



# Tipos de datos y condicionales

## Curso de Fundamentos

### Introducción

Existen algunos tipos de datos extremadamente comunes que encontrará en JavaScript, y estas lecciones sobre los aspectos básicos le brindarán una base muy sólida en todos ellos. Sin embargo, antes de comenzar a profundizar, lea esta [descripción general de los tipos de datos más comunes en JavaScript](#).

### Resumen de la lección

Esta sección contiene una descripción general de los temas que aprenderá en esta lección.

- Nombra los ocho tipos de datos en JavaScript.
- Comprenda la diferencia entre comillas simples, dobles y de acento grave.
- Incrustar una variable/expresión en una cadena.
- Entender qué es un método.
- Nombra los tres operadores lógicos.
- Comprenda qué son los operadores de comparación.
- Comprenda qué son los condicionales.
- Comprenda qué es la anidación.
- Comprenda qué son los valores verdaderos y falsos.

## Instrumentos de cuerda

Dependiendo del tipo de trabajo que estés realizando, es posible que termines trabajando más con fragmentos de texto que con números. Una **cadena** es un fragmento de texto... y es un componente fundamental del lenguaje.

1. Lea y codifique junto con este [tutorial de MDN sobre cadenas en JavaScript](#) . Omita esta **Concatenation in context** sección, ya que cubre conceptos que abordaremos en una lección posterior sobre manipulación del DOM.
2. Consulta [la lección de W3Schools sobre métodos de cadenas](#) para aprender un poco más sobre lo que puedes hacer con cadenas. Además, asegúrate de echar un vistazo a [la página de Referencia de cadenas](#) cerca de la parte inferior y realizar los ejercicios en la sección de tareas [a continuación](#) .
3. Tiempo de vocabulario: un **método** es un poco de funcionalidad incorporada en el lenguaje o en tipos de datos específicos. En [la lección de W3Schools sobre métodos de cadenas](#) , aprendiste acerca de algunos métodos que se pueden usar en cadenas, como **replace** y **slice** . Todavía hay muchos más métodos de cadenas incorporados, de los cuales [la documentación de MDN para cadenas proporciona una referencia exhaustiva](#) . No se espera que los memorices, pero la documentación será una referencia muy útil para volver a consultar, ¡así que agrégala a tus favoritos!

## Condicionales

Ahora es el momento de la parte divertida... Hasta ahora, no hemos hecho mucho con nuestra programación que no se pueda hacer con conocimientos matemáticos básicos. Claro, le hemos dicho a nuestra computadora cómo hacer los cálculos, lo que la hace más rápida, pero la esencia de la programación es enseñarle a la computadora cómo tomar decisiones para hacer cosas más complejas. Los condicionales son la forma de hacerlo.

1. El primer paso para aprender sobre condicionales es asegurarse de tener un buen conocimiento de [las comparaciones](#) .
2. W3Schools también tiene una [lección sobre condicionales en JavaScript](#) .
3. JavaScript.info has a [good tutorial on logical operators](#). A little heads up regarding this reading's tasks: there will be questions where you see **alert()** with a number or string inside the parenthesis. What's happening here will be discussed later in

the curriculum. Some of the answers may not make sense now, but they are accurate, and you will understand them as you progress in the curriculum. Don't worry too much about it now!

4. The [MDN article on conditionals](#) reinforces the concept and provides several interesting examples of how you could use it building websites.
5. JavaScript.info's [lesson on if/else](#) covers the same basic concept (read through it as a review!) and - more importantly - offers the usual 'tasks' at the bottom of the page!
6. [Learn about the switch statement](#), which is handy when you have multiple conditions.

## Assignment

To give you a good bit of practice, we have created replit.com exercises for you to play with. We believe it's best to practice programming on your *own* computer rather than in an online environment, but we'll get to that soon enough.

Be sure to do the lessons in the order presented here. Pressing "run" at the top will run the code. Read all directions, watch the terminal, and read all the errors. Don't forget to use 'console.log' extensively.

