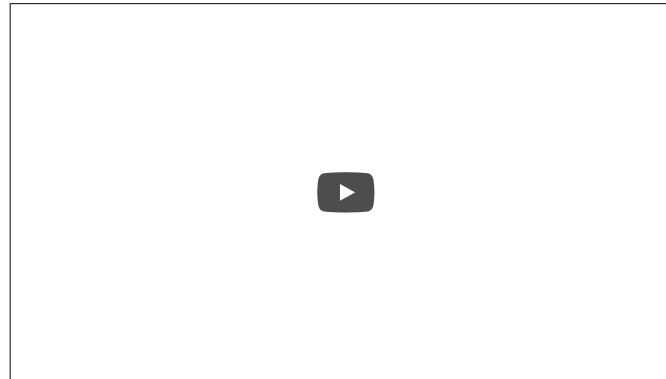




- ✓ 1. Intro to Data Types
- ✓ 2. Numbers
- ✓ 3. Comments
- ✓ 4. Quiz: First Expression (2-1)
- ✓ 5. Strings
- ✓ 6. String Concatenation
- ✓ 7. Variables
- 8. Quiz: Converting Temperatures (2-2)
- 9. String Index
- 10. Escaping Strings
- 11. Comparing Strings
- 12. Quiz: Favorite Food (2-3)
- 13. Quiz: String Equality for All (2-4)
- 14. Quiz: All Tied Up (2-5)
- 15. Quiz: Yosa Buson (2-6)
- 16. Booleans
- 17. Quiz: Facebook Post (2-7)
- 18. Null, Undefined, and NaN
- 19. Equality
- 20. Quiz: Semicolons! (2-8)
- 21. Quiz: What's my Name? (2-9)
- 22. Quiz: Out to Dinner (2-10)
- 23. Quiz: Mad Libs (2-11)
- 24. Quiz: One Awesome Message (2-12)
- 25. Lesson 2 Summary

Variables



With variables, you no longer need to work with one-time-use data.

At the beginning of this course, you declared the value of a string, but you didn't have a way to access or reuse the string later.

```
"Hello"; // Here's a String "Hello"
"Hello" + " World"; // Here's a new String (also with the value "Hello") concatenated with " World"
```

Storing the value of a string in a variable is like packing it away for later use.

```
var greeting = "Hello";
```

Now, if you want to use "Hello" in a variety of sentences, you don't need to duplicate "Hello" strings. You can just reuse the `greeting` variable.

```
greeting + " World!";
```

Returns: Hello World!

```
greeting + " Mike!";
```

Returns: Hello Mikel

You can also change the start of the greeting by *reassigning* a new string value to the variable `greeting`.

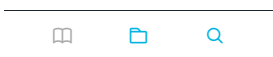
```
greeting = "Hola";
greeting + " World!";
```

Returns: Hola World!

```
greeting + " Mike!";
```

Returns: Hola Mike!

Naming conventions



- ✓ 1. Intro to Data Types
- ✓ 2. Numbers
- ✓ 3. Comments
- ✓ 4. Quiz: First Expression (2-1)
- ✓ 5. Strings
- ✓ 6. String Concatenation
- ✓ 7. Variables
- 8. Quiz: Converting Temperatures (2-2)
- 9. String Index
- 10. Escaping Strings
- 11. Comparing Strings
- 12. Quiz: Favorite Food (2-3)
- 13. Quiz: String Equality for All (2-4)
- 14. Quiz: All Tied Up (2-5)
- 15. Quiz: Yosa Buson (2-6)
- 16. Booleans
- 17. Quiz: Facebook Post (2-7)
- 18. Null, Undefined, and NaN
- 19. Equality
- 20. Quiz: Semicolons! (2-8)
- 21. Quiz: What's my Name? (2-9)
- 22. Quiz: Out to Dinner (2-10)
- 23. Quiz: Mad Libs (2-11)
- 24. Quiz: One Awesome Message (2-12)
- 25. Lesson 2 Summary

```
var totalAfterTax = 53.03; // uses camelCase if the variable name is multiple words
var tip = 8; // uses lowercase if the variable name is one word
```

Not using camelCase for your variables names is not going to necessarily *break* anything in JavaScript. But there are recommended style guides used in all programming languages that help keep code consistent, clean, and easy-to-read. This is especially important when working on larger projects that will be accessed by multiple developers.

You can read more about Google's JavaScript StyleGuide [here](#).

QUIZ QUESTION

Which of these are good variable names?

☐ var thingy = 1;

☒ var count = 1;

☒ var postLiked = false;

☐ var firstname = "Richard";

SUBMIT

NEXT