PLTW COMPUTER SCIENCE

Activity 1.3.7c

For Loops: Hangman

Introduction

As you've learned in the previous activities, computers can calculate and make decisions. A single calculation or decision would be unimpressive. Computers (and brains!) are impressive because they can make billions of calculations and decisions per second. Most programs don't have billions of instructions. A small handful of instructions repeated in a loop can be very powerful. In *Python*®, for and while loops are two of the control structures for iteration.

Iteration is a powerful idea even without computers. In knitting for example, a simple pair of stitches (knit and purl shown above) can be repeated with iteration in various patterns. What is something you enjoy doing that relies on iteration?

1. Complete the following. You must include a written deliverable in addition to your code.

   For the written deliverable, break the problem into at least two sprint tasks. (See Figure 9 at scrumreferencecard.com/ScrumReferenceCard.pdf). Each sprint task should have a clear, written "definition of done". The definition of the final task could be the IPython session shown in the problem. Perhaps the first sprint task is to get the function to work correctly for the first element of the first argument.

   Define a function `hangman_display(guessed, secret)` that returns the string a hangman player would see. The arguments are:

   `guessed`: letters guessed so far
   `secret`: the full secret word or phrase

   Hint: Start with the null string and add onto it one character at a time.

```
In []: hangman_display('aer', 'earth orbit')
Out[]: ear-- -r---
In []: hangman_display('fun', 'program puzzles')
Out[]: ------- -u-----
```