level: 1,  
chapter: 1,  
firstImage: **'https://s3-us-west-1.amazonaws.com/codr/L1I1.png'**,  
secondImage: **'https://s3-us-west-1.amazonaws.com/codr/L1I2.png'**,  
challengeText: **abcd apples**,  
instructionText: **Fill in the text below so that the box is labeled**,  
learnText:

**To start coding, we need a way to 'save' the values from our coding. We do this by defining a variable with a name that will be tied to that value just as the value is tied to the variable name. Once you create (or declare) a variable as having a particular name, you can then call up that value by typing the variable name. The way to declare a variable is: var bananas;**,  
points: 25,  
solution: [**'var'**]

level: 2,  
chapter: 1,  
firstImage: **UNKNOWN'**,  
secondImage: **'UNKNOWN**,  
challengeText:  
**var x = ‘apples’**

**What is x? \_\_\_\_\_**

instructionText: **Fill in x**,  
learnText:

**Let’s do some more practice! I’ve created a variable x. Can you tell us what x is pointing to?**,  
points: 25,  
solution: [**'apples'**]

level: 3,  
chapter: 2,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**What is 6+10?**  
**What is 6-2?**

instructionText: **How could we solve these math problems?**,  
learnText:

**Did you know that it’s also possible to do math using programming? It works almost the exact same way real-life math does. Here’s a quick look at some problems:**

**1+1 equals 2**

**1-1 equals 0**,  
points: 25,  
solution: [**16, 4**]

level: 4,  
chapter: 2,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**What is 8\*9? \_\_\_\_**

**What is 5/1? \_\_\_\_**  
instructionText: **Here are some more problems, this time with multiplication and division? Can you figure out the answers?**,  
learnText:

**We can also do multiplication and division using programming. Multiplication is done with ‘\*’, while division is done with ‘/’. Here’s a quick look at some problems:**

**1\*1 equals 1**

**1/1 equals 1**,  
points: 25,  
solution: [**72, 5**]

level: 5,  
chapter: 2,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**What is -8\*9? \_\_\_\_**

**What is 1/5? \_\_\_\_**  
instructionText: **We’ve adjusted the problems to use negative numbers and decimals. How do you think this will change the answer?**,  
learnText:

**Like real math, programming math also uses negatives and decimal numbers (but not fractions!). The rules are the exact same as in real math!**,  
points: 25,  
solution: [**-72, 0.2**]

level: 6,  
chapter: 2,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**var x = 6+10 – (-8\*9);**

**What is x? \_\_\_\_**  
instructionText: **Time to combine variables and math! We’ve created a variable that’s also a mathematical equation. What do you think happens now?**,  
learnText:

**One of the fun things about programming is combining different concepts. It’s common to create variables that are mathematical equations, or even mathematical equations that use variables. For instance:**

**var x = 1+1;**

**In this case, x equals ‘2’!**,  
points: 25,  
solution: [**88**]

level: 7,  
chapter: 2,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**7 < 2;**

**Is this ‘true’ or ‘false’? \_\_\_\_**  
instructionText: **Do you think this is true or false?**,  
learnText:

**Programming also has a concept called booleans. A boolean will either be ‘true’ or ‘false’. But ‘True’ or ‘False’ do not work!**,  
points: 25,  
solution: [**false**]

level: 8,  
chapter: 2,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**5 > 9 \_\_\_\_**

**10 <= 10 \_\_\_\_**

**3 === 10 \_\_\_\_**

**3 !== 10 \_\_\_\_**  
instructionText: **Are the following things true or false? Remember not to capitalize!**,  
learnText: **Finally, just like math, programming has comparative operators like ‘greater than’ or ‘less than’. Here’s a full list:**

**Greater than: >**

**Less than: <**

**Greater than or equal to: >=**

**Less than or equal to: <=**

**Equal to: ===**

**Not equal to: !==**,  
points: 25,  
solution: [**false, true, false, true**]

level: 9,  
chapter: 3,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**if (5 < 10) {**

**var x = 2;**

**}**

**\_\_\_\_**  
instructionText: **We’ve created a program just like the above, but with a different condition. What is ‘x’?**,  
learnText:

**Time for a fundamental feature of programming: ‘if’ statements. Let’s say we want to write a program in which we create a variable ‘x’ depending on certain conditions. If 10 is more than 5, we set ‘x’ as 1. Let’s take a look:**

**if (10 > 5) {**

**var x = 1;**

**}**

**Here’s exactly how the statement works:**

**First we add an ‘if’ keyword.**

**Next the ‘if’ keyword, we add a condition inside two parentheses ‘( )’.**

**Then we add a pair of curly braces ‘{ }’.**

**If the condition is true, we run the code inside the curly braces.**

**Here the condition is true: 10 is greater than 5.**

**Therefore we run the code inside the curly braces: x = 1**,  
points: 25,  
solution: [**2**]

level: 10,  
chapter: 3,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**if (5 < 10) {**

**var x = 2;**

**} else {**

**var x = 3;**

**}**

**\_\_\_\_**  
instructionText: **What is ‘x’ now? Is the condition inside the first parenthesis still true?**,  
learnText:

**What happens if the condition is false? For instance, what happens if we say:**

**if (5 > 10) {**

**var x = 1;**

**}**

**In that case, we need to use an ‘if’ / ‘else’ statement:**

**if (5 > 10) {**

**var x = 1;**

**} else {**

**var x = 2;**

**}**

**We add ‘else’ at the very end of our statement.**

**If the condition is ‘true’, the code inside the first pair of curly braces will run.**

**‘x’ would be 1.**

**If the condition is ‘false’, the code inside the second pair of curly braces will run.**

**‘x’ would be 2.**

**Here the condition is ‘false’, so x is equal to 2.**  
points: 25,  
solution: [**2**]

level: 11,  
chapter: 3,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**var y = ‘apple’;**

