Numbers, Strings & Booleans

Learn JavaScript's most basic variable types: numbers, strings, and booleans. Learn how to use and work with variables.

Types of Variables

There are a few different types of variables in JavaScript. We'll discuss **numbers**, **strings**, and **booleans** first.

Numbers

Numbers are just what you'd expect and what we've been using in the examples above. `19`, `30`, `924311933` are all numbers. We can't use commas, spaces, or other characters when we create numbers. We have to use continuous digits.

Strings

Strings are different. We can think of them as the way we represent text in JavaScript. They're created using the `'` and `"` characters. For example, if we wanted to store a representation of the word `Hello` in JavaScript, we could write it two different ways.

```
let string = 'Hello';
let string2 = "Hello";
```

Deciding whether to use `'` or `"` is based on personal preference. We just can't mix them up when creating a string.

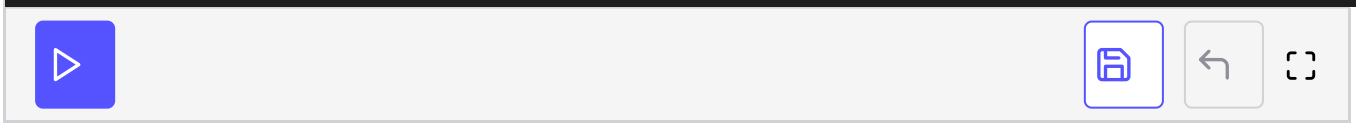
In other words, we have to end a string with the same character (`'` or `"`) that we used to start it. We can write `'Hello'` or `"Hello"` but not `'Hello"` or `"Hello'`.

String Characters

Strings don't have to be words. We can put any typeable keyboard character into a string. We can also put pretty much any text you can find on the internet into a string. Even emojis.

```
let lettersAndNumbers = 'abcd1234';
let symbols = '!@#$%^&*()_+|}{';
let emojis = '😀 🦋 🦏 🌵 🍄 🌍 🏵';

console.log(lettersAndNumbers);
console.log(symbols);
console.log(emojis);
```



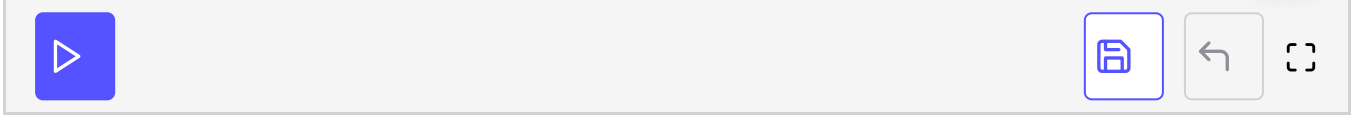
Escaping Characters

What happens if we want to use the `'` or `"` characters in our strings? We can *escape* them. This means that JavaScript will interpret them as normal string characters and won't end the string.

To escape a character, we put a backslash `\` in front of it. The string will then be interpreted correctly.

The following line prints: "We're learning about escaping characters!"

```
console.log('We\'re learning about escaping characters!');
```



Booleans

A third basic type of variable is called a boolean. Booleans contain 1 of 2 possible values: `true` or `false`.

```
let booleanTrue = true;
let booleanFalse = false;
```

`true` and `false` are reserved keywords in JavaScript. Because of this, they

can't be used as variable names.

Naming Variables

It might be tempting to use single-letter variable names such as `x` and `y`. It saves typing time, so why not?

We always want to give our variables descriptive names. One of the hallmarks of good code is that it is easy to read.

If someone else is reading your code, they'll have no idea what variables named `x` or `y` mean. It'll make much more sense if instead, we used `name` and `password` as our variable names.

It also helps us write the code. Often, code logic gets quite confusing and poorly named variables can make it impossible to read our own code. Descriptive variable names are vital.

Quiz

Feel free to test your understanding.

Types

1 What is the type of this variable?

```
let variable = 0;
```

- A) number
- B) string
- C) boolean

2 What is the type of this variable?

```
let variable = '17';
```

- A) number
- B) string
- C) boolean

3 What is the type of this variable?

```
let variable = 'true';
```

- A) number
- B) string
- C) boolean

4 What is the type of this variable?

```
let variable = false;
```

A) number

B) string

C) boolean

CHECK ANSWERS