Variables and Constants

var, let, and const.

FSDI 103 - Programming Fundamentals

You would learn…

How to create “containers” to store and manage data that will help you to define the behavior of your code.

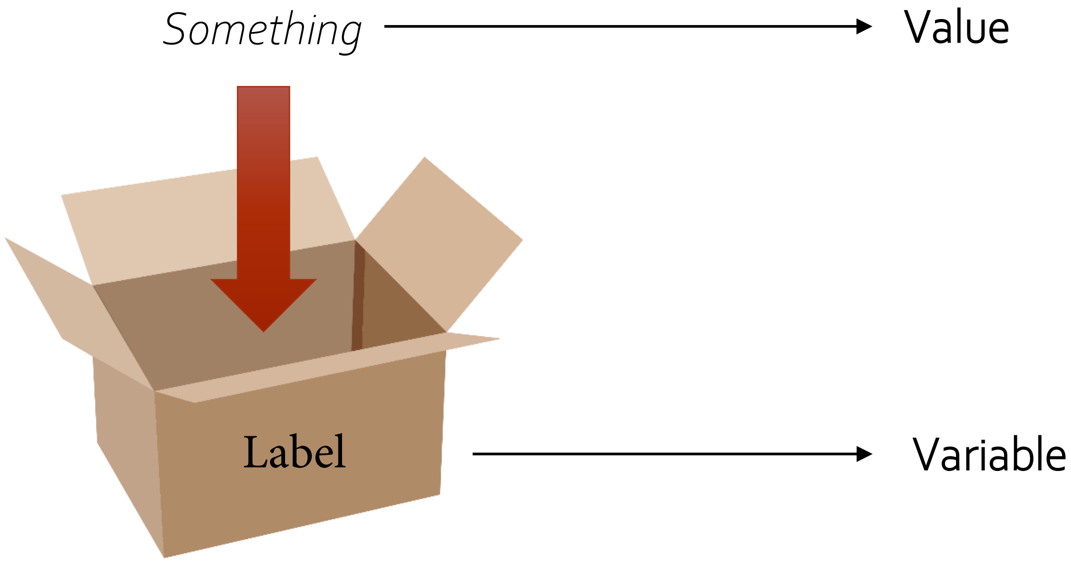
Before we start

Tell me what have you heard about VARIABLE
or what do you think VARIABLES are.

Google says that, in programming, a variable is:

A **value** that can **change**, **depending on** conditions or on information passed to the program.

Understanding the concept



Something → Value

Label → Variable

Change

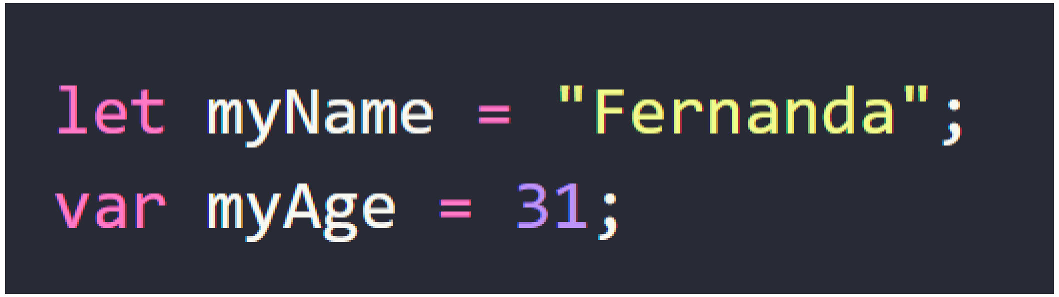
Variables in JS

Keyword: let

Structure of a variable declaration

keyword variableName = value ;

Example:



```
let myName = "Fernanda";
var myAge = 31;
```

Some data types of variables

- String: A sequence of characters (text).
- Numerical: Can be a integer or a floating-point number.
- Boolean: Data type with only two possible values: *true or false.*

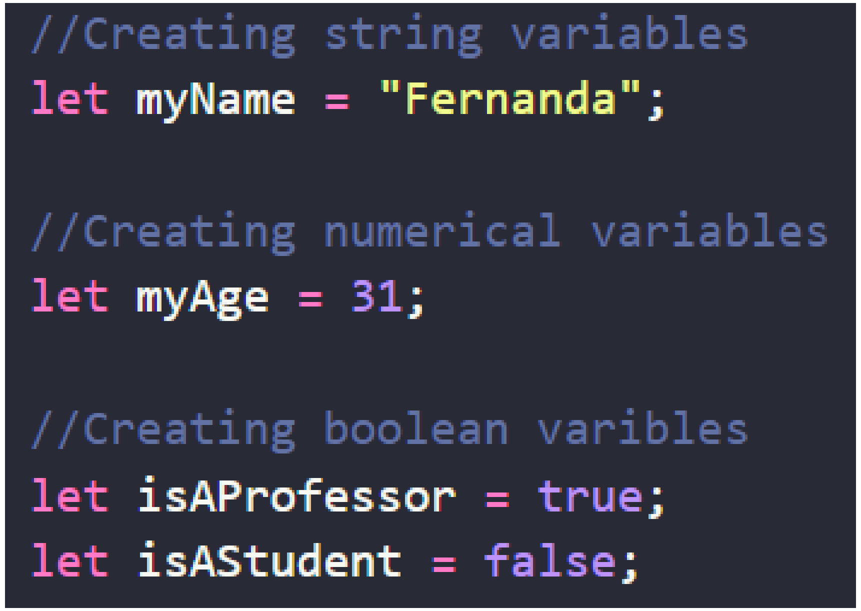
Let’s practice

Go to VS Code

Some data types of variables

- **String:** A sequence of characters (text).
- **Numerical:** Can be a integer or a floating-point number.
- **Boolean:** Data type with only two possible values: *true or false.*

Example:



