

# CURSO DESARROLLO DE APLICACIONES

## FULL STACK JAVASCRIPT TRAINEE – Módulo 3 Lección 5

# Ejercicio Práctico 1



Crea un programa en el que se solicite al usuario ingresar un determinado número de palabras (a su elección). Con esto, debes contar la cantidad de vocales presentes en el total de palabras ingresadas.

# Ejercicio Práctico 1



Como ya hemos visto, crea en tu editor un nuevo archivo HTML con los elementos básicos de su estructura. Crear en tu editor el archivo app.js y luego agrega a `<script>` el atributo `src` con el nombre del archivo.

# Ejercicio Práctico 1



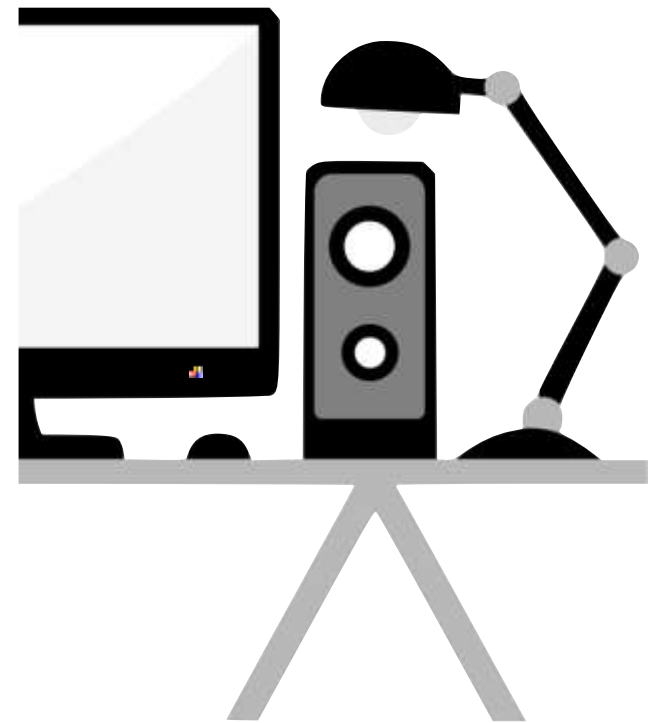
Solicita datos de entrada al usuario consultándole primero cuántas palabras desea ingresar y luego de cuáles se tratarían. A continuación, almacena cada palabra en una lista.

# Ejercicio Práctico 1

```
var cantidad= parseInt(prompt("Ingrese cantidad de palabras: " ));

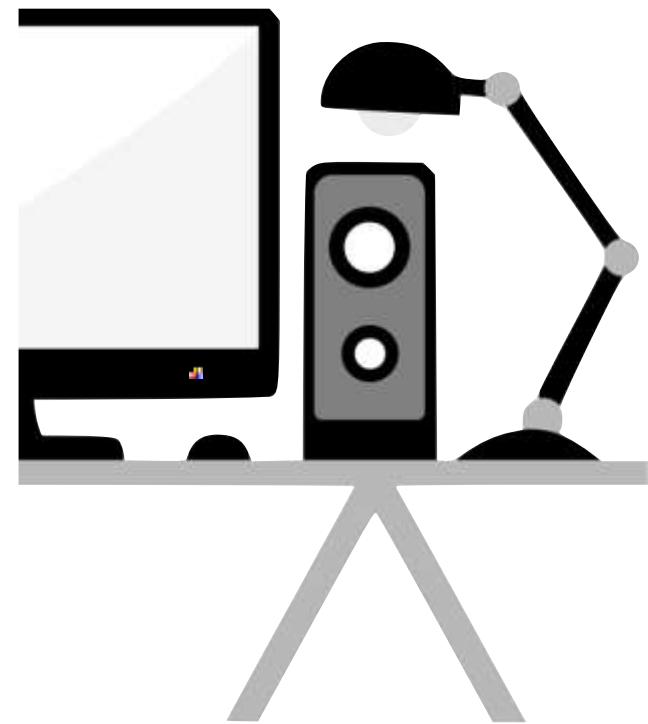
let list=[]
for(let i=0;i<cantidad;i++){
    contador=i+1;
    var palabra= prompt("Ingrese palabra número "+ contador+": " );
    list.push(palabra);
}
```

# Ejercicio Práctico 1



Si almacenas una función al interior de una variable, se trata entonces de una función de tipo expresiva. En este caso no puedes situarla al final de tu programa, porque de ser así no será leída al momento de invocarse en líneas anteriores.

# Ejercicio Práctico 1



Declararemos de forma abreviada una función que permita realizar el conteo de vocales en una palabra.

# Ejercicio Práctico 1

```
const contarVocales = palabra => {  
  const vocales = "aáeéííoóuú";  
  let cantidadVocales = 0;  
  for (const letra of palabra) {  
    if (vocales.includes(letra.toLowerCase())) {  
      cantidadVocales++;  
    }  
  }  
  return cantidadVocales;  
};
```

“Se declara la función contarVocales, con un argumento (palabra) el cual retornará (=>) cantidadVocales”  
.



