

Contents

1	Languages	1
1.1	Types of Languages	1
2	Python Basics	2
2.1	Case Sensitivity	2
2.2	Basic operations	2
2.3	Basic Math Functions (and Libraries)	2
2.4	The <code>os</code> Library	3
2.5	Script Files	3

1 Languages

There are many computer languages available (some would argue that there are too many). Python has rapidly become one of the most popular programming languages for general purpose programming.

1.1 Types of Languages

Fortran, C, C++, and Javascript (among others) are called strongly typed languages. This means that every variable that is used in a program must be given a specific type (double, integer, logical, complex, string) and array variables need to be given a fixed size before being used.

MATLAB is a weakly typed language. A variable can be whatever it needs to be at any time. For example, the sequence of statements below would be legal in MATLAB

```
>> a = 1           % a starts as an integer
>> a = 1.345      % a is now a double precision
>> a = -2 + 2i    % a is now a complex
>> a = (1>2)      % a is now a logical
>> a = [1 2 3 4 5 6] % a is now an integer array
```

This sequence would not be permissible in most languages. The variable `a` would need to be given a specific, fixed type before it could be used and this type could not change during the program execution. For the last statement, the language would need to be explicitly told that `a` is an array variable and the array would need to be given a fixed length.

```
INTEGER(10) :: a
integer a[10];
var a = new array(10);
```

Python is somewhere between MATLAB and strongly typed languages. In many cases, Python variables will behave like MATLAB variables in that they can change types as needed however, there are cases where variables will need to be given specific types.

One important thought to keep in mind is that Python is ultimately a programming language and thus has fundamental similarities with all other languages

- It has variables.
- It can perform operations such as basic math operations on these variables.
- It can perform conditional logic (`if-then` testing).
- It can perform looping operations.
- It permits you to write your own functions.

2 Python Basics

There is only 1 MATLAB environment. Octave exists, but it aims to exactly reproduce the MATLAB environment. There many Python *environments*. One some level they are all the same, but they may change appearance and how you interact with them. We will use the IDLE environment (mainly because this is the one installed on the campus network and the one a basic Python installation comes with). You can download this for use on your own computer.

The easiest way to work with this environment is to place any file with a `.py` extension into the directory you want to work in. Right click on this file and choose **Edit with IDLE** from the context menu. This will open the Pthyon editor and also set your working directory to whatever directory the file was in.

2.1 Case Sensitivity

Like MATLAB, variable names in Python are case sensitive. For example, open up the Python shell if you have not already done so, then enter the following

```
>>> A = 3
>>> a = 4
>>> A
3
>>> a
4
>>> c
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    c
NameError: name 'c' is not defined
```

Like MATLAB, we can display the value of a variable on the screen just by typing the value at the shell command line. However, we can't automatically display the value by leaving a `(;)` off the end of the line. An error occurs when we try to display the value of `c` because this variable does not exist.

2.2 Basic operations

The four basic operations (add, subtract, multiply, divide) are the same as in most other languages, but there are some additional operations to be familiar with There are actually a great many more operations,

Syntax	Operation
<code>x = y</code>	Set <code>x</code> equal to <code>y</code>
<code>x**y</code>	Raise <code>x</code> to the <code>y</code> power
<code>x//y</code>	Divide <code>x</code> by <code>y</code> and round down
<code>x%y</code>	Reminder when <code>x</code> is divided by <code>y</code>

Table 1: Math operators in Python

but we will not need to use them.

2.3 Basic Math Functions (and Libraries)

The basic math functions are also available. If you wanted to compute the sine of 5, you might try

```
>>> sin(5)
>>> Sin(5)
>>> sin[5]
>>> Sin[5]
```

however, all of these will give you a *name not found* error similar to the one in the example from the previous section. In order to gain access to these functions, we must make them available to Python.

These functions are contained in a *library*. A library is a collection of Python functions that can be used. Most languages (including MATLAB) have libraries for providing additional computing capabilities, however, MATLAB makes these known to the shell automatically. In Python, you need to explicitly bring these into your shell using the `import` command.

```
>>> import math          # Comment symbol in Python is the # sign
>>> a = sin(5)           ##### This gives an error
>>> a = math.sin(5)
>>> a
-0.9589242746631385     ##### Python uses 16 digit double precision
```

Because the sine function is part of the `math` library, it needs to be preceded by `math.` in order to access it. This is characteristic of all library functions in Python. We will use several libraries in the next couple of weeks.

Some of the standard math functions in the library are given in Table 2.

Function Name	Operation
<code>sqrt</code>	Square root function
<code>sin, cos, tan</code>	Sine, cosine and tangent
<code>asin, acos, atan</code>	Inverse sine, cosine and tangent
<code>sinh, cosh, tanh</code>	Hyperbolic sine, cosine and tangent
<code>asinh, acosh, atanh</code>	Inverse hyperbolic sine, cosine and tangent
<code>atan2(x,y)</code>	Four quadrant inverse tangent
<code>exp</code>	Exponential
<code>log</code>	Natural log
<code>log(x,y)</code>	Base y log of x
<code>degrees, radians</code>	Convert degrees to radians or vice versa
<code>pi, e, nan, inf</code>	Built in values for π , e , not a number and infinity.

Table 2: Some of the math functions in the Python Math library. Remember that each of these needs to be preceded by the `math.` designator. Also, all angles are assumed to be in radians when working with the trigonometric functions.

2.4 The os Library

Another library to know is the `os` library. This library is necessary if you want to enter command to navigate system directories in your shell. To use it, import the library. For example,

```
import os
>>> os.getcwd()          # Get current working directory
'R:\\Classes\\AF2018\\ESC251'
>>> os.chdir("P:\\ESC251") # Change working directory to P:\\ESC251
>>> os.getcwd()
'P:\\ESC251'
```

2.5 Script Files

As with MATLAB, you can enter your commands from the shell or you can use a *script file*. This is a file that contains a series of Python commands that get executed as if you typed them in at the shell prompt. Python script files are plain text files and have the `.py` extension. For example, enter the following into a file called `test.py`

```
import math
a = math.sin(5)
b = math.exp(2)
c = (a+b)/2
```

```
d = (a+b)//2
print(a)
print(b)
print(c)
print(d)
```

then enter the following at the command shell

```
>>> import test
```

This will import the commands in the script to the shell and execute the statements contained in the file `test.py`. A better way to execute the script is to choose **Run Module (F5)** from the **Run** menu in the IDLE editor.