

Elaborado por:

Lcdo. Diego Medardo Saavedra García, Mgtr.

Profesor Ocasional de la Universidad de las Fuerzas Armadas ESPE
Departamento de Ciencias de la Computación.

Nombre:

- Tipanguano Samantha

NRC: 10047

Fecha: 31/05/2023

○ **Variables y operaciones**

■ **Responde las siguientes preguntas en la sección de comentarios:**

● **¿Qué es una variable y para qué sirve?**

Una variable es un “contenedor” de datos, que se utiliza para guardar los datos, se utiliza variables para almacenar datos que serán de utilidad, o serán llamados dentro del código.

● **¿Cuál es la diferencia entre declarar e inicializar una variable?**

La diferencia con respecto a declarar es indicar un nombre y un tipo de dato dentro del lenguaje en que estemos trabajando sin asignarle un valor en específico. En cambio, inicializar una variable es darle un valor específico al dato. Las 2 son muy importantes a la hora de programar.

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

Sumar significa realizar una operación aritmética entre números con el fin de obtener otro como resultado, mientras que concatenar quiere decir juntar string o cadenas de manera que se obtiene una cadena más grande.

● **¿Cuál operador me permite sumar o concatenar?**

El símbolo de “+”

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

- Nombre -> Variable: “Nombre” de tipo: “cadena” (string).
- Apellido -> Variable: “Apellido” de tipo: “cadena” (string).
- Nombre de usuario en ESPE -> Ejemplo: sntipanguano -> Variable: “sntipanguano” de tipo: “cadena” (string)

- Edad -> variable: "edad" de tipo "entero" (integer)
- Correo electrónico -> Variable "correoElectronico" de tipo "cadena" (string)
- Mayor de edad -> Variable "mayorEdad" de tipo booleano (boolean)
- Dinero ahorrado -> Variable "dineroAhorrado" de tipo flotante (float)
- Deudas -> Variable "deuda" de tipo flotante (float)

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

```
1 //Samantha Tipanguano
2 //Nombre -> Variable: "Nombre" de tipo: "cadena" (string).
3 // Apellido -> Variable: "Apellido" de tipo: "cadena" (string).
4 //Nombre de usuario en ESPE -> Ejemplo: sntipanguano -> Variable:"sntipanguano" de tipo: "cadena" (string)
5 // Edad -> variable: "edad" de tipo "entero" (integer)
6 // Correo electrónico -> Variable "correoElectronico" de tipo "cadena" (string)
7 // Mayor de edad -> Variable "mayorEdad" de tipo booleano (boolean)
8 // Dinero ahorrado -> Variable "dineroAhorrado" de tipo flotante (float)
9 // Deudas -> Variable "deuda" de tipo flotante (float)
10
11 var Nombre = "Santiago";
12 var Apellido = "Tipanguano";
13 var usuarioESPE = "sntipanguano";
14 var Edad = 25;
15 var correoElectronico = "ejemplo@example.com";
16 var mayorEdad = true;
17 var dineroAhorrado = 1000.50;
18 var deuda = 500.25;
19 |
```

■ **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)



```
l.24.js > ...  
//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)  
  
// Variables anteriores  
var Nombre = "Santiago";  
var Apellido = "Tipanguano";  
var usuarioESPE = "sntipanguano";  
var Edad = 25;  
var correoElectronico = "ejemplo@example.com";  
var mayorEdad = true;  
var dineroAhorrado = 1000.50;  
var deuda = 500.25;  
  
// Nombre completo calculado  
var nombreCompleto = Nombre + " " + Apellido;  
  
// Dinero real calculado  
var dineroReal = dineroAhorrado - deuda;  
  
// variables calculadas  
console.log("Nombre completo: " + nombreCompleto);  
console.log("Dinero real: " + dineroReal);
```

Resultado:

```
var Edad = 25;  
var correoElectronico = "ejemplo@example.com";  
var mayorEdad = true;  
var dineroAhorrado = 1000.50;  
var deuda = 500.25;  
  
// Nombre completo calculado  
var nombreCompleto = Nombre + " " + Apellido;  
  
// Dinero real calculado  
var dineroReal = dineroAhorrado - deuda;  
  
// variables calculadas  
console.log("Nombre completo: " + nombreCompleto);  
console.log("Dinero real: " + dineroReal);  
Nombre completo: Santiago Tipanguano  
Dinero real: 500.25  
< undefined  
>
```

○ Funciones

- Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es una función?

Es como una herramienta creada para una función en específica y esta se puede utilizar en otras secciones del código

- ¿Cuándo me sirve usar una función en mi código?

