what is the difference between storing data in a variable and storing it in a state? (support your answer with code samples)

After reading the last couple of articles you should now know what JavaScript is, what it can do for you, how you use it alongside other web technologies, and what its main features look like from a high level. In this article, we will get down to the real basics, looking at how to work with the most basic building blocks of JavaScript — Variables.

[**What is a variable?**](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#what_is_a_variable)

A variable is a container for a value, like a number we might use in a sum, or a string that we might use as part of a sentence.

<button id="button\_A">Press me</button>

<h3 id="heading\_A"></h3>

const buttonA = document.querySelector("#button\_A");

const headingA = document.querySelector("#heading\_A");

buttonA.onclick = () => {

const name = prompt("What is your name?");

alert(`Hello ${name}, nice to see you!`);

headingA.textContent = `Welcome ${name}`;

};

In this example pressing the button runs some code. The first line pops a box up on the screen that asks the reader to enter their name, and then stores the value in a variable. The second line displays a welcome message that includes their name, taken from the variable value and the third line displays that name on the page.

### [Without a variable](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#without_a_variable)

To understand why this is so useful, let's think about how we'd write this example without using a variable. It would end up looking something like this:

<button id="button\_B">Press me</button>

<h3 id="heading\_B"></h3>

const buttonB = document.querySelector("#button\_B");

const headingB = document.querySelector("#heading\_B");

buttonB.onclick = () => {

alert(`Hello ${prompt("What is your name?")}, nice to see you!`);

headingB.textContent = `Welcome ${prompt("What is your name?")}`;

};

You may not fully understand the syntax we are using (yet!), but you should be able to get the idea. If we didn't have variables available, we'd have to ask the reader for their name every time we needed to use it!

Variables just make sense, and as you learn more about JavaScript they will start to become second nature.

One special thing about variables is that they can contain just about anything — not just strings and numbers. Variables can also contain complex data and even entire functions to do amazing things. You'll learn more about this as you go along.

## [Declaring a variable](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#declaring_a_variable)

To use a variable, you've first got to create it — more accurately, we call this declaring the variable. To do this, we type the keyword let followed by the name you want to call your variable:

let myName;

let myAge;

Here we're creating two variables called myName and myAge. Try typing these lines into your web browser's console. After that, try creating a variable (or two) with your own name choices.

myName;

myAge;

They currently have no value; they are empty containers. When you enter the variable names, you should get a value of undefined returned. If they don't exist, you'll get an error message — try typing in

[**Updating a variable**](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#updating_a_variable)

Once a variable has been initialized with a value, you can change (or update) that value by giving it a different value. Try entering the following lines into your console:

JSCopy to Clipboard

myName = "Bob";

myAge = 40;

variable types

### [Numbers](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#numbers)

You can store numbers in variables, either whole numbers like 30 (also called integers) or decimal numbers like 2.456 (also called floats or floating point numbers). You don't need to declare variable types in JavaScript, unlike some other programming languages. When you give a variable a number value, you don't include quotes:

### [Strings](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#strings)

Strings are pieces of text. When you give a variable a string value, you need to wrap it in single or double quote marks; otherwise, JavaScript tries to interpret it as another variable name.

### [Booleans](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#booleans)

Booleans are true/false values — they can have two values, true or false. These are generally used to test a condition, after which code is run as appropriate. So for example, a simple case would be:

### [Arrays](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#arrays)

An array is a single object that contains multiple values enclosed in square brackets and separated by commas. Try entering the following lines into your console:

### [Objects](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#objects)

In programming, an object is a structure of code that models a real-life object. You can have a simple object that represents a box and contains information about its width, length, and height, or you could have an object that represents a person, and contains data about their name, height, weight, what language they speak, how to say hello to them, and more.