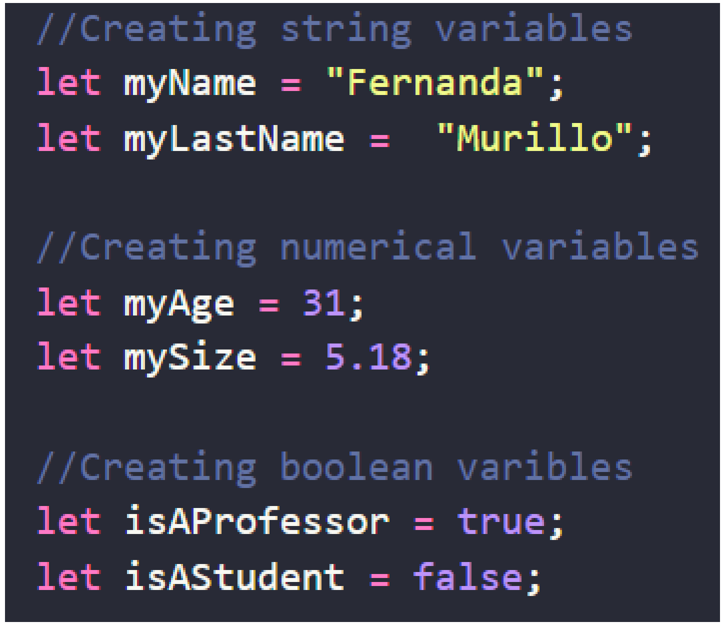
```
//Creating string variables
let myName = "Fernanda";

//Creating numerical variables
let myAge = 31;

//Creating boolean varibles
let isAProfessor = true;
let isAStudent = false;
```

Exercise 1

1. Create a variable to save your last name.
2. Create a variable to save your size.

Result

JS file:



```
//Creating string variables
let myName = "Fernanda";
let myLastName =  "Murillo";

//Creating numerical variables
let myAge = 31;
let mySize = 5.18;

//Creating boolean varibles
let isAProfessor = true;
let isAStudent = false;
```

3 Things we can do with variables

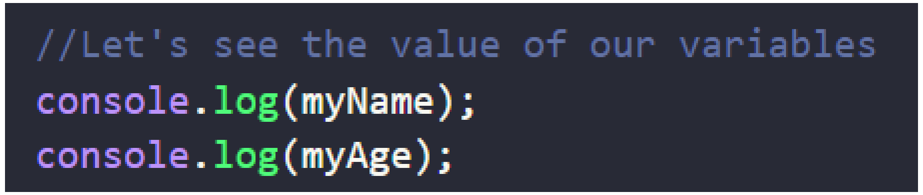
1. See the value that is stored.
2. Concatenate.
3. Operations.

3 Things we can do with variables

1. See the value that is stored.
2. Concatenate.
3. Operations.

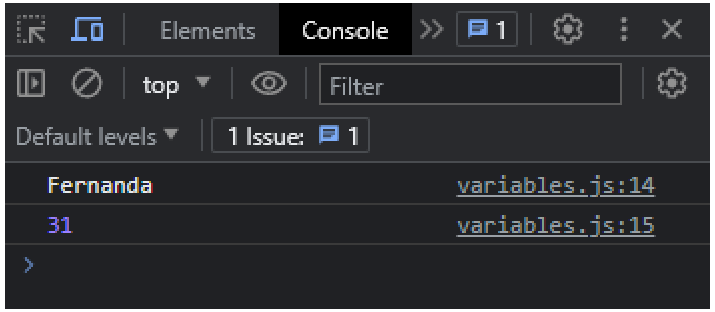
See values in console

JS file:



```
//Let's see the value of our variables
console.log(myName);
console.log(myAge);
```

Browser console:



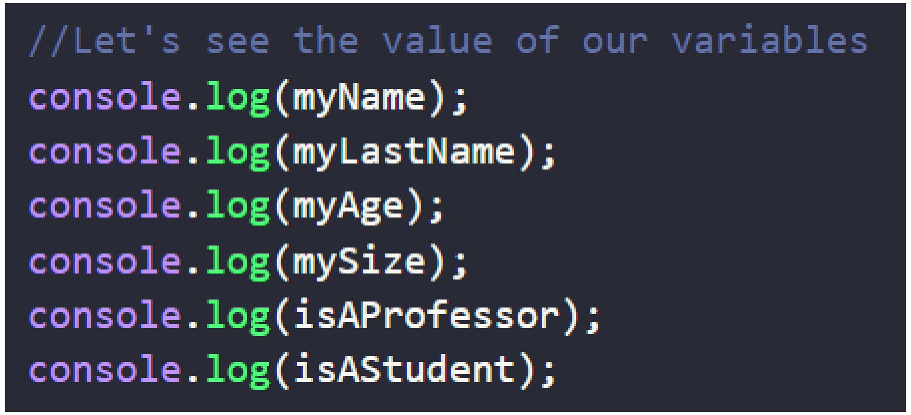
```
Fernanda                              variables.js:14
31                                    variables.js:15
```

Exercise 2

- Show in the console the value of all the variables.

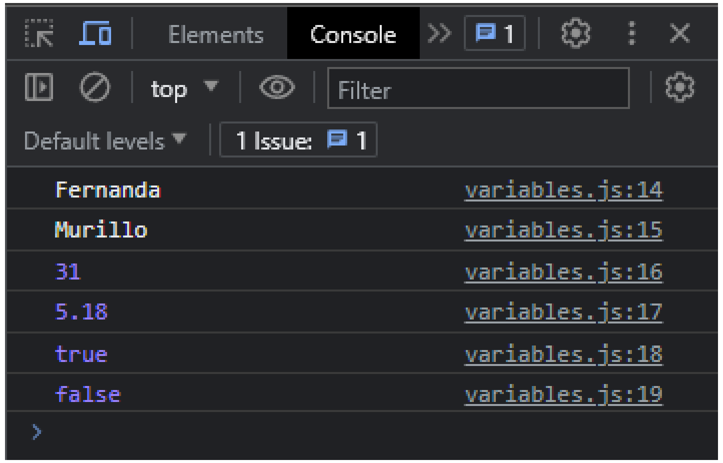
Result

JS file:



```
//Let's see the value of our variables
console.log(myName);
console.log(myLastName);
console.log(myAge);
console.log(mySize);
console.log(isAProfessor);
console.log(isAStudent);
```

