

Test de JavaScript.

Test de JavaScript

Departamento de Ciencias de la Computación.

Lcdo. Diego Medardo Saavedra García. Mgtr.¹

1. Universidad de las Fuerzas Armadas ESPE., Departamento de Ciencias de la Computación.



Diego Saavedra

Test JavaScript <https://github.com/5t4t1ck>

Table of contents

1	Test de JavaScript.	1
1.1	Instrucciones para tomar esta prueba	1
1.2	Variables y operaciones	2
1.2.1	Responde las siguientes preguntas en la sección de comentarios: . . .	2

1.2.2	Determina el nombre y tipo de dato para almacenar en variables la siguiente información:	2
1.2.3	Traduce a código JavaScript las variables del ejemplo anterior y deja tu código en los comentarios.	2
1.2.4	Calcula e imprime las siguientes variables a partir de las variables del ejemplo anterior:	2
1.3	Funciones	2
1.3.1	Responde las siguientes preguntas en la sección de comentarios: . .	2
1.3.2	Convierte el siguiente código en una función, pero, cambiando cuando sea necesario las variables constantes por parámetros y argumentos en una función:	3
1.4	Condicionales	3
1.4.1	Responde las siguientes preguntas en la sección de comentarios: . .	3
1.4.2	Replica el comportamiento del siguiente código que usa la sentencia switch utilizando if, else y else if:	3
1.4.3	Replica el comportamiento de tu condicional anterior con if, else y else if, pero ahora solo con if (sin else ni else if).	4
1.5	Ciclos	4
1.5.1	Responde las siguientes preguntas en la sección de comentarios: . .	4
1.5.2	Replica el comportamiento de los siguientes ciclos for utilizando ciclos while:	4
1.5.3	Escribe un código en JavaScript que le pregunte a los usuarios cuánto es 2 + 2. Si responden bien, mostramos un mensaje de felicitaciones, pero si responden mal, volvemos a empezar.	4
1.6	Listas	5
1.6.1	Responde las siguientes preguntas en la sección de comentarios: . .	5
1.6.2	Crea una función que pueda recibir cualquier array como parámetro e imprima su primer elemento.	5
1.6.3	Crea una función que pueda recibir cualquier array como parámetro e imprima todos sus elementos uno por uno (no se vale imprimir el array completo).	5

Table of contents

1.6.4	Crea una función que pueda recibir cualquier objeto como parámetro e imprima todos sus elementos uno por uno (no se vale imprimir el objeto completo).	5
1.7	¿Cómo te fue?	5

1 Test de JavaScript.

¡Es hora de poner a prueba cuánto sabes sobre JavaScript!

Esta lectura es una prueba de JavaScript. A diferencia de un examen, nadie te obligará a nada. **Puedes hacer trampa**, ese es el camino fácil. Pero tengo mucha fe en ti, confío en que seguirás estos consejos.

1.1 Instrucciones para tomar esta prueba

- Evalúa muy críticamente tu conocimiento.
- Si logras resolver la prueba, no importa cuánto te cueste, puedo asegurarte que tienes todo para continuar.
- Si no lo logras, no te preocupes, absolutamente nadie puede juzgarte, solo tú. anota los temas clave donde puedes mejorar, ubica las clases donde puedes aprenderlos y estudia vigorosamente.
- Es completamente válido hacer búsquedas en Google, cursos y tutoriales, incluso usar tus notas sin importar si es físico o virtual.
- Puede utilizar la siguiente referencia para reforzar los conceptos base Tutorial de JavaScript Básico.
- Antes de subir esta prueba, por favor realizar lo siguiente
 - Cree un repositorio en su cuenta de Github,
 - Suba el código de respuesta a esta prueba,
 - Genere un archivo comprimido con el código resultante,
 - Suba el código al entorno virtual,
 - Ubique en un comentario el repositorio donde está almacenado al momento de subir su esta actividad.

Recuerda que **el éxito no se mide por cuánto tiempo te toma aprender**, esa métrica es relativamente inútil. Mejor concéntrate desarrollar proyectos que realmente demuestran que dominas cada tecnología.

¡Mucha suerte!

1.2 Variables y operaciones

1.2.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es una variable y para qué sirve?
Una variable es un tipo de nombre que ocupa un espacio de memoria, donde se puede almacenar uno o varios tipos de datos de un programa.
- ¿Cuál es la diferencia entre declarar e inicializar una variable?

