A Practical Introduction to Python

Lesson 2 of 3 - Magic 8 Ball

Lesson Description

This unit of work is about students exploring and using textual programming in a way that lets them do successful things. It is not about making sure that they understand the syntax for a while loop or an if statement.

In this lesson students will create a virtual Magic 8-Ball; a mystical, fortune-telling device that answers yes/no questions.

They will think about the sequence of instructions, type up the code provided and extend it to improve the program. The typing up of code and correction of inevitable syntax errors is a large part of the lesson. There is enough to do that students will feel a sense of achievement and ownership, but not so much that it becomes a lesson in typing.

Lesson Objectives

All students should learn:
- How to put instructions into a logical order
- How to type Python code to create a program

Most students should learn:
- How to correct simple syntax and typing errors
- How to use **selection** (if / elif statements)

Some students might learn:
- How to use functions with multiple parameters
- How to use a **while** loop to repeat code

Resources

- Software: Python 3.x
- Software: EasyGUI (easygui.sourceforge.net)

- 8-ball cut out sheet (double sided, cut into 6 cards, 1 set per group/pair)
- Magic 8-ball zip file containing
    - 8-ball Python program (incomplete)
    - 13x GIF images of a Magic 8-Ball, 1 blank & 12 responses

Lesson Outline

Starter

Load an online Magic 8-Ball and have students ask it yes/no questions
e.g. http://www.indra.com/8ball/front.html

Task 1

Hand out sets of the programming cards to students in pairs. Point out that one side of each card has natural English/prose and the other has code (written in Courier New). Students should sort the 6 cards into a logical order (algorithm). The PDF document has the 6 in the correct order.

Task 2

Students should download and unpack the resources for the lesson and open the Python file using IDLE. Turning the 6 cards over the students will be presented with code that they need to type into the program underneath the line **while True:**. It is vital that students indent their code correctly, using the tab key.

Once typed up the students should test the program and then troubleshoot the inevitable syntax errors. Patience and a supportive approach are key.

Task 3

Students should copy and paste the 3-line **elif** section of the code and edit it to add further text and images based on those provided. They will also need to amend the random number generator (e.g. choice = random.randint(1,6)).

Extension

If students complete all of this then they can use image editing techniques to generate new 8-ball responses.

Plenary

I always struggle with plenaries, you'll probably do a better job than I would ;-)