Browser console:



```
Fernanda                    variables.js:14
Murillo                     variables.js:15
31                          variables.js:16
5.18                        variables.js:17
true                        variables.js:18
false                       variables.js:19
>
```

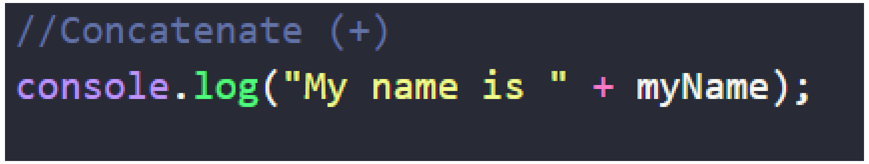
3 Things we can do with variables

1. See the value that is stored.
2. Concatenate.
3. Operations.

Concatenation

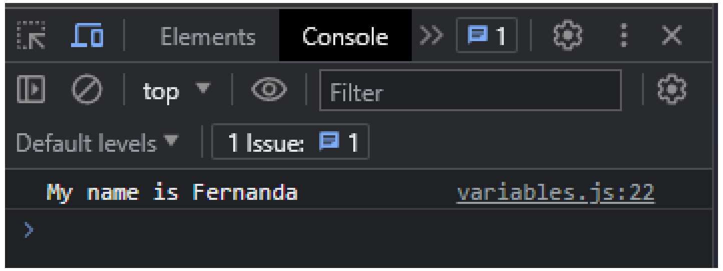
Join variables value to create a sentence.

JS file:



```
//Concatenate (+)
console.log("My name is " + myName);
```

Browser console:



```
My name is Fernanda                variables.js:22
>
```

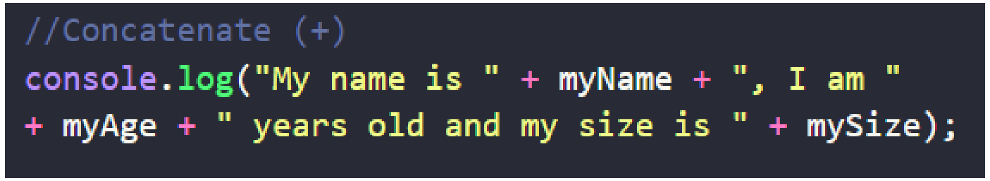
Exercise 3

- Use concatenation concept to show in console the following sentence:

  My name is *your name*, I am *your age* years old and my size is *your size*.

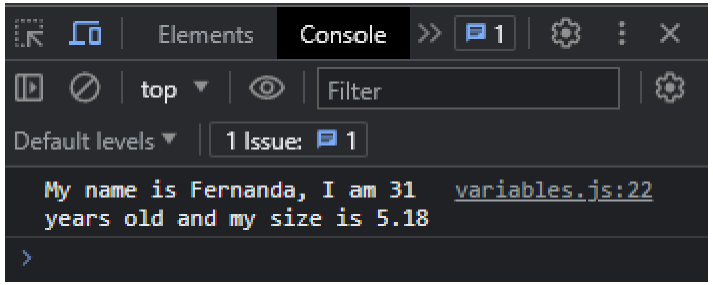
Result

JS file:



```
//Concatenate (+)
console.log("My name is " + myName + ", I am "
+ myAge + " years old and my size is " + mySize);
```

Browser console:



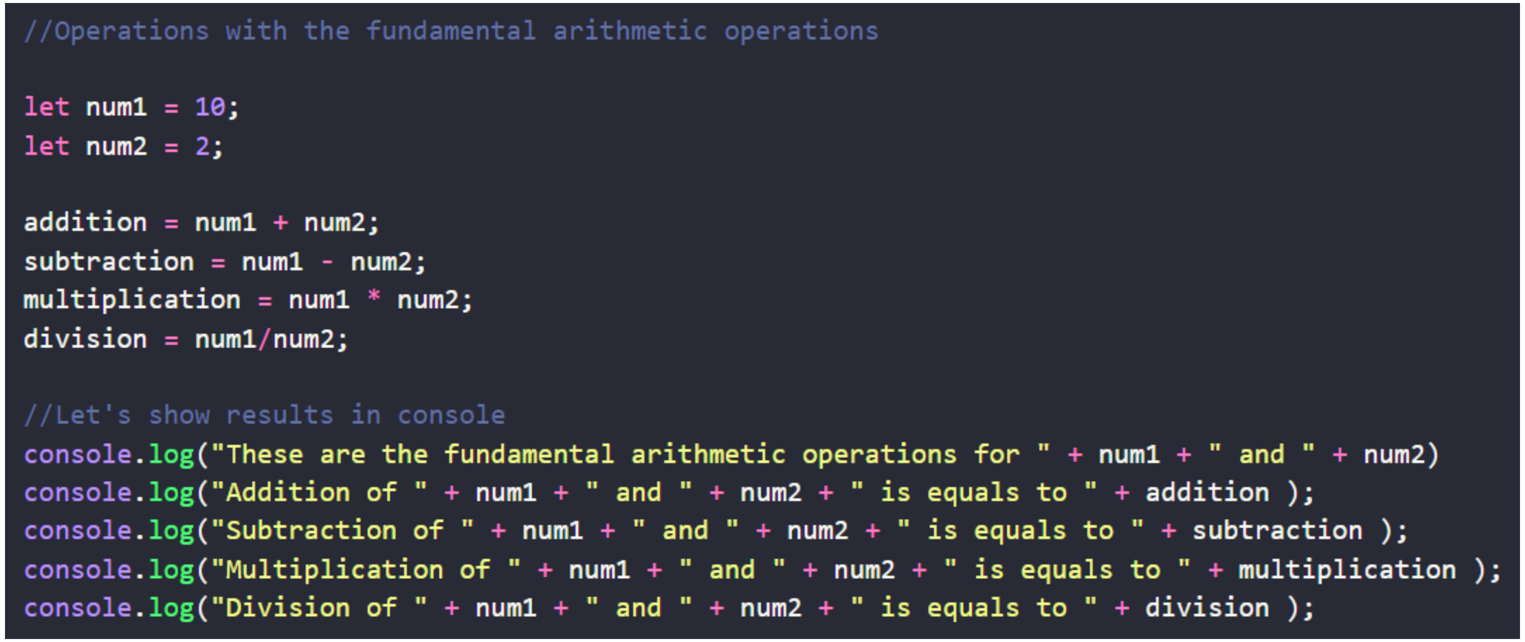
```
My name is Fernanda, I am 31
years old and my size is 5.18
```

