
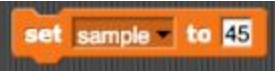


**Ongoing Documentation - you are each responsible for documenting the work you (individually) got done each day.**

DATE:	Explain what you got done today.
<p>Sample Research Day 1            _____5/5/2017_____</p>	<p>Today I researched JavaScript. I learned that it's good for a variety of uses and is commonly used on websites and can be used with HTML to make good but simple games. I want to do the "fizzbuzz" challenge, which is a very simple game used to help elementary school students to practice division and is also used as a common interview question for programmers. computation and conditional statement (in this case, just a string of numbers from 1 to 100) I am interested in using JS.</p>
<p>Sample Research Day 2            _____5/7/2017_____</p>	<p>Today I spent time on <a href="https://www.w3schools.com/js/">https://www.w3schools.com/js/</a> which is a free course on JavaScript. It has explanations, exercises and an area that you can use to try their exercises. I spent today learning how variables work in JavaScript. I learned how to set variables and that there are different kinds of variables in JS - just like in Snap there are local and global variables. However, in JS it works differently. The variable is automatically local or global depending on where you define it. But it also has constant variables, which let you set a variable once in a program - and it stays the same after that. These are named like this: const = 45; There are also variables like in Snap, which can be set and reset, as needed: var = 45; It's also really important to remember that variables are CASE sensitive in JavaScript. So aVariable is different than AVariable</p>
<p>Sample Coding Day            _____5/8/2018_____</p>	<p>Today I wrote the code to do a "for loop" in JavaScript. I only wanted to use the For Loop today I just had it "say" the numbers in the series that I was working with. I learned that you have to set the variable separately, which is equivalent to the round "i" in the Snap For block. For the iteration to work you use the word "for" and inside parentheses you set three parameters that tell the code how to iterate and how long to iterate. The first parameter tells it where to start, below it's 0, the next set tells it how long to go, in this case it's until it reaches the number 10, the final one tells it how to iterate, in this case it's by an increment of one. Inside the code you tell it what to do each time it iterates. i++ tells the code to set it to one plus itself. You could also set it like this i = i+1 and that's how you would make it increase by another increment - i = i + 2. Then I added a conditional to check if it was equal to the number 3. To do this I had to nest my code within the for function because I wanted it to check each number. I also learned that while a single equal will set a variable to a given thing, a double equals == will check whether it is equal to it. Also a != means not equal and is used to check whether a value is not equal to another and reports true only if it is not equal.</p> <pre>var i; for (i = 0; i &lt; 11; i++) {   console.log("it's"+i); }</pre>

```
    if (i == 3) {  
      console.log("It's " + i);  
    }  
  
    for (var i=1; i <= 20; i++)  
    {  
      if (i === 3) {  
        console.log("it is " + i);  
      } else {  
        console.log(i);  
      }  
    }  
  }
```

You are each individually responsible for learning and documenting at least four (4) concepts in the language you are working in, and for creating a side-by-side translation/dictionary between that language and Snap.

Concepts		{insert your language here}
<p>Variables are set in JavaScript using a single equals sign =</p> <p>Just like in Snap, there are different kinds of variables in JS. The important types to know are: <b>Local</b> versus <b>Global</b> and <b>Const</b> versus <b>Var</b>.</p> <p>A variable is automatically <b>local</b> or <b>global</b> depending on <i>where</i> you define it. If it's defined inside a function (which would be a custom block in Snap) it would be <b>local</b> and it wouldn't be accessible outside that function. If you define it outside of a function, it would be <b>global</b> and accessible anywhere in the program - inside a function or outside a function. This is similar to Snap in functionality since the end result is the same, but different because the programmer doesn't explicitly define variables as <b>local</b> or <b>global</b>.</p> <p>JS also has <b>constant</b> variables, which let you set a variable <i>once and only once</i> in a program - and it stays the same after that. These are named like this: <b>const theSame = 45;</b> There are also variables like in Snap, which can be set and reset, as needed: <b>var canChange = 45;</b> This is different than Snap because in Snap all variables can be redefined.</p> <p><b>**It's also really important to remember that variables are CASE sensitive in JavaScript.</b></p> <p>So <b>var aVariable = 45</b> is different than <b>var avariable = 45</b> <b>**</b></p>		<pre>var sample = 45</pre> <p>or</p> <pre>const sample = 45</pre>