Sirve para poder reutilizar esta función en varias partes del código dando así una buena estructura y organización del código.

- ¿Cuál es la diferencia entre parámetros y argumentos de una función?

Los marcadores son parámetros de posición o el valor que la función espera recibir, mientras que los argumentos son valores reales que se pasan a la función cuando esta es llamada.

- **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 + ".");
```

```
1.3.2js > ...
1 //Samantha Tipanguano
2 //1.3.2 Convierte el siguiente código en una función, pero, cambiando cuando
3 //sea necesario las variables constantes por parámetros y argumentos en una función:
4 //const name = "Diego Medardo";
5 //const lastname = "Saavedra García";
6 //const completeName = name + lastname;
7 //const nickname = "statick";
8 //console.log("Mi nombre es "
9 //+ completeName
10 //+ ", pero prefiero que me digas "
11 //+ nickname + ".");
12
13
14 function imprimir(nombre, apellido, apodo) {
15     const completeName = nombre + " " + apellido;
16     console.log("Mi nombre es " + completeName + ", pero prefiero que me digas " + apodo + ".");
17 }
18
19 // Llamada a la función
20 imprimir("Diego Medardo", "Saavedra García", "statick");
21
22
```

Resultado:

```
> function imprimir(nombre, apellido, apodo) {  
    const completeName = nombre + " " + apellido;  
    console.log("Mi nombre es " + completeName + ", pero prefiero que me  
    digas " + apodo + ".");  
}  
  
// Llamada a la función  
imprimir("Diego Medardo", "Saavedra García", "statick");  
  
Mi nombre es Diego Medardo Saavedra García, pero prefiero que me digas statick. VM796:3  
  
< undefined
```

- Condicionales

■ Responde las siguientes preguntas en la sección de comentarios:

● ¿Qué es un condicional?

Es una estructura de control que permite tomar decisiones al código en base a una condición establecida.

● ¿Qué tipos de condicionales existen en JavaScript y cuáles son sus diferencias?

- **if:** Permite ejecutar un bloque de código si una condición es verdadera.
- **if-else:** Permite ejecutar un bloque de código si una condición es verdadera y otro bloque si es falsa.
- **if-else if-else:** Permite evaluar múltiples condiciones y ejecutar diferentes bloques de código según el resultado de cada una.
- **switch:** Permite evaluar una expresión y ejecutar diferentes bloques de código según el valor de la expresión.

● ¿Puedo combinar funciones y condicionales?

Sí, si se puede para lograr dar condición a distintos bloques del código se puede crear un condicional dentro de cada función y se puede realizar lo mismo de forma inversa que sería llamar a la función dentro de una condición.

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

javascript

```
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");

}
```

```
1.4.2.js > ...
1 //Samantha Tipanguano
2 //1.4.2 Replica el comportamiento del siguiente código que usa la sentencia
3 //switch utilizando if, else y else if:
4 //const tipoDeSuscripcion = "Basic";
5 //switch (tipoDeSuscripcion) {
6 //case "Free":
7 //console.log("Solo puedes tomar los cursos gratis");
8 //break;
9 //case "Basic":
10 //console.log("Puedes tomar casi todos los cursos de ESPE durante un mes");
11 //break;
12 //case "Expert":
13 //console.log("Puedes tomar casi todos los cursos de ESPE durante un año");
14 //break;
15 //case "ExpertPlus":
16 //console.log("Tú y alguien más pueden tomar TODOS los cursos
17 //de ESPE durante un año");
18 //break;
19
20
21 const tipoDeSuscripcion = "Basic";
22
23 if (tipoDeSuscripcion === "Free") {
24     console.log("Solo puedes tomar los cursos gratis");
25 } else if (tipoDeSuscripcion === "Basic") {
26     console.log("Puedes tomar casi todos los cursos de ESPE durante un mes");
27 } else if (tipoDeSuscripcion === "Expert") {
28     console.log("Puedes tomar casi todos los cursos de ESPE durante un año");
29 } else if (tipoDeSuscripcion === "ExpertPlus") {
30     console.log("Tú y alguien más pueden tomar TODOS los cursos de ESPE durante un año");
31 }
```

Resultado:



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



top Filter Default levels 1 Issue: 1

```
> 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 [VM745:6](#)