Al declarar una variable le estamos diciendo al programa desde donde empieza a existir esa variable, así como su nombre y su tipo de dato, por otro lado, al momento de inicializar una variable le estamos dando a un nombre que ya está creado o se crea ese momento un valor predeterminado.

- ¿Cuál es la diferencia entre sumar números y concatenar strings?

En el momento de que trato de sumar dos o más números lo que hace es por ejemplo $1 + 4 = 5$, y por el lado de concatenar strings, lo que sucede es que va a unir las palabras o las letras para formar una sola oración o cadena de strings, puede ser “Hola” y concatenar con “mundo” al suceder eso formaremos “Hola mundo”, entonces, su principal diferencia es el manejo de cada uno de los tipos de variables.

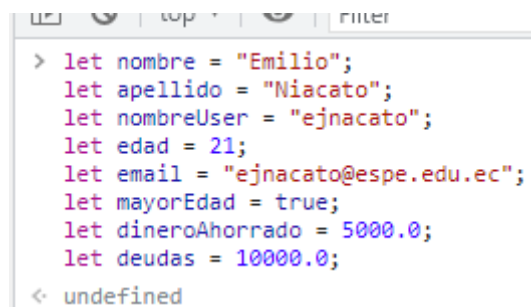
- ¿Cuál operador me permite sumar o concatenar?

Ya sea para concatenar o sumar variables lo que se usa en JavaScript es el operador suma “+”

1.2.2 Determina el nombre y tipo de dato para almacenar en variables la siguiente información:

- Nombre: string
- Apellido: string
- Nombre de usuario en ESPE: string
- Edad: number(entero)
- Correo electrónico: string
- Mayor de edad: booleano
- Dinero ahorrado: number(decimal)
- Deudas: number(decimal)

1.2.3 Traduce a código JavaScript las variables del ejemplo anterior y deja tu código en los comentarios.



```
> let nombre = "Emilio";
let apellido = "Niacato";
let nombreUser = "ejnacato";
let edad = 21;
let email = "ejnacato@espe.edu.ec";
let mayorEdad = true;
let dineroAhorrado = 5000.0;
let deudas = 10000.0;
< undefined
```

1.2.4 Calcula e imprime las siguientes variables a partir de las variables del ejemplo anterior:

- Nombre completo (nombre y apellido)

- Dinero real (dinero ahorrado menos deudas)

```
let nombre = "Emilio";
let apellido = "Niacato";
let nombreUser = "ejnacato";
let edad = 21;
let email = "ejnacato@espe.edu.ec";
let mayorEdad = true;
let dineroAhorrado = 5000.0;
let deudas = 10000.0;

console.log("Su nombre es " + nombre + " y su apellido es " + apellido);
let dinero = dineroAhorrado - deudas;
console.log("Usted tiene un total de dinero en $ de: " + dinero);

Su nombre es Emilio y su apellido es Niacato
Usted tiene un total de dinero en $ de: -5000
undefined
```

1.3 Funciones

1.3.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es una función?
Una función es un bloque de código o subprograma que es posible ejecutarlo sin la necesidad de volver a escribir el código cada vez, y tiene como por objetivo el de reutilizar este código una o varias veces en varias partes del programa.
- ¿Cuándo me sirve usar una función en mi código?
Se lo usa normalmente cuando en el programa se va a repetir el mismo código en distintos lugares del programa, así, por lo tanto, se puede utilizar el código dentro de una función las veces que sea necesario.
- ¿Cuál es la diferencia entre parámetros y argumentos de una función?
Los argumentos son aquellos valores que son enviados al llamar a la función, en cambio los parámetros son los elementos o valores que la función espera recibir a la hora de ser llamada. Es posible enviar, por ejemplo, dos argumentos, pero la función requiere 3 parámetros, por lo que se hace esta distinción.