3 Things we can do with variables

1. See the value that is stored.
2. Concatenate.
3. Operations.

Perform operations

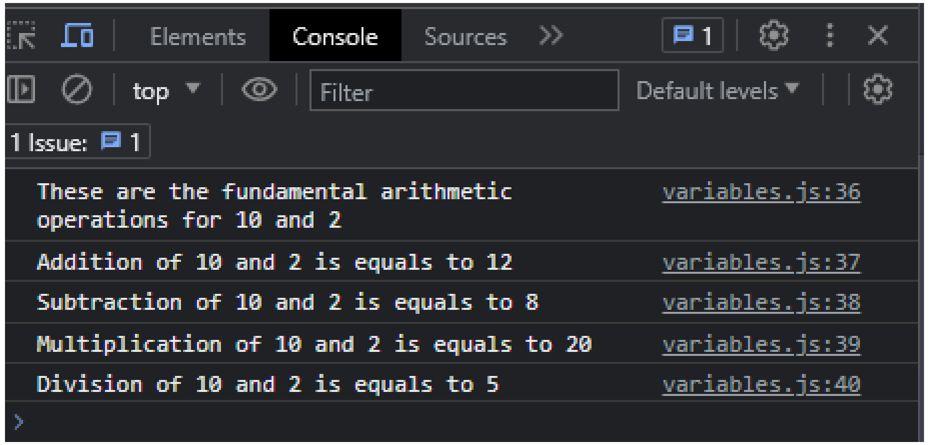
JS file:



```
//Operations with the fundamental arithmetic operations

let num1 = 10;
let num2 = 2;

addition = num1 + num2;
subtraction = num1 - num2;
multiplication = num1 * num2;
division = num1/num2;

//Let's show results in console
console.log("These are the fundamental arithmetic operations for " + num1 + " and " + num2)
console.log("Addition of " + num1 + " and " + num2 + " is equals to " + addition );
console.log("Subtraction of " + num1 + " and " + num2 + " is equals to " + subtraction );
console.log("Multiplication of " + num1 + " and " + num2 + " is equals to " + multiplication );
console.log("Division of " + num1 + " and " + num2 + " is equals to " + division );
```

Perform operations

Browser console:



```
These are the fundamental arithmetic        variables.js:36
operations for 10 and 2
Addition of 10 and 2 is equals to 12        variables.js:37
Subtraction of 10 and 2 is equals to 8      variables.js:38
Multiplication of 10 and 2 is equals to 20  variables.js:39
Division of 10 and 2 is equals to 5         variables.js:40
>
```

$$a = \pi \ * r^2$$

Challenge 1

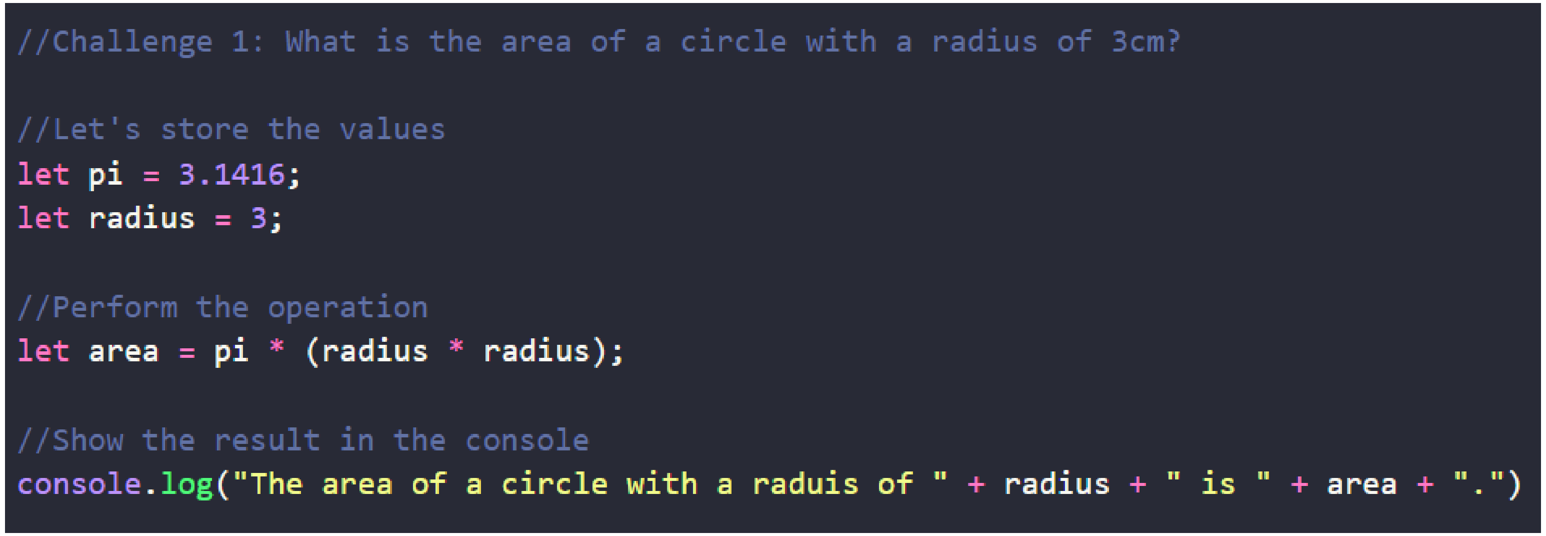
What is the area of a circle with a radius of 3cm?