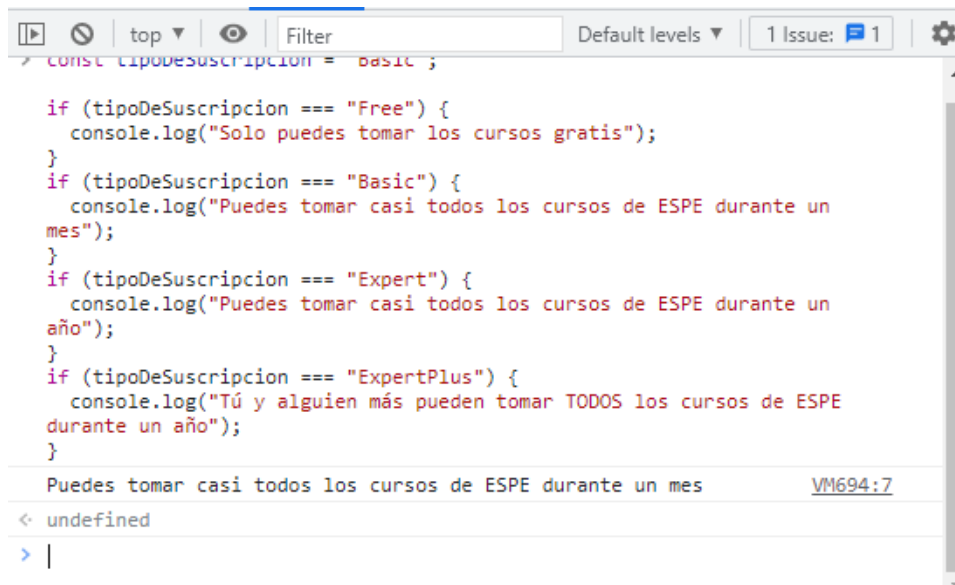
< undefined

>

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

```
1.4.3.js > ...
1 //Samantha Tipanguano
2 //1.4.3 Replica el comportamiento de tu condicional anterior con if, else y
3 //else if, pero ahora solo con if (sin else ni else if).
4
5
6 const tipoDeSuscripcion = "Basic";
7
8 if (tipoDeSuscripcion === "Free") {
9   console.log("Solo puedes tomar los cursos gratis");
10 }
11 if (tipoDeSuscripcion === "Basic") {
12   console.log("Puedes tomar casi todos los cursos de ESPE durante un mes");
13 }
14 if (tipoDeSuscripcion === "Expert") {
15   console.log("Puedes tomar casi todos los cursos de ESPE durante un año");
16 }
17 if (tipoDeSuscripcion === "ExpertPlus") {
18   console.log("Tú y alguien más pueden tomar TODOS los cursos de ESPE durante un año");
19 }
20
```

Resultado:



```
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 VM694:7

< undefined

> |

○ Ciclos

- Responde las siguientes preguntas en la sección de comentarios:

● ¿Qué es un ciclo?

Es un bucle que se detiene solo cuando cumple una condición en específico

● ¿Qué tipos de ciclos existen en JavaScript?

- ★ For: Se utiliza cuando se conoce el número de ciclos
- ★ While: Se ejecuta cuando la condición es verdadera, antes de cada ciclo se evalúa si es Verdadero o falso

★ Do While: Se ejecuta el proceso hasta que se cumpla una condición a diferencia del while este se ejecuta luego de cada ciclo.

● ¿Qué es un ciclo infinito y por qué es un problema?

Es cuando un proceso no termina y se repite infinitas veces y consume muchos recursos del sistema al consumir tiempo y memoria sin ningún fin.

● ¿Puedo mezclar ciclos y condicionales?

Si se puede ya que con esto se puede condicionar distintos ciclos en ciertos bloques del código esto permite que no se creen bucles y el proceso sea más eficiente

■ 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);  
  
}
```

```
1.5.2.js > ...  
1  < //Samantha Tipanguano  
2  //1.5.2 Replica el comportamiento de los siguientes ciclos for utilizando ciclos  
3  //while:  
4  //for (let i = 0; i < 5; i++) {  
5  //console.log("El valor de i es: " + i);  
6  //}  
7  //for (let i = 10; i >= 2; i--) {  
8  //console.log("El valor de i es: " + i);  
9  //}  
10  
11  // Ciclo for: i < 5  
12  let i = 0;  
13  < while (i < 5) {  
14    console.log("El valor de i es: " + i);  
15    i++;  
16  }  
17  
18  // Ciclo for: i >= 2  
19  let i = 10;  
20  < while (i >= 2) {  
21    console.log("El valor de i es: " + i);  
22    i--;  
23  }  
24
```

Resultado:

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

El valor de i es: 0	VM606:4
El valor de i es: 1	VM606:4
El valor de i es: 2	VM606:4
El valor de i es: 3	VM606:4
El valor de i es: 4	VM606:4

```
< 4
>
```

