);
mostrarPorConsola(personas);
mostrarPorConsola(notas);
mostrarPorConsola(resumenTDC);
```

```
let frutas = ["manzana","pera","uva","banana"];
// quisiera mostrar las frutas por consola...
for(let i = 0; i <= frutas.length; i++) {
  console.log(frutas[i]);
};
let personas = ["Joan", "Carla", "Miranda", "Nico"];
// quisiera mostrar las personas por consola...
for(let i = 0; i <= personas.length; i++) {
  console.log(personas[i]);
};
let notas = [10, 5, 7, 9, 5, 8.8, 9, 10, 10, 10];
// quisiera mostrar las notas por consola...
for(let i = 0; i <= notas.length; i++) {
  console.log(notas[i]);
};
let resumenTDC = [22563.40, 1145, 109, 14223.12,12500];
// quisiera mostrar las compras por consola...
for(let i = 0; i <= resumenTDC.length; i++) {
  console.log(resumenTDC[i]);
};
```

```
let frutas = ["manzana","pera","uva","banana"];
let personas = ["Joan", "Carla", "Miranda", "Nico"];
let notas = [10, 5, 7, 9, 5, 8.8, 9, 10, 10, 10];
let resumenTDC = [22563.40, 1145, 109, 14223.12,12500];

function mostrarPorConsola(lista) {
  for(let i = 0; i <= frutas.lista; i++) {
    console.log(lista[i]);
  };
}

mostrarPorConsola(frutas);
mostrarPorConsola(personas);
mostrarPorConsola(notas);
mostrarPorConsola(resumenTDC);
```


Mon
Tue-Thur
Fri -Sat
Sunday

Programamos

trainees

Sin rueditas de apoyo

Es hora de quitarle las rueditas de apoyo a la bici y empezar a tener autonomía sobre cómo resolver ejercicios de programación.

En las siguientes láminas, deberán acordar como grupo, una solución para cada ejercicio.

Sugerimos que una persona comparta su pantalla, por cada ejercicio. No deben repetir personas.

¿Que pasa si me toca compartir algo y no tengo idea de como resolver el ejercicio? Apoyate en el squad lead y tus compañerxs.

¿Que pasa si me toca compartir algo y la tengo clara? Queremos que saques el squad lead que hay en ti para explicar el ejercicio mientras lo resuelves.

Han pasado __ años, *funcional*

Crea una función que muestre por consola los números del 0 a una edad.

```
function mostrarNumeros(edad) {
  // el ciclo aquí
};

mostrarNumeros(30);
// muestra por consola los números del 0 al 30
mostrarNumeros(4);
// muestra por consola los números del 0 al 4
```

Consulta aquí un punto de partida

Follow me

Agrega elemento a un array utilizando los métodos unshift y push.

Tomando en cuenta un array de twitter handles, los cuales pertenecen a grandes referentes en el mundo del front-end, utiliza el método unshift, o push, para agregar personas a la lista.

Punto de partida

Follow me, *funcional*

Partiendo del ejercicio anterior, desarrolla una función que tome un username, y un array de users, y que retorne el array con el username dentro de sus elementos.

Considera este gist para empezar

Stalker

Desarrolla una función que verifique si un string está dentro de un array.

La función debe tomar una lista y un string. La lista representa un array de usernames y el string el nombre a buscar.

Se espera que la función muestre por consola el mensaje correspondiente al resultado.

Punto de partida

Unfollow, *funcional*

Agrega al ejemplo anterior la funcionalidad de poder eliminar elementos de un array con el método splice.

Desarrolla una función que tome un username, y un array de users, y que retorne el array sin el username dentro de sus elementos.

Adicionalmente, muestra un mensaje por consola que indique si el username pasado como argumento no fue encontrado en la lista.

Tip: utiliza la función de stalker para determinar si un username está en la lista de nombres.

Promedio

Desarrolla una función que retorne el promedio de una lista de números. La función deberá recibir como argumentos un array de números.

Para saber como calcular un promedio combinando conceptos de arrays e iteraciones, visita este gist, o puedes buscar en línea alguna solución.

Te ofrecemos este punto de partida.

Escándalo-tron v0.4

El **Escándalo-tron** ahora puede recibir una N cantidad de decibelios, haciéndolo útil para determinar si un gran grupo de personas está produciendo mucho ruido.

Extiende la función **`escandalotron()`** (realizada en el encuentro pasado) para que tome un array de números (cantidad de decibelios).

La lógica de decisión sobre el resultado deberá ser igual, pero ahora deberás de considerar el promedio de decibelios para establecer la decisión.

Escándalo-tron v0.4

Si el **promedio** de decibelios es menor o igual a 30, el **Escándalo-tron** arrojará un mensaje indicando que las personas están susurrando, si es mayor o igual a 60, indicará que el sonido es normal, y si es mayor a 90, indicará que las personas están haciendo un ESCÁNDALO.

La invocación de la función espera un array de números, por ejemplo:

```
escandalotron([10,5,20,30,10,5]);
```

Se espera un mensaje por consola indicando el resultado.

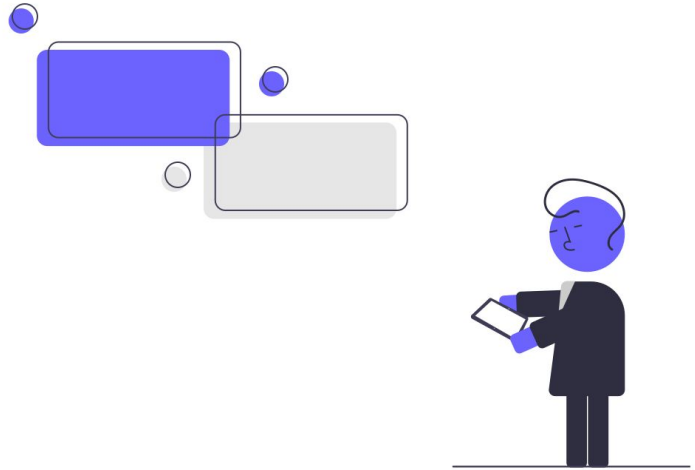
Tip: utiliza la función de promedio, construida en el ejercicio anterior para obtener el promedio de los decibelios.

Soluciones

Soluciones

A continuación te damos una solución a los ejercicios planteados, asegurate de revisarlos una vez que completes los tuyos, para poder comparar.

Hace __ años
Follow me
Stalker
Unfollow
Promedio
Escándalotron

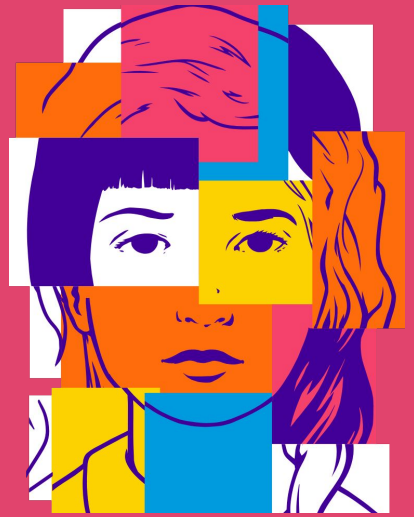


Para la próxima

1) Termina los ejercicios del encuentro de hoy.
2) Lee la toolbox 22
3) Resuelve el challenge.

ACÁMICA

Recursos extra



Sintaxis

¿como construyo un array?

El valor de un array se declara entre corchetes, escribiendo los valores separados entre comas.

Cada valor ocupa una posición del array, o index

```
let numeros = [24,30,15];
```

0 1 2

A su vez, cada index tiene un valor que indica su ubicación dentro del array.

Longitud

¿cuanto mide un array?

La longitud de un array es igual a la cantidad de valores que tenga adentro.

Este array tiene tres valores. Por lo tanto su longitud es de 3.