1. Create the variables to store the necessary values.
2. Perform the operations.
3. Show the result in the console as the following sentences:

The area of a circle with a radius of radius value is the area value.

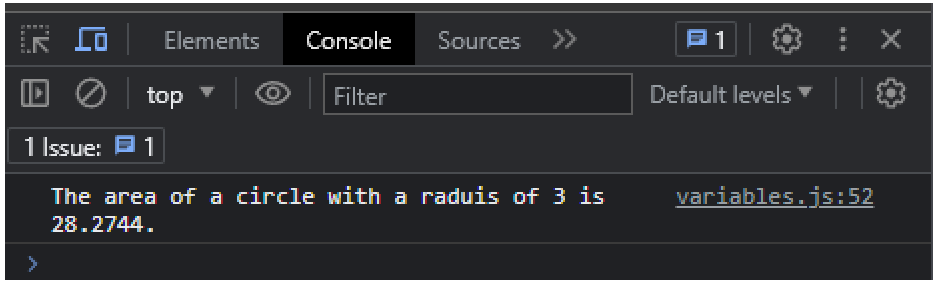
Result

JS file:



```
//Challenge 1: What is the area of a circle with a radius of 3cm?

//Let's store the values
let pi = 3.1416;
let radius = 3;

//Perform the operation
let area = pi * (radius * radius);

//Show the result in the console
console.log("The area of a circle with a raduis of " + radius + " is " + area + ".")
```

Console:



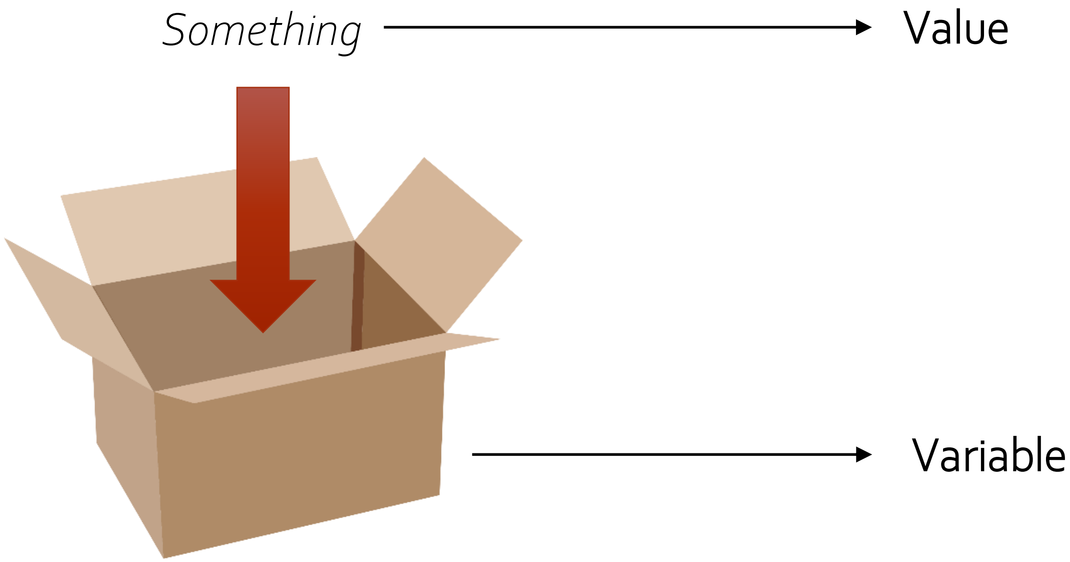
```
The area of a circle with a raduis of 3 is
28.2744.
```

Constants in JS

Keyword: const

Google says that, in programming, a constant is:

.. is a named data item with a **predefined** value.