**if (y === ‘apple’) {**

**var x = ‘apple’;**

**} else {**

**var x = ‘orange’;**

**}**

**\_\_\_\_**  
instructionText: **Can you solve the problem now? What is ‘x’?**,  
learnText:

**Let’s do some more practice. We’ve created another ‘if’ condition. Except this program uses knowledge we’ve previously learned, like ‘===’.**  
points: 25,  
solution: [**‘apple’**]

level: 12,  
chapter: 3,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**var y = ‘apple’;**

**if (y === ‘orange’) {**

**var x = ‘orange’;**

**} else if (y === ‘banana’ {**

**var x = ‘banana’;**

**} else {**

**var x = ‘apple’**

**}**

**\_\_\_\_**  
instructionText: **We’ve modified the previous equation to have another condition? What do you think ‘x’ is now?**,  
learnText:

**How tough was that? Hopefully not too tough. :) But what happens if we want to add more than one condition? Then it’s time for the ‘else if’ condition! Let’s take a look at the following:**

**var y = 10;**

**if (y < 0) {**

**var x = ‘apple’;**

**} else if (y > 5) {**

**var x = ‘orange’;**

**} else {**

**var x = ‘banana’;**

**}**

**Here we have the following structure:**

**if ( /\* initial condition \*/ ) {**

**// do something**

**} else if ( /\* another condition \*/ {**

**// do something else**

**} else { /\* if none of the conditions are true \*/**

**// do the third thing**

**}**

**y equals 10.**

**If y < 0, x equals ‘apple’.**

**If y > 5, x equals ‘orange’.**

**Otherwise x equals ‘banana’.**

**y > 5, so x equals ‘orange.’**

points: 25,  
solution: [**‘apple’**]

level: 13,  
chapter: 4,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**var x = 0;**

**for (var i = 1; i < 4; i++) {**

**x++;**

**}**

**// At the end, x equals 3.**

**\_\_\_\_**  
instructionText: **Submit ‘true’ to continue!**,  
learnText:

**For our final chapter, we’re going to introduction ‘for’ loops. This lesson just shows you the final structure of a ‘for’ loop.**

points: 25,  
solution: [**‘true’**]

level: 14,  
chapter: 4,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**var x = 0;**

**for (var i = 1; i < 4; i++) {**

**x++;**

**}**

**// At the end, x equals 3.**

**\_\_\_\_**  
instructionText: **At the very beginning of the loop, what is i?**,  
learnText:

**Let’s focus on how a ‘for’ loop starts. Every ‘for’ loops uses a variable to count. The variable can be called anything, but here we choose to call it ‘i’.**

**The first part of the loop tells us to start with a value of 1 (‘var i = 1’).**

points: 25,  
solution: [**1**]

level: 15,  
chapter: 4,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**var x = 0;**

**for (var i = 1; i < \_; i++) {**

**x++;**

**}**

**// At the end, x equals 3.**

**\_\_\_\_**  
instructionText: **If we want the loop to end when x equals three, how should we fill in the blank space?**,  
learnText:

**How do we know when a ‘for’ loop ends? Well, let’s take a look at this example:**

**var x = 1;**

**for (var i = 1; i < 10; i++) {**

**x++;**

**}**

**Here, this loop will keep on running until i = 9 (that is, while i < 10).**

**When i equals 10, i will not be less than 10.**

**So the loop will no longer run.**

points: 25,  
solution: [**4**]

level: 16,  
chapter: 4,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**var x = 0;**

**for (var i = 1; i < 4; i\_\_) {**

**x\_\_;**

**}**

**// At the end, x equals 3.**

**\_\_\_\_**  
instructionText: **Fill in the blanks with the appropriate symbols.**,  
learnText:

**Let’s talk about the ‘++’. ‘++’ means that we’re going to increase the value of something by one.**

**For instance, let’s say we have:**

**var x = 1;**

**x++;**

**Now x is equal to two.**

**By the way, ‘—-‘ does the opposite of ‘++’.**

points: 25,  
solution: [**++**]

level: 16,  
chapter: 4,  
firstImage: **'UNKNOWN'**,  
secondImage: **'UNKNOWN'**,  
challengeText:

**var x = 0;**

**for (var i = 1; i < 4; i++) {**

**x++;**

**}**

**\_\_\_\_**  
instructionText: **What is x at the end?**,  
learnText:

**Let’s wrap things up by going step-through-step through this loop. Here’s what happens:**

1. **We start with i = 1.**
2. **Is i less than 4?**
3. **Yes it is. Therefore we increase x by one. x = 1**
4. **Now we go back to the loop between the parenthesis. We increase i by one.**
5. **i = 2**
6. **Is i less than 4?**
7. **Yes it is. Therefore we increase x by one. x = 2**
8. **Now we go back to the loop between the parenthesis. We increase i by one.**
9. **i = 3**
10. **Is i less than 4?**
11. **Yes it is. Therefore we increase x by one. x = 3**
12. **Now we go back to the loop between the parenthesis. We increase i by one.**
13. **i = 4**
14. **Is i less than 4?**
15. **NO.**
16. **We’re done. We break out of the loop. x = 3**

points: 25,  
solution: [**3**]