```
let numeros = [24,30,15];
```

La propiedad .length nos indica la longitud de un array.

```
console.log(numeros.length) // 3
```

Sintaxis

¿qué valores puedo meter en un array?

```
let listaNumeros = [0,2,5,6,7,3,12,-65,3.5];
let listaPalabras = ["hola", "pasta", "margarita"];
let listaBooleanos = [true,false,false,true];
let listaMix = [0,"hola", true, 344, "computadora"];
```

En un array, pueden coexistir todos los valores permitidos en Javascript.

Lectura de valores

¿como puedo leer un valor de un array?

```
let decibelios = [24,30,15];

console.log(decibelios[1]); // ¿que muestra esto por consola?
```

Recap

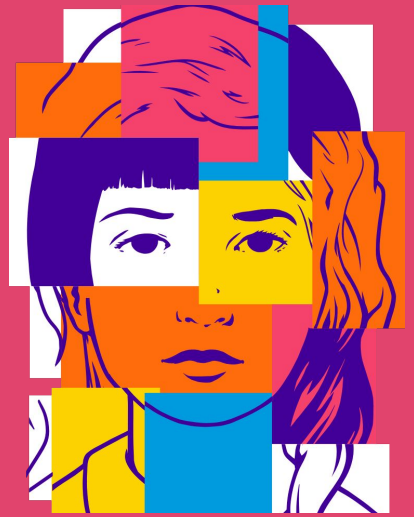
Los arrays son valores: estructurales

La sintaxis del array es: `let numeros = [24,30,15];`

Para calcular la longitud del array usamos: .length

Para leer el valor de un array podemos escribir: `console.log(decibelios[1]);`

Iteraciones



Iteraciones

Vimos que podíamos leer valores de un array de la siguiente forma:

```
let decibelios = [24,30,15];

console.log(decibelios[1]);
console.log(decibelios[0]);
console.log(decibelios[2]);
```

¿Qué pasaría si quisiéramos mostrar valores individuales de un array más largo?

```
let decibelios = [23,34,43,32,12,35,32,23,21];

console.log(decibelios[0]);
console.log(decibelios[1]);
console.log(decibelios[2]);
console.log(decibelios[3]);
console.log(decibelios[4]);
console.log(decibelios[5]);
console.log(decibelios[6]);
console.log(decibelios[7]);
console.log(decibelios[8]);
```

Ciclo `for`

Con los ciclos podemos repetir
tareas según una condición lógica.

```
for(let iteracion = 0; iteracion <= 10; iteracion++) {
  console.log(iteracion);
};
// ¿Qué mostrará esto por consola?
```

Sintaxis

palabra reservada

desde donde

hasta donde

en cuanto

```
for(let iteracion = 0; iteracion <= 10; iteracion++) {
  console.log(iteracion);
};
```

Patrón muy común para escribir un ciclo for

```
for(let i = 0; i <= 10; i++) {
  console.log(i);
};
```

Ciclos + Arrays

Vamos a crear un ciclo que nos muestre por consola cada string de este array.

```
let tareas = [
  "Terminar de leer la bitácora 21",
  "Hacer el challenge",
  "Empezar a ver The OA en Netflix",
  "Aprender sobre arrays"
];
```

¿desde donde? ¿hasta donde? ¿de cuanto en cuanto?

```
for (                ;                ;          ) {

};
```

¿que vamos a mostrar?

Ciclo `for`

Encuentra ejemplos haciendo click aquí

```
for(let i = 1; i < 10; i++) {
  console.log(i);
};
// ¿Qué mostrará esto por consola?
```

```
for(let i = 2; i <= 6; i++) {
  console.log(i);
};
// ¿Qué mostrará esto por consola?
```

```
for(let i = 1; i <= 9; i++) {
  console.log(i);
};
// ¿Qué mostrará esto por consola?
```