1.3.2 Convierte el siguiente código en una función, pero, cambiando cuando sea necesario las variables constantes por parámetros y argumentos en una función:

```
const name = "Diego Medardo";
const lastname = "Saavedra
García"; const completeName =
```

```
name + lastname; const nickname  
= "statick";  
  
console.log("Mi nombre es "  
+ completeName  
+ ", pero prefiero que me digas "  
+ nickname + ".");
```

```
> function persona(name, lastname, nickname) {  
    const completeName = name + " " + lastname;  
    return "Mi nombre es " + completeName + ", pero prefiero que me digas " + nickname + ".";  
}  
  
const name = "Diego Medardo";  
const lastname = "Saavedra García";  
const nickname = "statick";  
  
console.log(persona(name, lastname, nickname));  
Mi nombre es Diego Medardo Saavedra García, pero prefiero que me digas statick. VM2
```

1.4 Condicionales

1.4.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es un condicional?
Un condicional son sentencias en cual nos ayuda a tomar decisiones entre distintas opciones en base a valores de verdad, esto permite cambiar el orden de ejecución de las sentencias de programación, e incluso saltarse algunas de ellas.
- ¿Qué tipos de condicionales existen en JavaScript y cuáles son sus diferencias?
Los más comunes que existen son: if, if else, switch, for y while.
If: Quiere decir la sentencia “si” en caso que la sentencia sea verdadera ejecutara una decisión ya sea desde imprimir un mensaje, hacer un calculo o ejecutar otra sentencia, esto sucederá siempre y cuando la sentencia de entrada sea verdadera.
If else: Es similar a if else, primero analizará en caso que sea verdadera ejecutará una acción, pero en caso de ser falsa este ejecutará otra acción distinta.
Switch: Este es más usado en el caso de tener varias opciones, en este caso se lo usará cuando se tiene muchas condiciones encadenadas llenas de decisiones, se lo suele usar más como un menú de opciones.
For: Esta condición es de ciclo repetitivo donde ejecutara una acción repetidamente hasta que llegue a su limite de entrada, ya se imprimir varios mensajes o incrementar un valor de una variable. Se

utiliza cuando se conoce el número de veces que se repetirá el ciclo, o se puede conocer mediante otras funciones del lenguaje como length.

While: Es un ciclo repetitivo que permite ejecutar un código de forma repetitiva siempre que una condición sea verdadera y hasta que esa condición sea falsa.

- ¿Puedo combinar funciones y condicionales?

Si, porque dentro de una condición ya se de un if, if else, for, while, entre otros, se puede ejecutar una acción, y esta acción puede contener el ejecutar una función que calcule, imprima o realice alguna fracción de código en concreto.

1.4.2 Replica el comportamiento del siguiente código que usa la sentencia switch utilizando if, else y else if:

```
const tipoDeSuscripcion = "Basic";

switch (tipoDeSuscripcion) { case "Free":
  console.log("Solo puedes tomar los cursos
  gratis"); break;
  case "Basic": console.log("Puedes tomar casi todos los cursos de
  ESPE durante un mes"); break;
  case "Expert": console.log("Puedes tomar casi todos los cursos de
  ESPE durante un año"); break;
  case "ExpertPlus":
    console.log("Tú y alguien más pueden tomar TODOS los
    cursos de ESPE durante un año");
    break;
}
```

```
> const tipoDeSuscripcion = "Basic";
  if (tipoDeSuscripcion == "Free") {
    console.log("Solo puedes tomar los cursos gratis");
  } else if (tipoDeSuscripcion == "Basic") {
    console.log("Puedes tomar casi todos los cursos de ESPE durante un mes");
  } else if (tipoDeSuscripcion == "Expert") {
    console.log("Puedes tomar casi todos los cursos de ESPE durante un año");
  } else if (tipoDeSuscripcion == "ExpertPlus") {
    console.log(
      "Tú y alguien más pueden tomar TODOS los cursos de ESPE durante un año"
    );
  }
}
```

```
Puedes tomar casi todos los cursos de ESPE durante un mes
```

