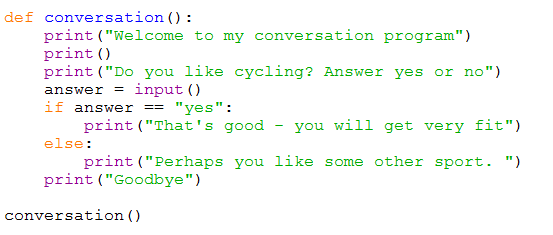
# Activity Sheet 2: Making choices

Last week we were using print() and input() statements to communicate with a computer program written in Python. Today we will write programs that do different things depending on what the person using the keyboard (the user) does.

**Task 1:** In pairs, look at the program below and write out what you think might happen when it runs



What would you expect the computer to do? Write it out exactly.

**Task 2**: Download and run the program and see if it does what you think it might do. You will find it at <insert your shared drive here>

Did the program run as you predicted? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 3**: There are a few new concepts in this program. In pairs, work out the answers to the following questions by examining the code and running it a few times.

1. What would happen if the program used the word Print() instead of print()?

…………………………………………………………………………………………………………

1. What would happen if you type “Yes” instead of “yes” or “no” when running the program ? (try it and see)

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

1. What is def used for?

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

1. Why does the program include conversation() at the bottom of the program?

…………………………………………………………………………………………………………

1. Why do you need a colon : at the end of the if statement and the word else?

…………………………………………………………………………………………………………

1. What is the difference between = and == ? Both are used in this program.

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

1. What would be the difference between running these two lines

* answer = input()
* answer = input(“Select 1 or 2…”) ?

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

1. Why is print(“Goodbye”) a few spaces **left** of the lines above? What happens if you change this?

…………………………………………………………………………………………………………

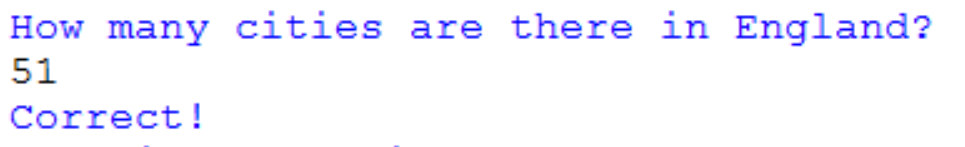
…………………………………………………………………………………………………………

Now add comments (beginning with #) to the program to make sure everything you understand is included in your program for future reference. When you think you understand how this program works then go on to the exercises.

**Task 4: Exercises**

In pairs, try these challenges:

1. Modify the program to have a conversation with somebody about sport. An example is given below but you can add your own questions.
2. Now write a function cities() and ask “*How many cities are there in England*?”. Now you are asking a question with a numeric answer use int(input(…)) to do this. The extra int() wrapped around the input will convert what you typed in into a whole number.



**Using > or < (greater than or less than)**

1. Write a new program to ask the user to enter a number bigger than 10. Give them some feedback if they have entered a valid number or not. Remember to use int(input()) – see what error you get if you forget to use it.
2. Now ask the user for their score in test. If they got less than 50 tell them that they have failed the test, otherwise tell them they have passed.

**Using >= (greater than or equal to>)**

1. Now ask the user of your program for their age. If they are 13 or over tell them they can have a paper round.

**Using != (not equal to)**

1. Write a program to ask the user to enter any number. As long as the number is NOT 99 that’s OK. Test that the number is not 99 by using if number != 99.

**Extension**

Write a program that asks the user to enter a number between 1 and 20. You can use and to check the condition. Give them positive feedback if it’s OK and say “too low” if it’s less than 1, and “too high” if it’s more than 20.

Make a note of any errors that you come across in your exercise book and how you fixed them.

**Key concepts from this sheet:**

|  |  |  |
| --- | --- | --- |
| Term | What it means | How to do it in Python |
|  |  |  |
| Selection | Where there is a choice point in the program design and an if statement is used to create more than one possible pathway. |  |
| Condition | What we test for inside an if statement. |  |
| Testing | Testing is essential to make sure that each pathway works correctly | Run the program as many times as there are options or combinations of options to ensure each pathway works. You can plan this using a *test plan* which is a table to record the results. A test plan makes sure your testing is systematic. |