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

El valor de i es: 10	VM643:4
El valor de i es: 9	VM643:4
El valor de i es: 8	VM643:4
El valor de i es: 7	VM643:4
El valor de i es: 6	VM643:4
El valor de i es: 5	VM643:4
El valor de i es: 4	VM643:4
El valor de i es: 3	VM643:4
El valor de i es: 2	VM643:4

```
< 2
> |
```

⋮ Console What's New ✕

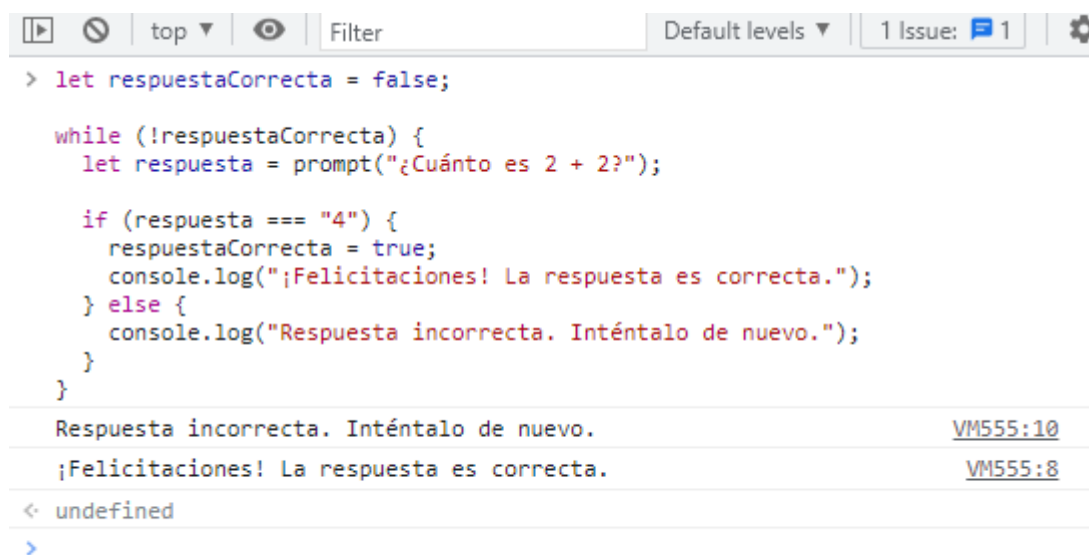
- Escribe un código en JavaScript que le pregunte a los usuarios cuánto es 2 + 2. Si responden bien, mostramos un mensaje de felicitaciones, pero si responden mal, volvemos a empezar.

```

1.5.3.js > ...
1 //Samantha Tipanguano
2 //1.5.3 Escribe un código en JavaScript que le pregunte a los usuarios cuánto
3 //es 2 + 2. Si responden bien, mostramos un mensaje de felicitaciones,
4 //pero si responden mal, volvemos a empezar.
5 // Pista: puedes usar la función prompt de JavaScript.
6
7 let respuestaCorrecta = false;
8
9 while (!respuestaCorrecta) {
10   let respuesta = prompt("¿Cuánto es 2 + 2?");
11
12   if (respuesta === "4") {
13     respuestaCorrecta = true;
14     console.log("¡Felicitaciones! La respuesta es correcta.");
15   } else {
16     console.log("Respuesta incorrecta. Inténtalo de nuevo.");
17   }
18 }
19

```

Resultado:



```

> let respuestaCorrecta = false;

while (!respuestaCorrecta) {
  let respuesta = prompt("¿Cuánto es 2 + 2?");

  if (respuesta === "4") {
    respuestaCorrecta = true;
    console.log("¡Felicitaciones! La respuesta es correcta.");
  } else {
    console.log("Respuesta incorrecta. Inténtalo de nuevo.");
  }
}

Respuesta incorrecta. Inténtalo de nuevo.
¡Felicitaciones! La respuesta es correcta.
< undefined
>

```

○ Listas

■ Responde las siguientes preguntas en la sección de comentarios:

● ¿Qué es un array?

Se podría decir que es como una lista que puede estar formado por letras, números, cadenas u objetos.

● ¿Qué es un objeto?

Es una entidad que une los métodos con propiedades usados para representar u organizar datos

• ¿Cuándo es mejor usar objetos o arrays?

Los objetos son útiles para almacenar datos con nombres descriptivos y asociar comportamientos a esos datos, mientras que los arrays son adecuados para almacenar listas ordenadas de elementos y realizar operaciones en conjunto con ellos.

• ¿Puedo mezclar arrays con objetos o incluso objetos con arrays?

Si se puede mezclar ya que con esto se obtiene un funcionamiento más completo, se puede acceder a los elementos de un array dentro de un objeto y se puede lograr también de manera inversa.

