

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
Try It! Using Conditional Execution	30 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 = 2;  
2  var y = 5;  
3  y = y*y;  
4  x = x+y;
```

What will be the value of x after this code has been executed?

☐ 9

☐ 2

☐ 7

☐ 29

☒ 27

Correct

Continue

6:17 / 9:41

update y's box to be 8.

Variables

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

Interactive Transcript

Search Transcript

English

0:03
Now that you have the algorithm for green screening an image, you want to translate it into code. But you have a lot of different pieces of code that you need to learn. We're going to learn these in JavaScript now, but the same basic concepts exist in pretty much any programming language. We'll switch to Java in the next course, and you'll see that much of this looks the same.

0:24
One thing you need is a way to give names to the values your algorithm comes up with. In the English description, we refer to them as fgImage, bgImage, output and currentPixel. These are going to turn into variables in your code, which you will learn about momentarily.

0:41
You also need a way to make images, whether reading in an existing image or creating a blank image. You need a way to operate on this images and their pixels looking at particular pixels see what color they are and setting the pixels in an image, you'll learn about this soon. Your algorithm calls for you to repeat some steps for each pixels in an image. You're going to need to learn how to write code that repeats steps which will be a for loop which you will learn about later. And finally, your algorithm looks at a condition, in this particular case whether a pixel is green or not, and makes a decision about what to do next based on that condition. This is going to turn into an if else statement which you will also learn about soon.

1:25
Here is an example of a JavaScript statement which declares a variable called x and initializes it to 3. Declaring a variable means telling JavaScript that you want it to make a new variable. Initializing a variable means giving it the first value it should hold when you create it. Some languages do not require you to initialize variables, but it's a good idea to always do so. Let us take a minute to break down this syntax of this statement. First is the keyword var, this keyword is what tells JavaScript that you're declaring a new variable with this statement. JavaScript is case sensitive so you have to write var in all lowercase otherwise, it does not recognize it as this keyword.

2:09
Next is the name of the variable. In this case, we pick for the variable's name. You can name variables pretty much anything you want. There are some rules, but they're somewhat complicated. If you stick to letters, you will be in great shape.

2:23
While we can name your variable anything, it's good to name it descriptively. Here, we're not using the variable as part of the real algorithm, so we can't give it a really meaningful name. However, remember that we called the variables in our green screen algorithm things like fgImage, bgImage and output, which indicated what they were for. Next is an equal sign, this equal sign indicates that we are going to assign a value to the variable. After the equal sign is the value that we want to assign, in this case 3. Finally, there's a semicolon at the end of the statement. In JavaScript and many other programming languages, semicolons are used to the end of statements much like periods are used to the end of a sentence in English. You might wonder, why programming languages don't end statements with a period since that seems more natural? However, periods get used for a lot of other purposes such as decimal points in numbers which would make it hard for the computer to figure out if a period ends a statement or serves some other purpose.

3:26
Now that you have seen the syntax, let us take a look at the semantics. Recall that semantics are the meaning of something. In the case of a programming language, the semantics are the meaning of what the code does. What you're telling the computer to do. We're going to keep track of where we are with the green error shown here, just before the start of the code.

3:49
This arrow will typically point between statements indicating that we have finished the previous statement but not yet started the next statement. We will move it across a statement as we do the effect of that statement. Here, we are declaring a variable called x and initializing it to 3. So the effect of that statement, its semantics are to create a box, labelled x, and put 3 in as its value. Here is a slightly larger example, with 3 variables. The first you just saw, we create a box labeled x and put 3 in it. The next statement is pretty similar, except the variable is named y, and is initialized to 4. So to execute this statement we would create a box called y and put 4 in it. The final statement is a bit more complex. It declares a variable called z, but the right hand side of the equal sign has a more complex expression to compute initial value. $x + 2 * y$, expression is the technical term for a combination of values and operations to compute a value from them. Here, we have a mathematical expression involving the variables we previously declared, so we just do the math. To get the value for x, we look in its box and find that it, x is 3. Likewise, to get the value for y, we look in its box and find that y is 4. Now we just have to compute $3 + 2 * 4 = 11$. So we will create a box named z and initialize it with the value 11. You can also update the value of an existing variable with an assignment statement. Notice that the syntax is fairly similar to declaring a variable. Except that you do not write var since you are changing an existing variable rather than making a new one. Instead of creating a new box, you update the value in existing box and particular, the box named on the left side of the equal sign. Here we're going to do $x = z - 1$, so we would compute $z - 1$ which is 10 and update x's box to be 10. The next statement says $y = y * 2$. This looks like an equation from algebra class and sometimes novice programmers think that this statement would solve an algebraic equation. However that is not the case, this statement follows the same rules as any other assignment statement. The current value for y is 4, so we compute 4 times 2 which is 8 and update y's box to be 8. Okay, great, now you have seen the basics of variables with some numeric values, but what if you want variable that hold things that are not numbers? How could you do that? Well remember, everything is a number. So under the hood, every variable is actually a number, but you can use other data types like images and strings where the numbers are interpreted in other ways.

6:42
Let us look at this example, we have two statements here, each of which declares and initializes a variable. The first declares fgImage and initializes it to a new simple image which would have the effect of creating a box named fgImage, whose value is this image of me and Robert. You may notice that the image is not directly inside of the fgImage box, but rather off to the side with an arrow pointing at it. When you do new you end up with an arrow pointing at the thing that new made. Why does it matter that new makes a thing and that the box has an arrow pointing at it? As you will see soon, you can have multiple arrows pointing at the same thing from different boxes. The second statement behaves similarly except that the image which bgImage is initialized to is dinos.png. The expression that we're using to initialize these variables is a bit more complex than the mathematical expressions we saw earlier. So let us dive into them for a second. First, we have the keyword new. This tells JavaScript to go create a new object. What is an object? It is a piece of data which also has methods that can operate on it. In the case of an image, you'll be able to examine and modify its pixels with these methods, which you will learn about soon.

8:05
Next is the name of the type to create. Here, you want to make a simple image which is part of the Duke Learn to Program libraries and gives you a simple way of working with images to get started.

8:17
Finally, there are parameters in parenthesis. These parameters specify more information about how we want the object created. In the case of SimpleImage we can pass in a string which names the image file that we want it to load. This parameter is what tells the first line to load drewRobert.png and the second line to load dinos.png. There's a lot more to learn about objects and constructing them. You'll learn a little bit more about using objects shortly. However we won't go into too much detail about how to define your own types or objects called classes until later courses. We'll talk a bit more about construction in later courses too, although if you want much more detail on that topic, you would also want to take the follow on specialization from our friends at UCSD.

9:08
Okay, great, so now you know about variables which are boxes that hold values, and how to declare and initialize them as well as how to update them with assignment statements. You've learned that expressions are combinations of values such as variables and constants, as well as operations such as plus, minus, and times. And you learn about new simple image which let's you create a new image by loading the contents of an existing image file. This is the first pieces you need to understand to be ready to turn the green screen algorithm into code.

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?