JS VARIABLES COURSE

APRIL 2022

YONATAN BENEZRA

What are Variables?

Variables are containers for storing data (storing data values).
If we simplify it, variables hold values, as follows:

```
let x = 13;
let y = 5;
```

Variables are used in expressions as well:

```
let z = x + y;
```

From the example above, you can conclude that z is calculated to be 18.
Variables can store all sorts of data, not just numbers.

For example:

```
let firstName = ""
let lastName = ""
firstName = "Yonatan"
lastName = "Benezra"
console.log(firstName + lastName)
```

What will the above example console?

Do you know how to fix it?

There are 3 types of variables:

- Const
- Let
- Var

Our default will be to use the “Const”, then “Let”.
If necessary, we will use “var” as well. Let’s discuss when each of the above types might be necessary.

When Should JavaScript's "var" Be Used?

The "var" keyword is used in all JavaScript code from 1995 to 2015.
The "let" and "const" keywords were added to JavaScript in 2015.
If you want your code to run in an older browser, you must use only var.

When Should JavaScript's "const" and "let" Be Used?

A general rule that can always be followed: Always declare variables with **const.**
If you think the value of the variable might change later on, use **let.**

In the below example, x, y, and total, are variables:

```
const x = 11;
const y = 4;
let total = x + y;
```

The two variables x and y are declared with the const keyword.
These are constant values and cannot be changed.
The variable total is declared with the let keyword.
This is a value that can be changed.

JavaScript Identifiers

All JavaScript variables must be identified with unique names.
These unique names are called identifiers.
Identifiers can be short names (like x and y), or more descriptive names (such as age, total, or fullName).

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter
- Names can also begin with $ and _ (mostly for special cases such as private variables - when we want everyone who is looking at our code to understand that this variable is private to this function)
- Names are case sensitive (y and Y are different variables)
- Reserved words (like JavaScript keywords [const, let …]) cannot be used as names

***Note:**
Identifiers are case sensitive

The Assignment Operator

In JavaScript, the equal sign **(=)** is an "assignment" operator, not an "equal to" operator.

To check if one variable is equal to another, we need to write “===”.
As we saw in earlier examples, we assigned a value to a variable using “=”.

You can even assign the variable to itself, as follows:

```
x = x + 1
```

Exercise:

Find a different syntax that can be used to write the above expression (use Google).
Need a hint? try searching the following:

x = x + 1 js alternative