- Crea un array con los siguientes elementos: "Hola", "¿", "Cómo", "estás", "?".

```
1.6.2.js > ...
1 //Samantha Tipanguano
2 //Crea un array con los siguientes elementos: "Hola", "¿", "Cómo", "estás", "?"
3
4 const miArray = ["Hola", "¿", "Cómo", "estás", "?"];
5 console.log(miArray);
6
```

Resultado:

```
> const miArray = ["Hola", "¿", "Cómo", "estás", "?"];
  console.log(miArray);
  ▶ (5) ['Hola', '¿', 'Cómo', 'estás', '?'] VM46:2
< undefined
```

- Accede al primer elemento del array que creaste en el ejercicio anterior e imprímelo en la consola.

```
1.6.3.js > ...
1 //Samantha Tipanguano
2 //Accede al primer elemento del array que creaste en el ejercicio anterior e imprímelo en
3 //la consola.
4
5 const miArray = ["Hola", "¿", "Cómo", "estás", "?"];
6 const primerElemento = miArray[0];
7 console.log(primerElemento);
8
```

Resultado:

```
> const miArray = ["Hola", "¿", "Cómo", "estás", "?"];
  const primerElemento = miArray[0];
  console.log(primerElemento);
  Hola VM84:3
< undefined
>
```

- Ahora supongamos que debes guardar datos de una persona. Crea un objeto con las siguientes propiedades: nombre, apellido, edad, hobbies (un array con varios elementos) y contacto (un objeto con las propiedades email y teléfono).

```
//Samantha Tipanguano
//Ahora supongamos que debes guardar datos de una persona. Crea un objeto con las
//siguientes propiedades: nombre, apellido, edad, hobbies (un array con varios elementos) y
//contacto (un objeto con las propiedades email y teléfono).

const persona = {
  nombre: "Juan",
  apellido: "Pérez",
  edad: 30,
  hobbies: ["correr", "leer", "cocinar"],
  contacto: {
    email: "juan@example.com",
    telefono: "123456789"
  }
};

console.log(persona);
```

Resultado:

```
3 Issues: 3
> const persona = {
  nombre: "Juan",
  apellido: "Pérez",
  edad: 30,
  hobbies: ["correr", "leer", "cocinar"],
  contacto: {
    email: "juan@example.com",
    telefono: "123456789"
  }
};

console.log(persona);

VM88:12
{nombre: 'Juan', apellido: 'Pérez', edad: 30, hobbies: Array(3), contact
o: {...}}
< undefined
> |
```

- Accede al valor de la propiedad hobbies y imprímelo en la consola.

```
1.6.5.js > ...
1 //Samantha Tipanguano
2 // Accede al valor de la propiedad hobbies y imprímelo en la consola
3
4
5 const persona = {
6   nombre: "Juan",
7   apellido: "Pérez",
8   edad: 30,
9   hobbies: ["correr", "leer", "cocinar"],
10  contacto: {
11    email: "juan@example.com",
12    telefono: "123456789"
13  }
14 };
15
16 console.log(persona.hobbies); // Notación de punto
17 console.log(persona["hobbies"]); // Notación de corchetes
18
```

Resultado:

```
> const persona = {
  nombre: "Juan",
  apellido: "Pérez",
  edad: 30,
  hobbies: ["correr", "leer", "cocinar"],
  contacto: {
    email: "juan@example.com",
    telefono: "123456789"
  }
};

console.log(persona.hobbies); // Notación de punto
console.log(persona["hobbies"]); // Notación de corchetes

▶ (3) ['correr', 'leer', 'cocinar'] VM92:12
▶ (3) ['correr', 'leer', 'cocinar'] VM92:13
< undefined
```

- Accede al valor de la propiedad email y imprímelo en la consola.



```
const persona = {
  nombre: "Juan",
  apellido: "Pérez",
  edad: 30,
  hobbies: ["correr", "leer", "cocinar"],
  contacto: {
    email: "juan@example.com",
    telefono: "123456789"
  }
};

console.log(persona.contacto.email); // Notación de punto
console.log(persona["contacto"]["email"]); // Notación de corchetes
```

Resultado:

```
> const persona = {
  nombre: "Juan",
  apellido: "Pérez",
  edad: 30,
  hobbies: ["correr", "leer", "cocinar"],
  contacto: {
    email: "juan@example.com",
    telefono: "123456789"
  }
};

console.log(persona.contacto.email); // Notación de punto
console.log(persona["contacto"]["email"]); // Notación de corchetes
```

juan@example.com

[VM96:12](#)

juan@example.com

[VM96:13](#)

< undefined