Something → Value

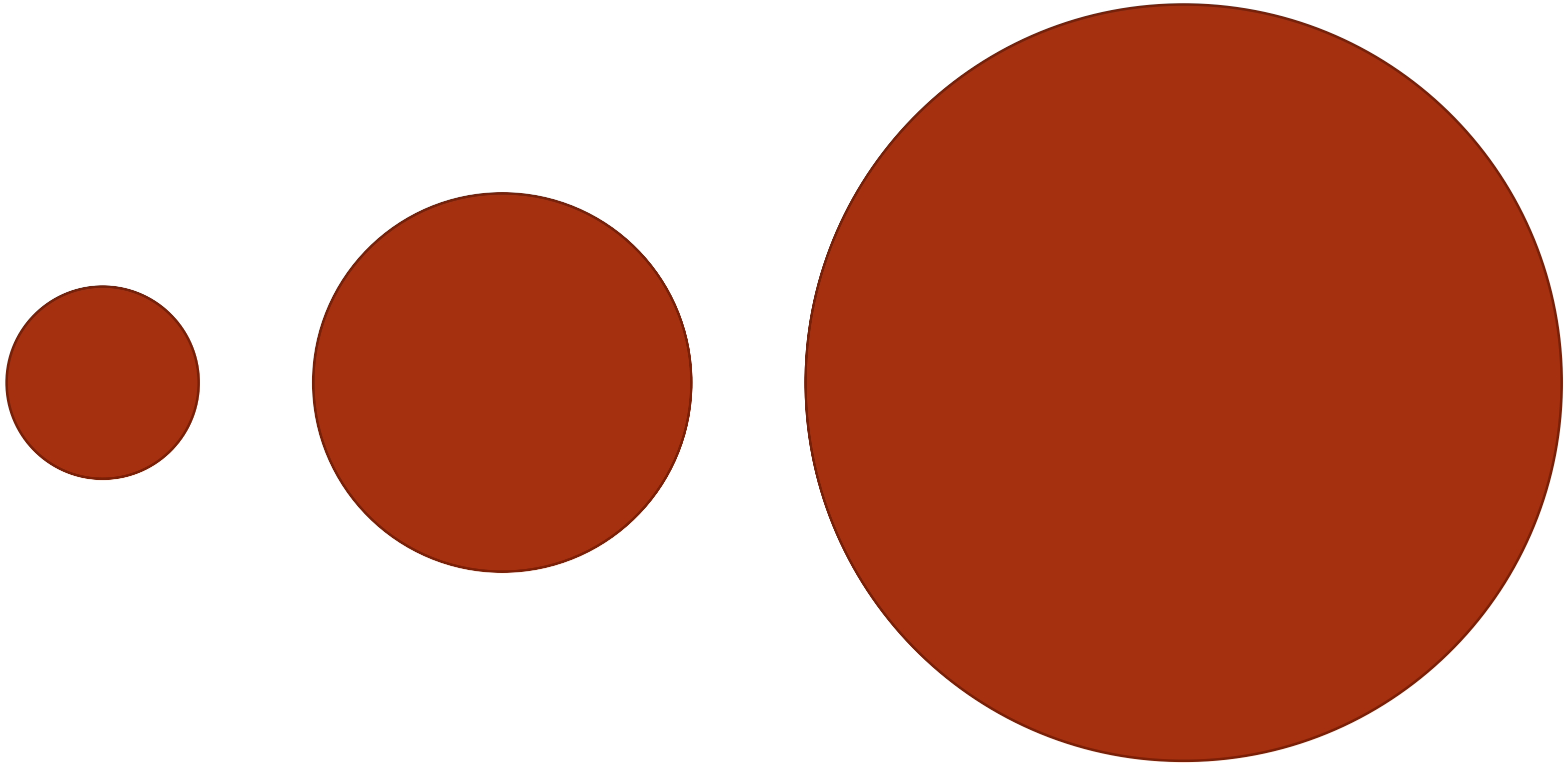
→ Variable

The value is
ALWAYS the same.

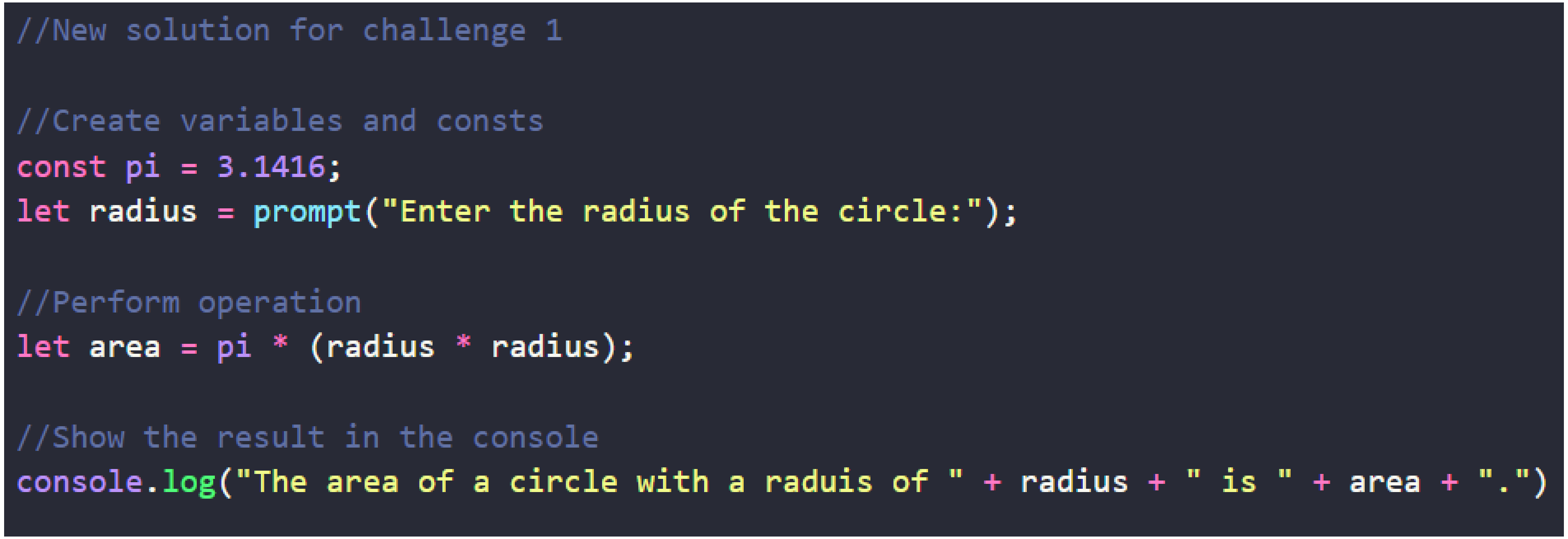
ALWAYS

In the equitation of the area of a circle…

pi value is a constant, but the area and the radius are variables.