# Ejercicio Práctico 1

```
const contarVocales = palabra => {  
  const vocales = "aáeéííoóuú";  
  let cantidadVocales = 0;  
  for (const letra of palabra) {  
    if (vocales.includes(letra.toLowerCase())) {  
      cantidadVocales++;  
    }  
  }  
  return cantidadVocales;  
};
```

Otra forma abreviada del bucle for es definiendo un elemento iterador que a la vez es parte de lo que se recorrerá. Por ejemplo, en el caso de una cadena, se recorrerá cada carácter.

# Ejercicio Práctico 1

```
const contarVocales = palabra => {  
  const vocales = "aáeeííoóuú";  
  let cantidadVocales = 0;  
  for (const letra of palabra) {  
    if (vocales.includes(letra.toLowerCase())) {  
      cantidadVocales++;  
    }  
  }  
  return cantidadVocales;  
};
```

El método  
toLowerCase()  
devuelve el valor en  
minúsculas de la  
cadena que realiza la  
llamada.

# Ejercicio Práctico 1

```
const contarVocales = palabra => {  
  const vocales = "aáeéííoóuú";  
  let cantidadVocales = 0;  
  for (const letra of palabra) {  
    if (vocales.includes(letra.toLowerCase())) {  
      cantidadVocales++;  
    }  
  }  
  return cantidadVocales;  
};
```

El método `array.includes()` determina si un o varios valores de entradas se encuentra dentro de otro array. En este sentido, al evaluarse cada caracter de la palabra ingresada, si está dentro del string vocales (que es una array de caracteres), se contabiliza una vocal adicional por parte de la variable `cantidadVocales`.

# Ejercicio Práctico 1



Puede que se hayan ingresado muchas palabras en la lista. En este caso cuando se quiera recorrer cada elemento de un array, se tomará el script completo. Por esta razón, seguiremos con la lógica de que deseamos recorrer cada carácter de una cadena. Por lo que se hace necesario convertir la lista a un string.

# Ejercicio Práctico 1

```
const palabraNueva=list.join("");  
  
const vocales = contarVocales(palabraNueva);  
  
console.log("Todas las palabras ingresadas en total tienen %d vocales", vocales);
```

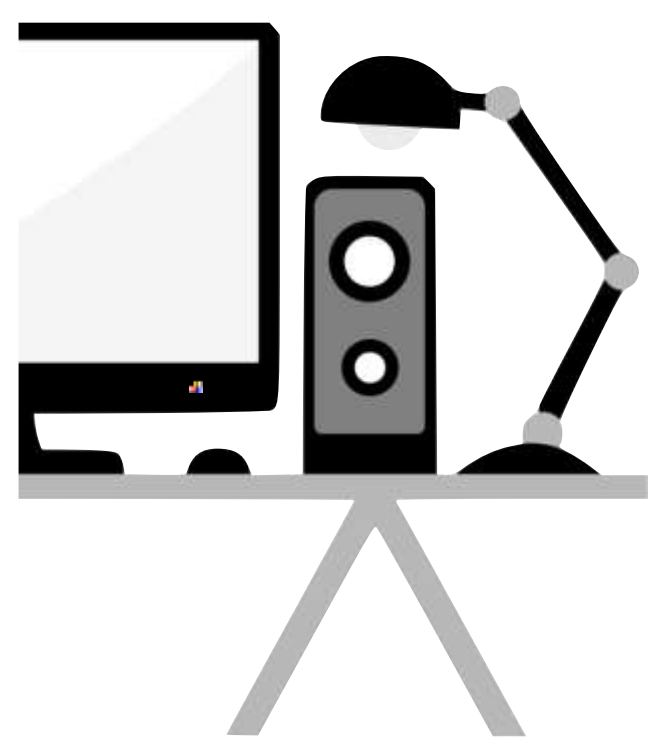
El método join() crea una nueva cadena concatenando todos los elementos de un array , separados por comas, o un separador específico señalado como valor de entrada. Como se emplea list.join(""), se acoplará una palabra con otra sin separador.

# Ejercicio Práctico 1

```
const palabraNueva=list.join("");  
  
const vocales = contarVocales(palabraNueva);  
  
console.log("Todas las palabras ingresadas en total tienen %d vocales", vocales);
```

Para finalizar, se invoca la función dentro de una variable que será sustituida por %d y construida en la ejecución del programa por la consola. La 'impresión formateada' o printf emplea formatos específicos, comenzando por %, indicando la ubicación y el método para traducir un dato. %d se emplea para números. Esto sólo puede ser utilizado por console.log() y console.debug().

# Ejercicio Práctico 1



Puedes revisar la documentación de los métodos empleados visitando el siguiente link:

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array/includes](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/includes)