Teaching Variables with Boxes

(A Hands On Activity)

Age group: 10 to adult
Abilities assumed: No previous programming experience
Time: 15-20 minutes
Size of group: 3+

Summary:

Using a computer made of “students”, run a program that swaps variables. Following the instructions of the program, students, with boxes that symbolize variables, act out the process of swapping variables. The process of creating variables is acted out physically, as well as how accessing a variable involves taking a copy of its value, and how storing values in a variable destroys any previous value stored.

Technical Terms:

Assignment, variable, initialization, data type, variable name, value

Materials:

3 A4 Boxes
Variable name labels - printed on white paper
Value cards - printed on paper that is the same color as the spelled word
The program - program slides for Processing (Programs will be written on the board)

The Activity:

1. Explain to students that variables are a critical part to programming but is an easy thing to be confused about. Get over this hurdle and programming will be far easier.
    a. A good way to think of variables is that they are like special boxes with their own private shredder and photocopier built in - boxes that can store but also create and destroy.
2. It is helpful if the students have seen a simple demonstration of variables. (e.g. declare a String variable and print out its value on a computer).
3. Write the first program on the board. It swaps the values stored in the variables. Explain that you are going to demonstrate how the program works, and in particular what and how variables and assignment work.

   a. **What does this program do?**

```
String color1 = “green”;
String color2 = “blue”;
String temp;

temp = color1;
color1 = color2;
color2 = temp;
```

4. When programmers write programs they need a way to tell the computer to store data and move it around. **Variables** are a way to refer to places that store data. Programs have lots and lots of variables so you need a way to tell which is which so each variable created has its own **variable name**.
   a. In Processing, variables need to be created explicitly with data types. Explain to students that whenever we store a word, we have to put the **data type** String before the variable name. (Don’t go into too much detail, as it might confuse them and it is not the main objective of this assignment).
5. Point to the first instruction in the program.
   a. **String color1 = “green”;**
   b. Explain that this creates a new variable called color1 and stores the string “green” in it. Emphasize that it is best to pronounce the equals symbol as “gets the value” as in “color1 gets the value “green.”
6. Create the human computer.
   a. Pick a student from the classroom and give them a box to hold. Point out that you have just created a new variable - a storage space. A variable is just like a box that can hold a single thing. Since you will be adding more variables, you need a way to tell which one is which, so you need to give your variable a name.
   b. Take the name label for color1 and get them to put it around their neck. Point out that any previous name (e.g. the on their parent’s gave them!) is now forgotten. From now on they will be called color1.
   c. The rest of the command is the **assignment**. It just stores a value in a variable (e.g. it puts something in the box) .
   d. First, on the left hand side of the equal sign, you give the name of the variable and on the right side of the equal sign you give the value to be stored there.
      i. The value to be stored in color1 is “green”.
      ii. Get one of the value cards with “green” written on it and put it in the box called color1.
7. Move to the second instruction in the program.
   a. **String color2 = “blue”;**
   b. Pick a second student from the classroom. Give them the name label color2 and have them put it around their neck.
   c. Get one of the value cards with “blue” written on it and put it in the box held by color2.

8. Storing something in a variable for the first time is called **initializing the variable**. In the above assignments, we initialized color1 with the value “green” and the color2 with the value “blue”.
9. Emphasize the difference between **values** and the **names of variables**. Values are the things that can be stored in boxes. Names are just labels (hung around the necks of the volunteers). In programming languages, strings have quotes around the words so we know that they are values and not the names of variables. If there are no quotes then it will either be a keyword or the name of a variable.
10. The next instruction is:
    a. **String temp;**
    b. We haven’t seen temp before, we need to declare the variable. Pull out a third a volunteer and give them a box and the temp label. Explain that since no value has been assigned to the variable yet, we are not putting anything into the box.
11. We have initialized our three variables. Now we can do something with the data we have stored there. In our simple little program, we will just swap the two values over, so that at the end, color1 holds “blue” and color2 holds “green”.
12. The next instruction is:
    a. **temp = color1;**
    b. This assignment is different from the previous two. The right hand side is the name of the variable (not a value, because there are no quotes around the word). That means that we are going to get a copy of what is in the variable (i.e. what is in color1) and put the copy in the variable called temp.
    c. Go to the person representing color1, making a show of checking their name. Ask them what is in their box. Create a copy of it by getting a blank card and writing out the value. Put that copy into the box labelled temp. Emphasize that color1 has not changed at all - it still has its original value, which was not touched. The variable temp now just holds a copy of it.
        i. Emphasize also that the assignment was “temp gets a copy of the value in color1,” not the other way around.
13. The next assignment is similar:
    a. **color1 = color2;**
    b. The variables referred to here are both ones we previously created. This assignment just makes a copy of the value in color2 and places it in color1.
    c. Go to the person acting as color2 and find out what value they are storing. Make a copy. Again something new is happening. We are not just doing initialisation. We are storing a value in a variable that already has a value. It is important to emphasize that a variable can only store one value. When you put a new value in a box, you destroy the old one. As you put the new value in color1 make a show of taking out the old one and destroying it - tear it up, twist it, anything that theatrically destroys it. Point out that the value is now gone forever, and the new value stored is the one just copied from the other variable.
14. The final assignment is similar, this time copying the value from temp into color2
    a. **color2 = temp;**

    b. Act this out similar to the previous one. Point out that the value in temp was the one saved from color1 in the earlier step. That means that though the value in color1 was destroyed, because a copy was stored safely first, we end up with the same value in color2.

15. Finally, look at the results. What has happened? The values in color1 and color2 have been swapped over - using temp as extra storage space. Originally color1 was “green” and color2 was “blue”. Now color1 is “blue” and color2 is “green”

Extras (Faulty Swap):

1. Repeat the dry run process using a faulty swap that doesn't use an extra temporary variable
2. **What does this program do?**

   ```
   String color1 = “green”;
   String color2 = “blue”;

   color1 = color2
   color2 = color1;
   ```

3. See if the class can predict in advance what is going to happen. Get them to call out what should be done at each step and at the end get them to try to explain what has happened and why.