Other solution for challenge 1



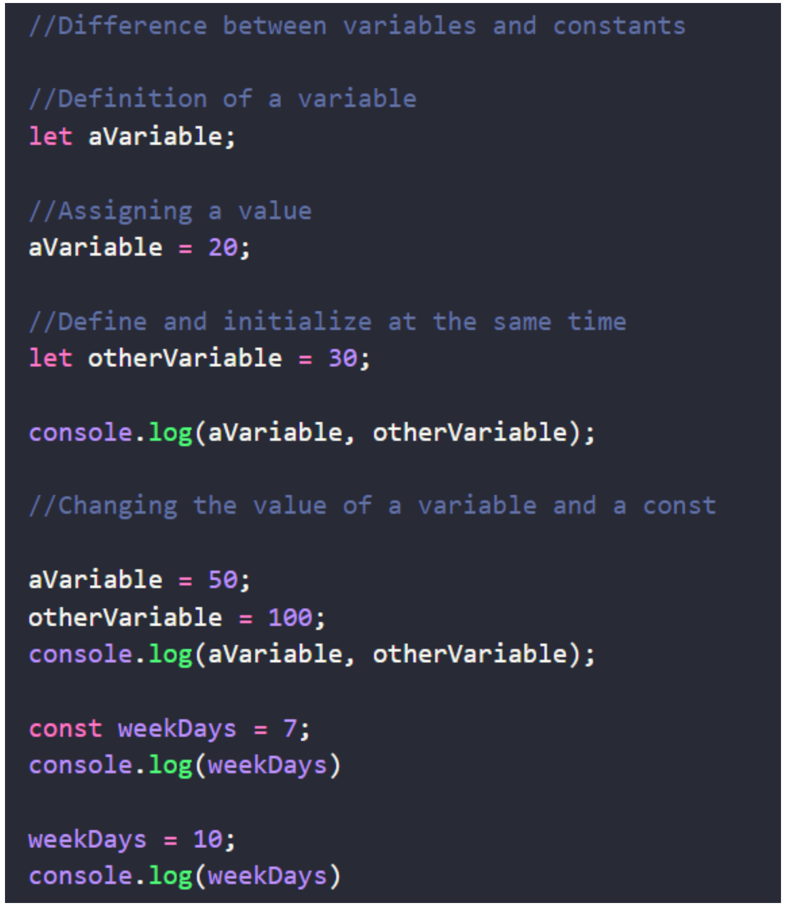
```
//New solution for challenge 1

//Create variables and consts
const pi = 3.1416;
let radius = prompt("Enter the radius of the circle:");

//Perform operation
let area = pi * (radius * radius);

//Show the result in the console
console.log("The area of a circle with a raduis of " + radius + " is " + area + ".")
```

Difference between variable and constant

Go to VS Code!

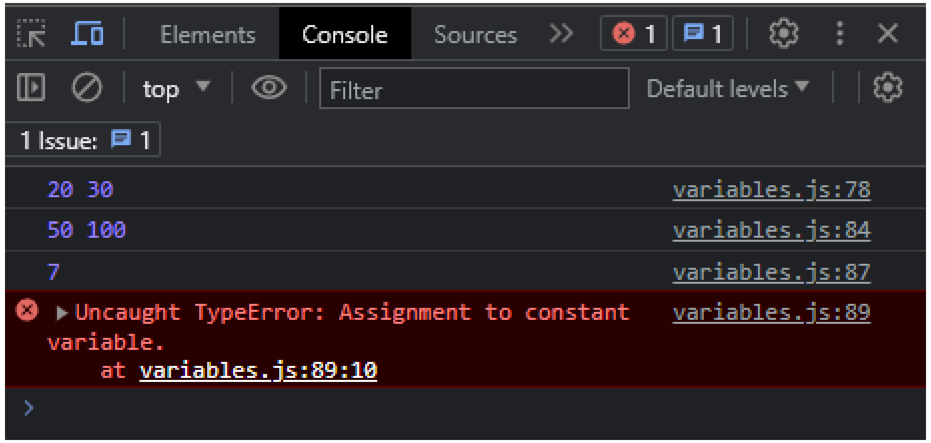
Constants do not change their value!

JS file:



```
//Difference between variables and constants

//Definition of a variable
let aVariable;

//Assigning a value
aVariable = 20;

//Define and initialize at the same time
let otherVariable = 30;

console.log(aVariable, otherVariable);

//Changing the value of a variable and a const

aVariable = 50;
otherVariable = 100;
console.log(aVariable, otherVariable);

const weekDays = 7;
console.log(weekDays)

weekDays = 10;
console.log(weekDays)
```

Console:



```
20 30                                                    variables.js:78
50 100                                                   variables.js:84
7                                                        variables.js:87
▶Uncaught TypeError: Assignment to constant              variables.js:89
variable.
    at variables.js:89:10
>
```

What about *var*?