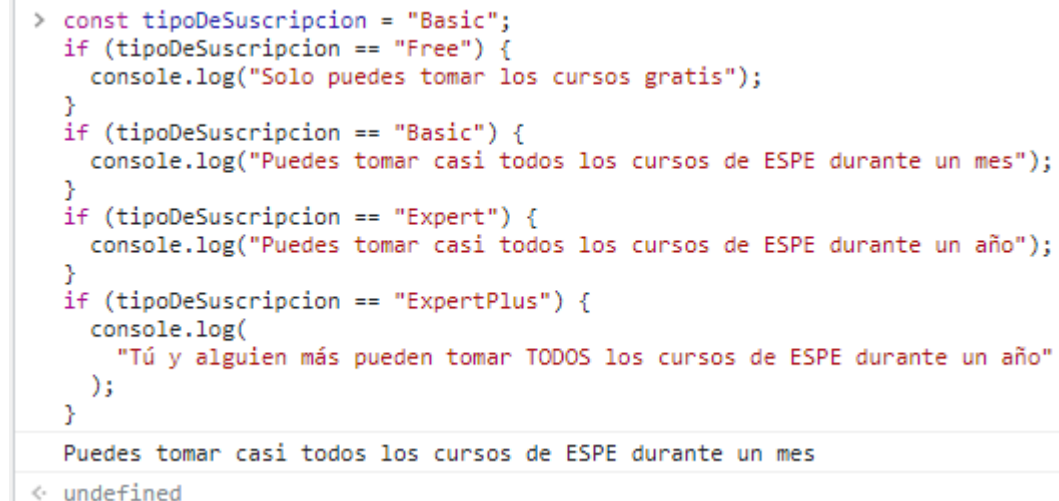
```
< undefined
```


1.4.3 Replica el comportamiento de tu condicional anterior con if, else y else if, pero ahora solo con if (sin else ni else if).

Bonus: si ya eres una experta o experto en el lenguaje, te desafío a comentar cómo replicar este comportamiento con arrays y un solo condicional.

```
const tipoDeSuscripcion = "Basic";

switch (tipoDeSuscripcion) { case "Free":
  console.log("Solo puedes tomar los cursos
  gratis"); break;
  case "Basic": console.log("Puedes tomar casi todos los cursos de
  ESPE durante un mes"); break;
  case "Expert": console.log("Puedes tomar casi todos los cursos de
  ESPE durante un año"); break;
  case "ExpertPlus":
    console.log("Tú y alguien más pueden tomar TODOS los
    cursos de ESPE durante un año");
    break;
}
```



```
> const tipoDeSuscripcion = "Basic";
  if (tipoDeSuscripcion == "Free") {
    console.log("Solo puedes tomar los cursos gratis");
  }
  if (tipoDeSuscripcion == "Basic") {
    console.log("Puedes tomar casi todos los cursos de ESPE durante un mes");
  }
  if (tipoDeSuscripcion == "Expert") {
    console.log("Puedes tomar casi todos los cursos de ESPE durante un año");
  }
  if (tipoDeSuscripcion == "ExpertPlus") {
    console.log(
      "Tú y alguien más pueden tomar TODOS los cursos de ESPE durante un año"
    );
  }
  Puedes tomar casi todos los cursos de ESPE durante un mes
  < undefined
```

1.5 Ciclos

1.5.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es un ciclo?
Un ciclo es una fracción de código que se repetirá varias veces ejecutando una acción en concreto, existen funciones donde nosotros podemos dar un inicio o un final del ciclo, así mismo como tenemos ciclos que solamente se ejecutará sin saber cuántas veces se lo realizará.
- ¿Qué tipos de ciclos existen en JavaScript?
Los comúnmente más usados son: For, while y do while
- ¿Qué es un ciclo infinito y por qué es un problema?
Un ciclo infinito es una fracción de código que se ejecutara varias veces sin fin, comúnmente eso pasa cuando no se define la condición para finalizar el ciclo, o no llega a suceder. Los más grandes problemas que puede ocasionar es que la memoria se satura de información de forma indefinida hasta finalizar el programa.
¿Puedo mezclar ciclos y condicionales?
Sí, ya que es posible anidar ciclos repetitivos dentro de condicionales, o viceversa.

