

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



Modifying Images	1h 30m
Practice Quiz: Modifying Images with JavaScript	8 questions

Implementing the Green Screen Algorithm

Review

Making Decisions: if/else

```
if (x < y) {  
  z = 2;  
}  
else {  
  a = y + 1;  
  y = x - 3;  
}
```

The condition (in parentheses)

Break down syntax

written inside a parentheses.

Conditional Execution

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

Interactive Transcript

Search Transcript

English

0:03
Welcome back. Now that you know how to repeat steps with a for loop, the only piece remaining to implement the green screen algorithm is to be able to make decisions based on a condition.

0:15
In this algorithm we want it to look at a pixel, see if it's green and then do different steps based on whether it was or was not green. Now let's learn how to implement decision-making. Decision-making is done with if/else statements. Here you see an if/else statement, which looks at the condition, Is x less than y?, and it decides which of those statements to do based on whether that is true or false.

0:41
This example assumes that the variables are declared and initialized previously.

0:47
Let's break down the syntax. First you have the keyword if, which tells JavaScript that you are writing and if/else statement.

0:54
Next you have the condition to make a decision based on, here it is x less than y.

1:02
[This condition is an expression written inside a parentheses.](#) After the conditional expression, you have the "then clause". The "then clause" is the set of statements to execute if the condition is true.

1:16
These are written inside of curly braces. After the then clause is the keyword else. Then the "else clause". The "else clause" is a set of statements to execute if the condition is false. Again, these are written in curly braces. Sometimes you may see an if statement with no else clause, in which case, both the keyword else and the else clause are omitted. This is shorthand for an empty else clause, whenever the programmer does not want to do anything in the case of a false condition.

1:51
Okay, now that you have the basic syntax, let's look at the semantics. Here we have just reached this if statement with our variables having the values shown on the right.

2:03
The first thing that happens is we evaluate the conditional expression. Is x less than y? To evaluate this, we look at the values of these variables, the values they have in their boxes. And see that we are checking if 2 is less than 7. 2 is less than seven, so this expression evaluates to true. So since the condition is true, we go inside the then clause and then begin executing statements there. We execute the statements here according to their normal rules.

2:35
This next statement is an assignment statement, z = 2. So we update z's box to have 2. Now we have the closed curly brace of the then clause. We are done with the statements here. I need to skip over the else clause since those statements are only for when the condition is false. So we jump down past the end of the else clause. We would then keep executing whatever code comes next.

3:04
Now let's look at the example again, but with different values for our variables. We again begin by evaluating the condition expression. Is x less than y?

3:15
However, this time x is 42 and y is 7 so, we're evaluating is 42 less than 7. This expression evaluates the false since 42 is not less than 7. Since a conditional expression evaluate defaults, we jump into the else clause and begin executing statements there. And again, we execute statements according to their normal rules. First, a equals y plus one, which updates a to be 8. Then y equals x minus 3 which updates y to be 39. Now we are at the end of the else clause. So can just go pass the } of the else clause and begin executing whatever code comes next.

3:59
You have seen several important concepts separately. But it's instructive to see them all put together as in this example.

4:08
One important thing to remember when putting pieces together is that they generally follow the same rules, no matter how they are combined, a concept called compositionality.

4:22
Let's see what this code does. First, we declare a variable called img, and initialize it to a new SimpleImage. Which is the two by two image you saw in previous examples. Next there is a for loop which literates over each pixel in img.values. We've already seen how this works and we will again follow the same rules.

4:46
We create a box for the pixel variable, initialize it to the first pixel in our image, and go into the body of the loop. Now, we have an if statement. It does not matter what this if statement is inside of our loop. It still follows the same rules. Is 0 >= 2/2? No, that is false, so we go into the else { and execute statements there. This will set the pixel's blue to the pixel's red value, which is 255, changing it from red to magenta. Now we're at the end of the else clause. So we go outside of it, which brings us to the closed curly brace of the for loop. So we go to the next pixel. And back to the top of the loop. We again go into the loop body, but now the pixel x is 1. So we're asking is 1 greater than or equal to 2 divided by 2? That's true. So we go into the then clause and execute the statements there. Pixel.getRed divided by 2 is 0 divided by 2, which is 0. So we set the pixels red to zero which foes not change anything since it was already zero. We are now at the end of the then clause. We jump just past the close curly brace of the else clause. We have again reached the end of the four loop. So we go to the next pixel then jump back to the top. This pixels x is 0 so we are again evaluating is 0 greater than or equal to 2 divided by 2, which is false so we go into the else clause. We execute that statement and leave the else clause and go to the next pixel as we jump back to the top of the loop. Going back into the loop for the last pixel, we evaluate the condition which is true so we enter the then clause. We are going to set the red to 127.5, but pixel set red will round that to 127, since the pixel only works with integer RGB values. Now the pixel is light Cyan instead of white. We have reached the end of the then clause so we jump past the end of the outs clause which brings us to the end of the four loop. We go back to the top and there's no more pixels so we jump past the end of the loop and begin executing whatever code might be there. 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?