Chris Gross

PWS1 - Online

5-29-15

Recursion and JavaScript

Recursion is used in JavaScript but also in may other forms of programing. Recursion as I understand it is code that calls upon itself as many times as is needed to complete an action.

To describe recursion itself, the analogy was used that if you hold 2 mirrors parallel to each other, the reflection of each mirror into the other is an example of infinite recursion.

Recursion in JavaScript is to call a function from within a function. The most simplistic explanation that I have seen so far to describe what recursion does in programing is the “FloodFill” tool in most art programs. The FloodFill tool uses recursion in order to fill all of the pixels in a designated area with the chosen color. The loop would have to repeat on every pixel until all pixels in that area are filled. From what I have read, recursion use in JavaScript can be tricky to decide when to use it but once a situation is identified it is handy to know.

If I am correct, we have already used recursive functions in class. For example, when we have needed something to count down or repeat something a certain number of times we have used a recursive function that has called itself until it has completed its function.

For (var i = 0, i < 10, i++)

We are telling the function that if “i “ is less than 10, it needs to start over after incrementing the variable by 1. The function would call itself until the variable was not less than 10.

Another way recursion can be used is for looking into databases or searching for information in a given topic.

The function would be called in order to grab all information it can find in a given area to display to the user.

According to quite a few articles that I have read, the correct method of writing recursive functions is to never change the function. Instead of changing the functions, you change the information that is passed into them. The function must always remain the same to allow for accurate results each and every time. What is to be changed is the information that goes into them. The function should not know of anything else in the code accept for the information that it is meant to deal with and other functions that it needs to work with. The results of one function can be passed into another function but the means in which those results are found should remain solely in the one function. Not shared with another. This allows for seamless results that are never affected by confusion from other code.

Although I do not really understand everything that I have read, I can see how knowing how recursive functions work and should be laid out that they will be very handy in my programing future. Also they don’t seem as “Greek” to me.

Refrences:

McMillan, M. (2014). Data Structures and Algorithms with JavaScript

Beijing: O’Reilly

*http://whatis.techtarget.com/definition/recursion*

http://www.smashingmagazine.com/2014/07/02/dont-be-scared-of-functional-programming/