Coding for Linguists

Course code =

Year of presentation =

Prerequisites = A laptop or PC

Contents page table is at link below

<https://d.docs.live.net/138e04b35f3df159/AllTeaching/_QMUL/Coding%20for%20Linguists/Syllabus%20-%20Coding%20for%20Linguists%20-%20Contents.docx>

Lesson 0  
About This Module

Course pre-requisites

1. You have a PC or laptop of your own running a Windows or Apple macOS or a Unix operating system
2. You know how to turn it on, connect to Wi-Fi, and browse the internet

Python 3

This is a coding course using Python 3.

What you will learn

The ability to:

* Design, build and test your own Python programs
* Make use of python’s software libraries for:
  + mathematics & statistics
  + text analysis & text processing
  + scraping text from websites and storing it on your PC
  + data analysis, machine learning & artificial intelligence
  + and many more tasks
* Communicate more productively with professional software developers

# Preparatory Work

1. Read – this reference
2. Read – This reference

Lesson 1  
Getting started with Python  
Variables, values, operations

Learning Objectives

On completion of this lesson and the practice exercises students will be able to:

* Install Python on own laptop/PC
* Have a basic understanding of the computer file system
* Start and use Python IDLE
* Write and execute Python statements in the interactive shell
* Save and reuse their Python programs as Python files/modules
* Understand what a ***value*** is and that it has a ***type***
* Create and use integer, float and string (text) value types
* Create a ***variable*** and assign it a value
* Understand and use the operators: +, -, \*, /, \*\*, %, //
* Understand and use the built-in functions int(), float() and str(), print() and input()
* Understand what polymorphism is and why it is useful

# You need a PC for this course

It is possible to successfully complete this course using one of the many browser-based python development environments available (e.g., replit.com or pythonanywhere.com) and interact with that through your tablet or phone but these lessons and course materials assume you are using a copy of python installed on your PC or laptop.

# What is ‘coding’?

Writing instructions for a computer to get things done.

# Why Python?

Because it’s ‘easy’ to get start learning it, there is lots of ready-made and high-quality software you can re-use for free, it’s used extensively for academic research and commercial projects.

Some words about learning a computer language

* Practice, practice, practice
  + The more you practice and the more often you practice the faster and better you will learn
* Vocabulary:
  + Expect to learn new words and new meanings for old words
* Pedantry:
  + Computers are the ultimate pedants and computer language interpreters and compilers put language mavens to shame
* Making mistakes:
  + Everyone makes mistakes, especially when writing code
  + Python will tell you when you make a syntax error and try and give you a helpful error message – READ IT!

# Python Books and Tutorials

## Books

There is no required textbook for this course

There are LOTS of suitable books. A few decent examples are:

* Python Basics: A practical introduction to Python 3 by the realpython.com tutorial team
* Automate the boring stuff with Python by Al Sweigart
* Learn Python in One Day and Learn It Well by Jamie Chan

## Tutorials

* <https://realpython.com/>
* <https://www.freecodecamp.org/>
* <https://www.w3schools.com/>
* and YouTube is awash with them

## Practice exercises and mini projects

As well as the practice exercise in this course and the above books and tutorials have a look at these :

* <https://snakify.org/en/>

# Installing Python on your PC

Open a web browser and go to:

[www.python.org](http://www.python.org)/downloads/

Install the latest version of Python 3 on your PC.

# Interactive programming using IDLE

* Using the Python IDLE interactive shell
* Numbers and arithmetic
  + Integer numbers
  + Floating point numbers
* BODMAS, BIDMAS, PEMDAS, BEDMAS
* Strings (text) and string operations
* Converting numbers to text and vice-versa with int(), float() and str()
* The notion of different ‘types’ of things in python

# Variables

Examples of variables

The problems variables solve

* Associates a name (the variable name) with a value.   
  That name can then be used in lieu of the value wherever that value is required.   
  If the value associated with the name is subsequently changed then it is automatically changed in every instance where that name is used
* Saves the result of a calculation so that it can be re-used without typing in and executing the whole calculation again
* Enables a problem to be broken down into simpler parts

# Computer file storage systems

Creating folders and files to contain your work

# Application programming using Python modules

Creating a python module

Re-using a python module

Re-using a python module within another Python module

# The print() function

* What print() does
* Your first ‘proper’ python program: ‘Hello World’
* Add two and two and print() the result
* Add ‘spam’ and ‘eggs’, and print() the result

# The input() function

* What input() does
* A program that asks for your name and then outputs: ‘hello, your name.’
* A program that asks for your name and then outputs: ‘hello, your name, my name is Python and I’m very pleased to meet you’

# Converting text to numbers and vice-versa

The built-in functions int(), float() and str()

# Doing calculations with user input

A few examples in class by me

# Computer Lab Practice

Practice writing code using integer, floating-point and text values

Manipulating values with operators

Lesson 2  
Functions & Parameters

# The problem that functions solve

Without them we keep having to write out the same code again and again. All that is changing is the values the code uses

# Functions are

Little nuggets of re-usable code

Name(parameters):

Block of code

Return a value

# Variables and Functions compared

A variable is the name of an address in memory where a value is store

A function is the name of an address in memory where executable code is stored

What a function is ‘really’

# Computer Internals

## Turing machines

Implies all languages with a conditional execution statement are equivalent.

## The little man computer and ‘Bruce’

John Searle’s ‘Chinese room’ model of a digital computer <https://en.wikipedia.org/wiki/Chinese_room>

Input → processing → output

The CPU

* Instructions: Machine code
* Operations: Arithmetical and logical

Memory

* Volatile or non-volatile
* Lightning fast … good-enough … slow … slowest

Lesson 3  
Conditional Execution, Collections, and Iteration - I

Lesson 4  
Conditional Execution, Collections, and Iteration - II

Lesson 5  
Reading and writing text files

Lesson 6  
Reading and Writing Web Pages

Lesson 7  
READING WEEK

Lesson 8  
Regular expressions ii

Lesson 9  
Statistical analysis of text

Lesson 10  
Data Visualisation

Lesson 11  
Jupyter Notebooks

Lesson 12  
Other programming languages  
Current trends  
What next?