

Computational Thinking

Programming Fundamentals with JavaScript

- ▶

Variables

9 min
- ▶

Methods

7 min
- ▶

Functions

5 min

▶

Types

4 min

- ▶

DukeLearnToProgram Environment

10 min
- 📄

Try It! Using Variables, Methods and Functions

30 min
- ▶

For Loops

6 min
- 📄

Try It! Using For Loops

30 min
- ▶

Conditional Execution

7 min
- 📄

Programming Exercise: Modifying Images

1h 30m
- ★

Practice Quiz: Modifying Images with JavaScript

8 questions

Implementing the Green Screen Algorithm

Review



Consider the following code.

```
1  var x = 17;  
2  var y = 7;  
3  
4  var a = "hello";  
5  var b = "world";  
6  
7  var z = x+y;  
8  var c = a+b;
```

After running this code, what are the values of **z** and **c**?

☒ **z** is 24, **c** is "helloworld"

Correct

☐ **z** is 24, **c** is "worldhello"

☐ **z** is "helloworld", **c** is 24

Continue

▶ 2:54 / 4:20

⋮ 🔊 ⚙️ ↗️

Types

👍

👎

🚩

Have a question? Discuss this lecture in the week forums.



Interactive Transcript

Search Transcript

English ▾

0:03

Welcome back. Now you've seen a better code that works with images and numbers, two different types of data. But what if we do something that does not make sense? Like try to do w.getWidth? w is the number for 80 which doesn't have a getWidth method In this particular case, the code will crash when execution reaches this line.

0:24

Why is this call to get width fine, but this one crashes the program?

0:30

If we go back to our diagram that depicts the state of the program, you will see that we were working with two different types of data.

0:39

fgImage refers to a simple image. Which has a getWidth method.

0:44

But w is a number, a type of data that does not have a getWidth method.

0:50

So what exactly is a type? Well, remember that everything is a number. But we have lots of different things we want those numbers to represent. Letters, images, sounds, and just plain numbers. So the type tells us what those numbers mean. That is, what kind of thing the number represents. This is important because the type specifies not only how the program should interpret the number, that is what the numeric value means. But also how to operate on it. The same operation may have different behavior on different types. This may sound confusing so let's look at an example. Here we have some code which has two plus operations in it. As you are about to see, these two pluses will do different things since they will operate on different types of data. Let's step through the code. n1 = 26, n2 = 16, s1 = "a". Here, s1 is a string, a sequence of letters. This particular sequence has just one letter, A. s2 = "b".

2:05

Now, we're about to do the first plus operation. Here we are doing N1 which is 26 plus N2 which is 16. Both operands are numbers, so we are just doing numeric addition. 26 plus 16 is 42. Now we are about to do the second plus operation. However, here we are doing plus on two strings, a and b.

2:29

What does it mean to do plus on strings. For strings, plus doesn't mean numeric addition it means concatenation. Concatenation is a fancy word for taking two strings and sticking them together one after the other. So you get s3 equals the string ab.

2:48

Notice how plus meant two different things based on the type of the data it [was operating on](#).

2:55

So how does JavaScript keep track of the type of the data.

2:59

It keeps information about the type, explicitly along with each value when it stores it in the computer's memory. Then whenever it needs to work with a value, it reads the type and value out of the memory to figure out what to do.

3:15

Because JavaScript keeps the types of all values in memory, meaning that it tracks and works with the types while it runs the program, It's called a dynamically typed language.

3:28

If you stay for course two and beyond in which you will learn Java types will work differently.

3:35

Java is a statically typed language which means that ever variable and every expression has exactly one type. Which must be known before the program can even be run.

3:47

For this to work in Java, you, the programmer, write down the types of your variables explicitly in your code. Then your code can be checked for certain kinds of mistakes, like trying to call getWidth on a number before it is even run, having this kind o mistake found before the program runs is really helpful. You'll learn more about types and Java in course 2. For now however, we have a bit more JavaScript to do so that you can implement the green screen algorithm. Thank you.

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt
Lecture Slides	pdf

Would you like to [help us translate](#) the transcript and subtitles into additional languages?