



Matrices y bucles

Curso de Fundamentos

Introducción

Uno de los desafíos de la programación es trabajar con grandes cantidades de datos. Por ejemplo, si quieres almacenar los nombres de todos los estudiantes de tu clase, ¿cómo lo harías? Podrías crear una variable para cada nombre, pero eso sería tedioso e ineficiente. También sería difícil de administrar y actualizar. ¿Qué pasa si quieres cambiar o acceder a los nombres más tarde?

Afortunadamente, existe una mejor manera de manejar este problema. En esta lección, aprenderá acerca de las matrices, que son estructuras de datos que pueden almacenar múltiples valores en una sola variable. Las matrices son muy útiles para organizar y manipular grandes cantidades de datos. También aprenderá acerca de los bucles, que son estructuras de control que le permiten ejecutar un bloque de código repetidamente. Los bucles son muy útiles para realizar la misma operación en cada elemento de una matriz. Finalmente, se le presentará el desarrollo impulsado por pruebas (TDD), que es la práctica de escribir pruebas para su código antes de escribir el código en sí.

Resumen de la lección

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

- Usando matrices.

- Usando métodos de matriz incorporados.
- Usando bucles.
- Ensuciarse las manos con ejercicios TDD.

Matrices

Las cadenas y los números pueden ser los elementos básicos, pero a medida que sus scripts se vuelvan más complejos, necesitará una forma de manejar grandes cantidades de ellos. Afortunadamente, JavaScript tiene un par de tipos de datos que se utilizan precisamente para eso. Una matriz es una colección ordenada de elementos (cadenas, números u otras cosas).

1. Revise [las matrices de JavaScript de W3Schools](#) .
2. [Métodos de matriz de JavaScript](#) cubre algunos de los métodos de matriz integrados más útiles. Estos conceptos básicos son algo que usará todos los días, así que no se apresure demasiado y se los pierda.
3. Este [curso intensivo sobre matrices de JavaScript](#) explica una descripción general de las matrices en JavaScript en aproximadamente 6 minutos.

Bucles

Las computadoras no se cansan y son muy, *muy* rápidas. Por eso, son muy adecuadas para resolver problemas que implican realizar cálculos varias veces. En algunos casos, una computadora podrá repetir una tarea *miles* o incluso *millones* de veces en tan solo unos pocos segundos, mientras que a un humano le llevaría muchas horas. (Obviamente, la velocidad aquí depende de la complejidad del cálculo y de la velocidad de la computadora en sí). Una forma de hacer que una computadora realice una tarea repetitiva es utilizando un **bucle** .

1. Lee [el código de bucle de MDN](#) . Es más extenso, pero asegúrate de abordar las secciones de "Aprendizaje activo" en la parte inferior de la página.
2. Once again, same info, slightly different context from [JavaScript.info's "Loops: While and For"](#). Be sure to do the exercises at the end of the page. You learn best by doing.

Test Driven Development

Test Driven Development (TDD) is a phrase you often hear in the dev world. It refers to the practice of writing automated tests that describe how your code should work before you actually write the code. For example, if you want to write a function that adds a couple of numbers, you would first write a test that uses the function and supplies the expected output. The test will fail before you write your code, and you should be able to know that your code works correctly when the test passes.

In many ways, TDD is much more productive than writing code without tests. If we didn't have the test for the adding function above, we would have to run the code ourselves over and over, plugging in different numbers until we were sure that it was working... not a big deal for a basic `add(2, 2)`, but imagine having to do that for more complicated functions, like checking whether or not someone has won a game of tic tac toe: `(game_win(["o", null,"x",null,"x",null,"x", "o", "o"]))`. If you didn't do TDD, then you might actually have to play multiple games against yourself just to test if the function was working correctly!

We will teach you the art of actually writing these tests later in the course. The following practice has the tests already written out for you. All you have to do is set up the testing environment, read the specs, and write the code that makes them pass!

Assignment

1. Go and review the [README of our javascript-exercises repository](#) to set up your local environment. Once you have cloned the repository and installed Jest, review each README file prior to completing the following exercises in order:
 - `01_helloWorld` (This exercise is intentionally very beginner friendly to ensure that you have set up everything properly!)
 - `02_repeatString`
 - `03_reverseString`
 - `04_removeFromArray`
 - `05_sumAll`
 - `06_leapYears`
 - `07_tempConversion`

Note: Solutions for these exercises can be found in the [solution](#) folder of each exercise.

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 is an array?](#)
- [What are arrays useful for?](#)
- [How do you access an array element?](#)
- [How do you change an array element?](#)
- [What are some useful array properties?](#)
- [What are some useful array methods?](#)
- [What are loops useful for?](#)
- [What is the break statement?](#)
- [What is the continue statement?](#)
- [What is the advantage of writing automated tests?](#)

Additional resources

This section contains helpful links to related content. It isn't required, so consider it supplemental.

- It looks like this lesson doesn't have any additional resources yet. Help us expand this section by contributing to our curriculum.

 [Improve on GitHub](#)  [Report an issue](#)

[See lesson changelog](#)



View Course

[Mark Complete](#)[→ Next Lesson](#)

¡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](#)

Guías

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

Legal

[Términos](#)[Privacidad](#)

[Contribuir](#)

[Contáctenos](#)

© 2025 El Proyecto Odin. Todos los derechos reservados.