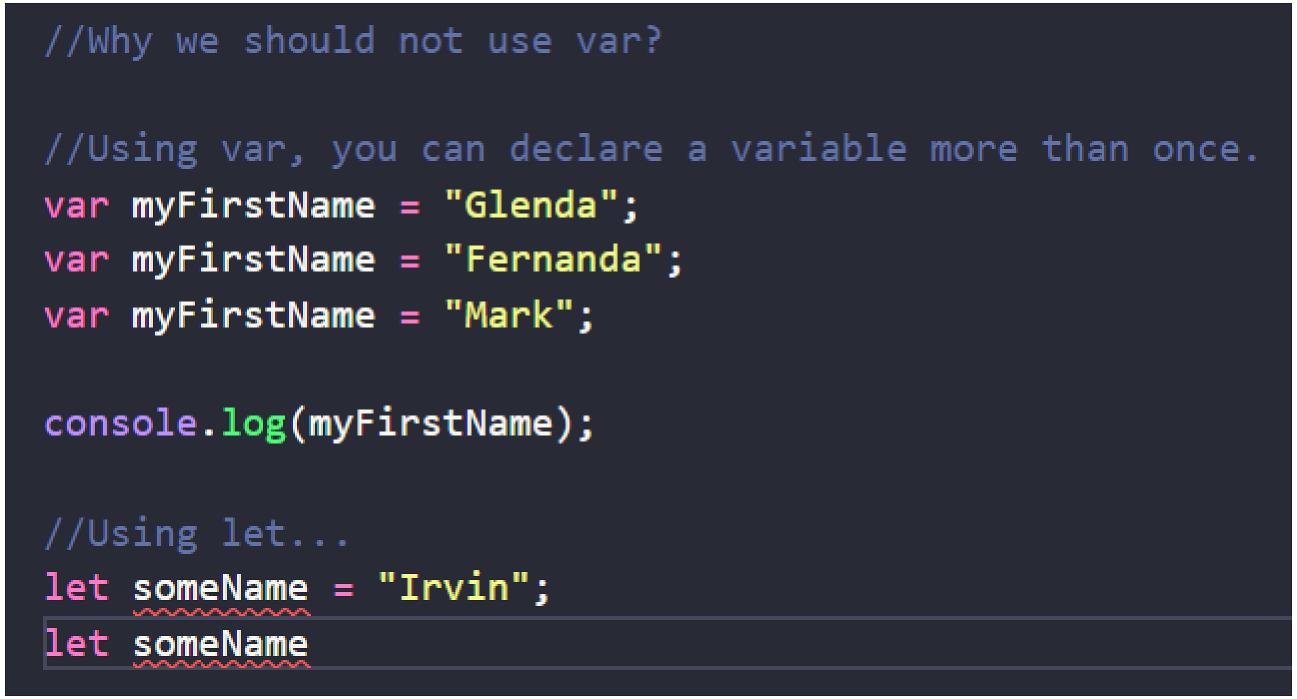
var
let
const

Developers’ mozilla documentation says:

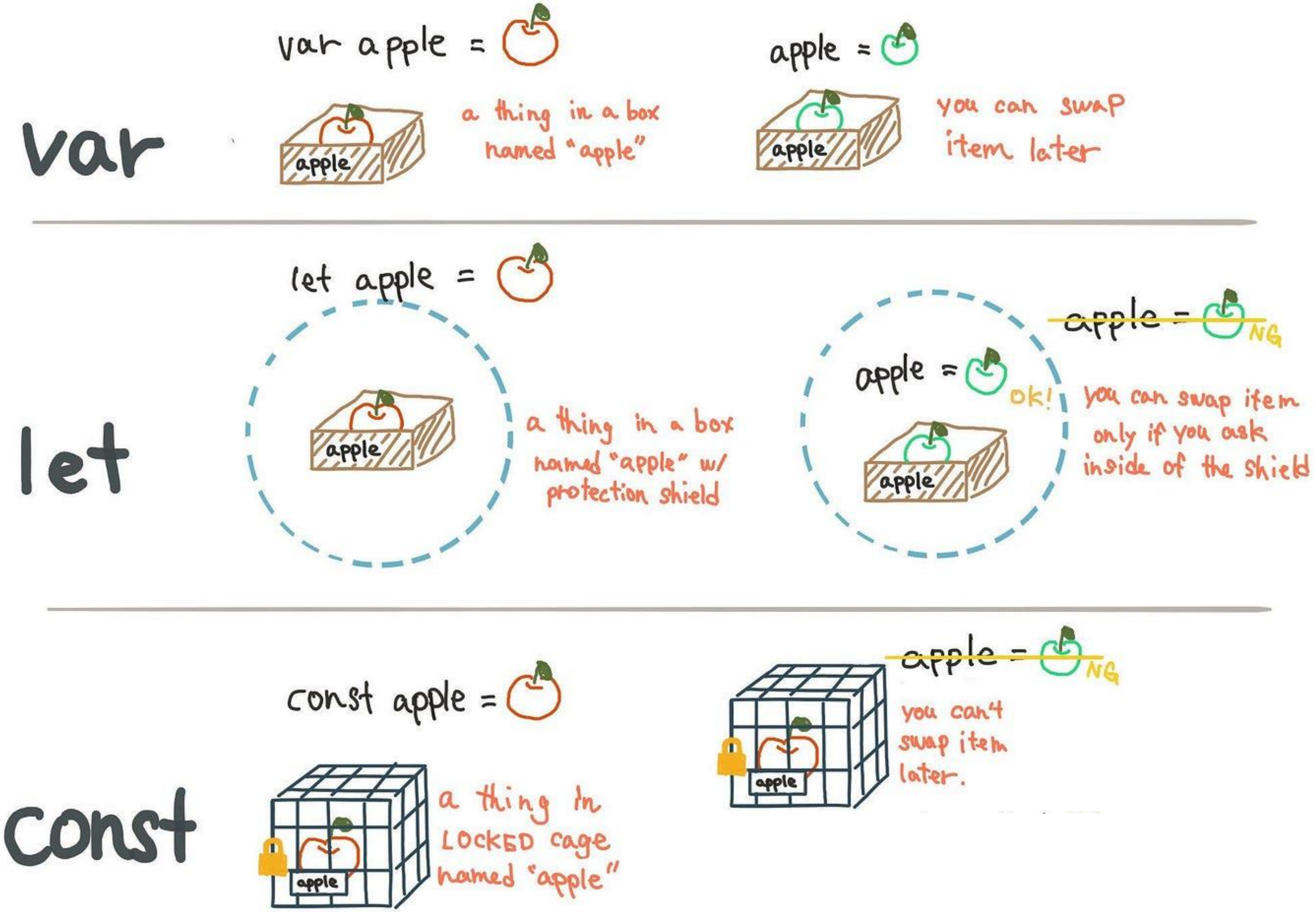
“Unless you are explicitly writing support for ancient browsers, there is no longer any reason to use var as all modern browsers have supported let since 2015”.

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#a_note_about_var

But why? ***To avoid logic errors!***



```
//Why we should not use var?

//Using var, you can declare a variable more than once.
var myFirstName = "Glenda";
var myFirstName = "Fernanda";
var myFirstName = "Mark";

console.log(myFirstName);

//Using let...
let someName = "Irvin";
let someName
```


var
var apple =
apple
a thing in a box named "apple"
apple =
apple
you can swap item later
let
let apple =
apple
a thing in a box named "apple" w/ protection shield
~~apple =~~ NG
apple =
ok!
apple
you can swap item only if you ask inside of the shield
const
const apple =
apple
a thing in LOCKED cage named "apple"
~~apple =~~ NG
apple
you can't swap item later.

So *var* makes logical error tracking much more complex.

Next session we will se more examples

Thank you!