Introduction to Unplugged Programming for Teachers - Dave White

This taster workshop is an extract from the first session in a Training Course in Algorithms and Programming for level 1 CAS Master Teachers. This exemplar Course is designed by a joint teacher/academic team in the Department of Computer Science at University College London. Module 1 is designed as a foundation course suitable for primary and secondary teachers. It is being trialled in a small number of primary and secondary schools. Three of the 15 Workbooks which make up Module 1 are devoted to unplugged programming. They are in a developmental stage and available for viewing and downloading on the website ispython.com. Feedback is welcome.

'Unplugged' Programming, Computational Thinking and Teaching

This Workshop is an introduction to programming away from the computer screen; programming 3 of the 5 control structures of programming: **sequence, repetition and functions (with and without parameters).** In this part of the course we focus on developing programs, which we will build, run, test and evaluate, away from the demands and distractions of the computer screen.

We start to drive a pet/robot/rocket/sprite/turtle with programs which use 3 basic instructions. Consequently these programs are immediately transferable to Scratch 2.0 and Python 3, when and if we are ready to move to the screen. If you want to work at the screen straight away, feel free --- just use the equivalent Scratch or Python equivalent instructions (Figure 1.).

We quickly develop our own user functions/instructions (tools) to equip us to deal with a range of tasks. In the process we will be looking at aspects of how to teach computer science and make explicit the computational thinking involved in the tasks.

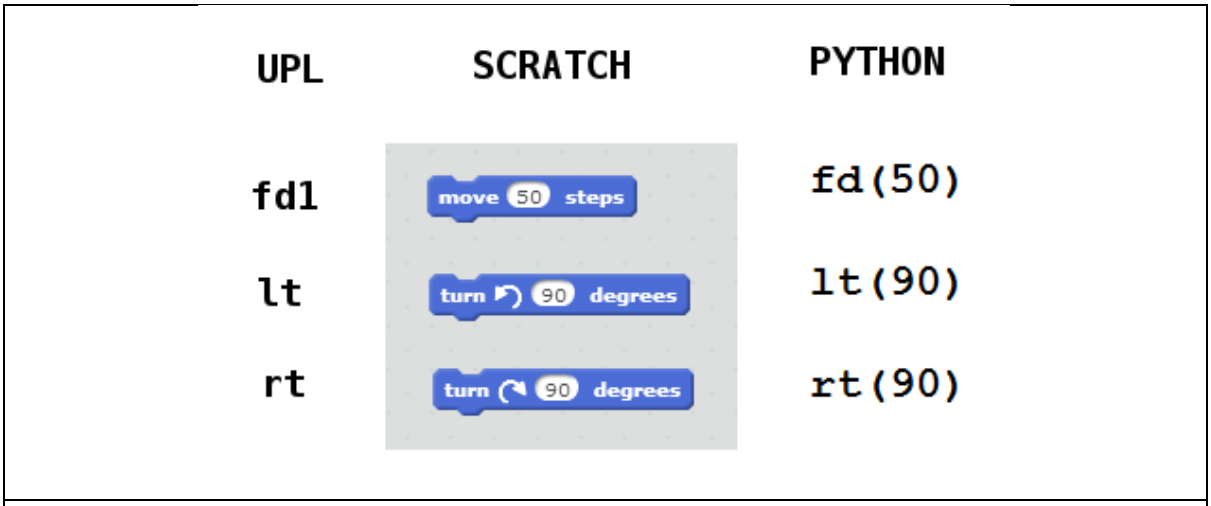

Figure 1. The 3 instructions for a pet/robot/sprite/turtle in UPL, Scratch and Python
UPL: Unplugged Programming Language

- `fd1` means move 1 pace forward and draw a path as you go;
  `lt` means just turn left through 90 degrees, similarly for `rt` to the right.
- The instructions can be acted out as commands to our pet/robot, where the pet/robot walks/rolls or turns left or right with each command given by a partner. The programs can all be demonstrated in this way and this helps to make the programming process more concrete.

- Alternatively the activities can be undertaken in pairs with pencil and (squared) paper.
- If you don't want to 'walk the talk', or talk and draw with pencil and paper, download up1.py from ispython.com/software and crack your simple programs by using the arrow keys.

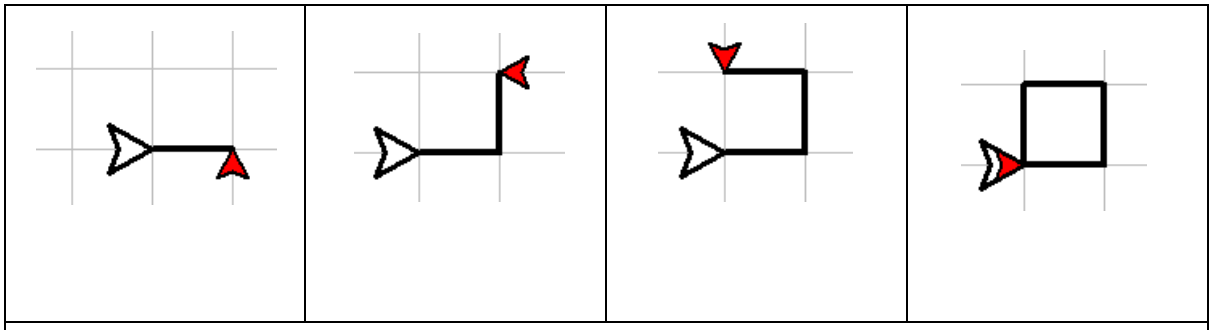

Figure 2. Repeat 4 times ['Move forward 1 pace and turn left']

Now Try your hand at ***Cracking the Code,*** **that is, discover the shape that the pet/robot traces out**

```
1. fd1 lt fd1 lt fd1 lt fd1 lt  # (this is function) sq

2. repeat 4[fd1 lt]
```

If sq is a function and represents the program in question 1.

```
3. repeat 3[sq fd1]

4. repeat 4[sq lt]  #(squaring the square)
```

If you don't get this and want to get it or if you get it and want a lot more then: come to the workshop, join in the action, and collect your first Workbook for a journey to the stars.