|  |  |
| --- | --- |
| Lecture 1 Example 2 | Introduction to Python - Lecture 1 |
| Variables, Types and Assignment | What do the words variables and types mean? In this example we will explore what variables are and the fact that variables can have different types and can be assigned. Variables, Assignment and Comments The Oxford English Dictionary gives the (computing) definition of a variable is:  “*Computing*. A data item that can take on more than one value during or between programs and is stored in a particular designated area of memory; the area of memory itself; (also ***variable name***) the name referring to such an item or location.”  What does that mean? Let’s look at an example. Type the commands below into the shell or into a text file.  A=3 # Set A equal to 3.  print(A) # Print A  A=10.1 # Set A equal to 10.1  print(A) # Print A again  What is A then?  Do you notice anything different about the two values of A? |

The “=” symbol is what we call an assignment statement. We are *assigning* the number to the letter. If you used MuPaD in Maths, you may have seen how assignments work in MuPaD.

In MuPad, assignment statements work by having a :=. The grey bracket [ means do maths.

[ A:=3

The “#” symbol represents a comment in Python. Comments are ignored by the interpreter and different languages use different characters to represent comments. Python also accepts multi-line comments. They start with three ‘ marks and end with three ‘ marks. For example:

'''

This is a multiline comment

Even if I put

print("Hello World")

in a line, nothing happens

'''

If you enter this (above) into a file and run it, nothing will be printed as everything between the first three ‘ and last three ‘ marks are ignored.

# Types

Look again at

A=3 # Set A equal to 3.

print(A) # Print A

A=10.1 # Set A equal to 10.1

print(A) # Print A again

You might look at this and think that A represents a number. Which is true, however, there is a difference between these numbers. One is an integer and one is decimal. So, what?

Well, this is an example of *types.* Initially, in this example, when we assign A=3, Python sets A to be what is called an **integer** type (for hopefully rather obvious reasons). The assignment A=10.1 then changes the type because it is no longer an integer (there is a decimal place). This is what is called a **float**, meaning floating point variable.

Why is this important?

What are the other types or variable?

# Basic Types Task

For each of the types below, try to identify what the types are. Note here that:

* / means divide
* \* means multiply
* == means check if what is on the left is the same as on the right

These are examples of *operations*, which means doing something with the numbers/values. We will come back to this.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Type** |  | **Type** |
| 10 | Int | 2==1 | Bool |
| 0.1 | float | 1/2 | Float |
| “a” | String | 2\*1.0 | Int |
| True | int | 3\*”a” | String |

For each of the values in the table above:

* assign the value/operation to a variable
* print out the variable
* print out the variable’s type by using the Python command *type.*

For example:

A=10

print(A)

print(type(A))

What do you notice about the value of 1/2?

Float

This is because…

It Is not an integer??

What then might you expect from 3/2?