Defining variables

In these notes, we will learn how to use variables to store information in your program.

What are variables?

In maths, we use a variable like x to represent some value, or a range of possible values. For example, the x in $f(x) = 5x^2$ could represent any real number, whereas the x in $3x - 6 = 0$ represents a particular number, in this case $x = 2$.

In programming, a variable can be a number, a string of text or various other data types. It holds one specific value at a time but can change.

Variable names

Python variables have names, which do not need to be a single letter. In fact, they are often better as more than a single letter because then you can give them a name that describes what they are. For example, a variable `NumberOfElephants` is clearer what it is counting than a variable `n`. This makes your code easier to read, and also helps when you reopen this code months or years later and can't remember what anything means!

There are some rules in Python about variable names.

- these can use any letter, the character _, and any number provided you do not start with it.
- variable names cannot contain spaces or signs with special meanings in Python such as `+` and `-`.
- you also cannot use a name that already has a special meaning in Python.

When you run your program, the complier should tell you about the error you have made, or your program will have strange results that will hopefully make it obvious that something has gone wrong.

If you want to use multiple words in your variable name, you cannot use a space. Two ways to deal with this are either squash the words together - use a capital at the start of each word to make it clearer to read - or use an underscore _ where you would use a space. So either `MySuperheroName` or `my_superhero_name`.

For example, the following are some acceptable variable names:

- `x`
- `giraffe123`
- `norfolk_building`

A note on case sensitivity

Case refers to whether a word is lower case or UPPER CASE.

Often, computers are case sensitive - as is maths! In maths, you could define $T = Ae^{kt}$ and there should be no confusion between T and t, as they are different things. The same is true in Python - the variables `T` and `t` are different.

Defining variables

You don't need to tell Python what type of information you want to store in each variable (whether it will be a number or text, for example). So to define a string or integer or decimal, simply use `=`.

For example, the following will work in Python:

```
x = 3
hello = "banana"
apple = 3.14159
```

Displaying a variable

In the IDLE Shell we could define `k=9` and then enter `k` to get it to display the current value of `k`. In a program, Python will not display output routinely.

If you want to know what value a variable holds you need to ask the program to output it using `print`.

For example if you run the following program, it should output "It was a bright cold day in April, and the clocks were striking thirteen.":

```
george="It was a bright cold day in April, and the clocks were striking thirte
print(george)
```

SymPy variables

Previously, we used SymPy to define mathematical variables. We did this by importing them from `sympy.abc`. For example

```
from sympy.abc import x
```

The `x` imported from sympy is special - it does not have a particular value, but acts like a mathematical variable and can be used in mathematical operations, for example

```
from sympy import *
from sympy.abc import x
print(sin(3*x))
```