1.5.2 Replica el comportamiento de los siguientes ciclos for utilizando ciclos while:

```
for (let i = 0; i < 5; i++) {  
  console.log("El valor de i es: " +  
    i);  
}
```

```
for (let i = 10; i >= 2; i--) {  
  console.log("El valor de i es: " +  
    i);  
}
```

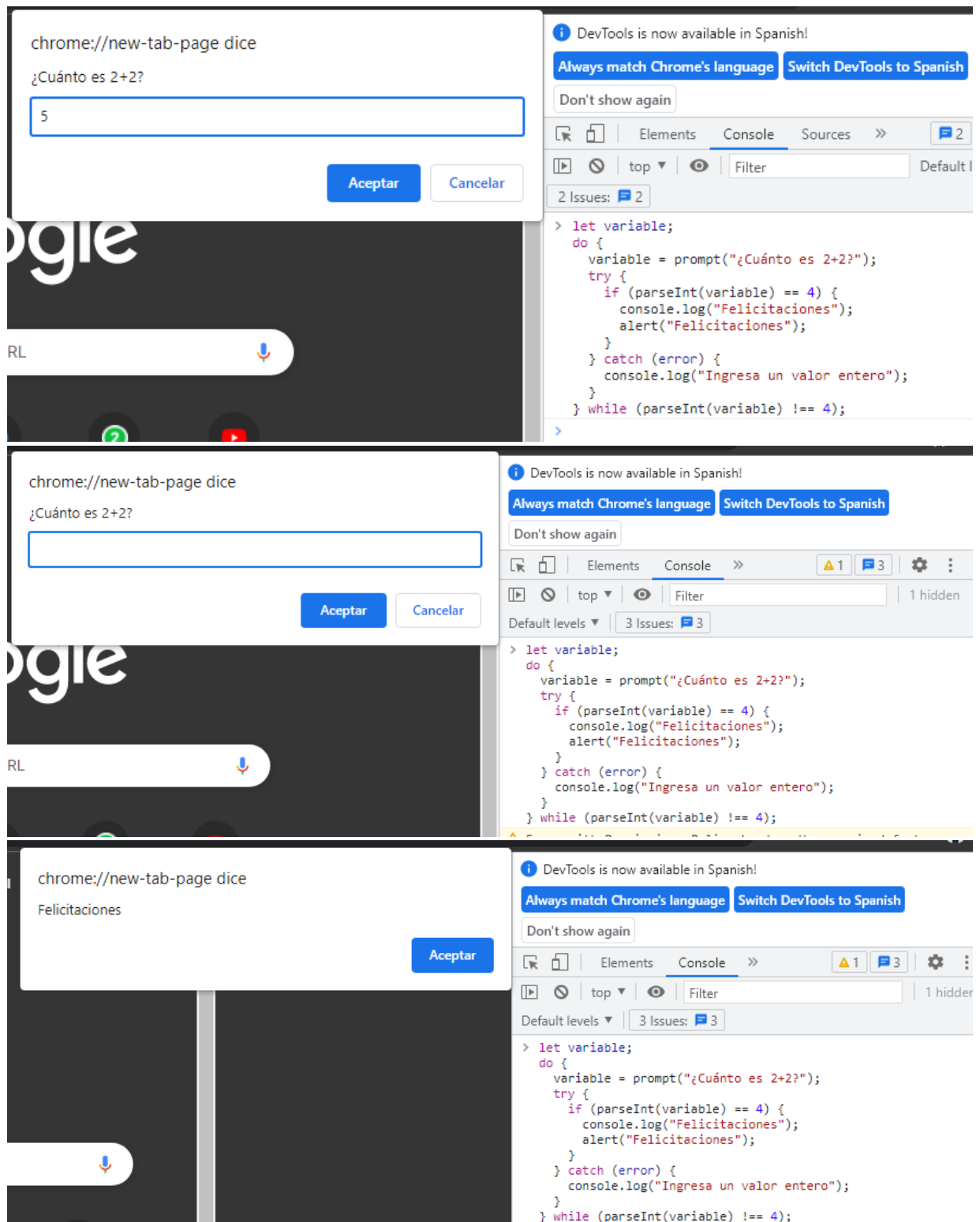
1 Test de JavaScript.

<pre>> let i = 0; while (i < 5) { console.log("El valor de i es: " + i); i++; } i = 10; while (i >= 2) { console.log("El valor de i es: " + i); i--; }</pre>	
El valor de i es: 0	VM541:3
El valor de i es: 1	VM541:3
El valor de i es: 2	VM541:3
El valor de i es: 3	VM541:3
El valor de i es: 4	VM541:3
El valor de i es: 10	VM541:9
El valor de i es: 9	VM541:9
El valor de i es: 8	VM541:9
El valor de i es: 7	VM541:9
El valor de i es: 6	VM541:9
El valor de i es: 5	VM541:9
El valor de i es: 4	VM541:9
El valor de i es: 3	VM541:9
El valor de i es: 2	VM541:9
< 2	

1.5.3 Escribe un código en JavaScript que le pregunte a los usuarios cuánto es 2 + 2. Si responden bien, mostramos un mensaje de felicitaciones, pero si responden mal, volvemos a empezar.