To get started, you will need to create a free [Replit](#) account. After you have done that, you can do the exercises down below by opening their link and clicking "Fork" or "Remix" in their Replit page. Note: Feel free to browse the files on the left column to gain familiarity with it.

### Replit fork limit

Replit now limits free accounts to having 3 repls at a time. If you reach the limit of 3 repls, you can delete one or more of the previous forks to create a room for the new one. To do so, go to [your repls](#) and delete what you no longer need.

### Replit and AI

Replit recently introduced an AI assistant, which is on by default. Before trying any of the exercises, you should first disable it, in order to prevent it

from spoiling the exercise. You can do so by clicking on the button labeled AI in the bottom left corner of the code view and then unchecking the "Enable" checkbox.

- [Exercise 1](#)

- In this exercise, you will be working out of the file called `troubleshooting.js`.

- [Exercise 2](#)

- You will be working out of `script.js`, and you will use the console in the 'webview' pane to check your work. To access the console, click the wrench icon, which is located on the right side of the address bar within the 'webview' pane.

- [Exercise 3](#)

- You will be working out of `math.js`.

- [Exercise 4](#)

- You will be working out of `follow.js`.

## Knowledge check

The following questions are an opportunity to reflect on key topics in this lesson. If you can't answer a question, click on it to review the material, but keep in mind you are not expected to memorize or master this knowledge.

- [What are the eight data types in JavaScript?](#)
- [Which data type is NOT primitive?](#)
- [What is the relationship between null and undefined?](#)
- [What is the difference between single, double, and backtick quotes for strings?](#)
- [¿Cuál es el término para unir cuerdas?](#)
- [¿Qué tipo de cita le permite incrustar variables/expresiones en una cadena?](#)
- [¿Cómo se incrustan variables/expresiones en una cadena?](#)
- [¿Cómo se utilizan caracteres de escape en una cadena?](#)

- [¿Cuál es la diferencia entre los métodos de cadena slice/substring?](#)
- [¿Cuáles son los tres operadores lógicos y qué representan?](#)
- [¿Qué son los operadores de comparación?](#)
- [¿Qué son los valores verdaderos y falsos?](#)
- [¿Cuáles son los valores falsos en JavaScript?](#)
- [¿Qué son los condicionales?](#)
- [¿Cuál es la sintaxis de un condicional if/else?](#)
- [¿Cuál es la sintaxis de una declaración switch?](#)
- [¿Cuál es la sintaxis de un operador ternario?](#)
- [¿Qué es la anidación?](#)

## Recursos adicionales

Esta sección contiene enlaces útiles a contenido relacionado. No es obligatoria, por lo que se la puede considerar complementaria.

- [Las expresiones regulares](#) , comúnmente conocidas como regex, son una herramienta que busca o localiza patrones en cadenas para su validación. Sin embargo, es importante saber [cuándo no se deben usar expresiones regulares](#) . Existen otros métodos para procesar cadenas y, en comparación, las expresiones regulares pueden ser más lentas.
- [Expresiones regulares de Web Dev Simplified en 20 minutos](#)

 [Mejorar en GitHub](#)

 [Informar un problema](#)

[Ver registro de cambios de la lección](#)



Ver curso

Marcar como completado



Próxima lección

# ¡Apóyanos!

El Proyecto Odin está financiado por la comunidad. ¡Únase a nosotros para ayudar a estudiantes de todo el mundo apoyando el Proyecto Odin!

[Más información](#)[Dona ahora](#)

THE ODIN PROJECT

Educación en codificación de alta calidad mantenida por una comunidad de código abierto.



## Sobre nosotros

[Acerca de](#)[Equipo](#)[Blog](#)[Casos de éxito](#)

## Apoyo

[Preguntas frecuentes](#)[Contribuir](#)[Contáctenos](#)

## Guías

[Guías de la comunidad](#)[Guías de instalación](#)

## Legal

[Términos](#)[Privacidad](#)

© 2025 El Proyecto Odin. Todos los derechos reservados.