Pista: puedes usar la función prompt de JavaScript.

1 Test de JavaScript.

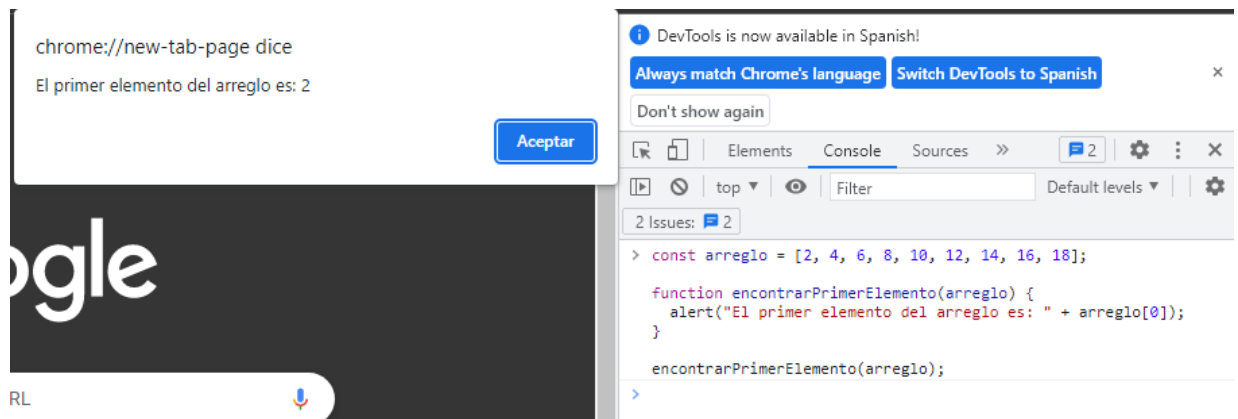


1.6 Listas

1.6.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es un array?
Un array es una conjunto de elementos ordenados en el cual se pueden almacenar varios tipos de datos o incluso se puede guardar arreglos dentro de arreglos, formando así matrices.
- ¿Qué es un objeto?
Un objeto es un
- ¿Cuándo es mejor usar objetos o arrays?
- ¿Puedo mezclar arrays con objetos o incluso objetos con arrays?

1.6.2 Crea una función que pueda recibir cualquier array como parámetro e imprima su primer elemento.



1.6.3 Crea una función que pueda recibir cualquier array como parámetro e imprima todos sus elementos uno por uno (no se vale imprimir el array completo).

1 Test de JavaScript.

```
> const arreglo = [2, 4, 6, 8, 10, 12, 14, 16, 18];
for (i = 0; i < arreglo.length; i++) {
  console.log("Elemento " + i + ": " + arreglo[i]);
}
```

Elemento 0: 2	VM725:4
Elemento 1: 4	VM725:4
Elemento 2: 6	VM725:4
Elemento 3: 8	VM725:4
Elemento 4: 10	VM725:4
Elemento 5: 12	VM725:4
Elemento 6: 14	VM725:4
Elemento 7: 16	VM725:4
Elemento 8: 18	VM725:4
< undefined	

1.6.4 Crea una función que pueda recibir cualquier objeto como parámetro e imprima todos sus elementos uno por uno (no se vale imprimir el objeto completo).

```
> let objetoAutos = {
  marca: "Volkswagen Polo",
  precio: "20000",
  color: "rojo",
  anio: 2010,
};

function mostrarElementos(objetoAutos) {
  Object.keys(objetoAutos).forEach(function (valor) {
    console.log(valor + ": " + objetoAutos[valor]);
    return valor;
  });
}
mostrarElementos(objetoAutos);
```

marca: Volkswagen Polo
precio: 20000
color: rojo
anio: 2010

1.7 ¿Cómo te fue?

¡Felicidades por completar la prueba de JavaScript! Confío en que hayas completado cada paso y hayas podido repasar los temas de los ejercicios que se te complicaron.