**Storyboard for Module 1.2 – Programming with Python in a Jupyter Notebook**

**Last revised: 26th June 2019**

**Roles:**

* Curriculum design:
* Story writer/resource finder:
* Jupyter notebook coder:
* Tester/rewriter:

**Time scale:**

* Target date for completion of Modules 1.1 to 1.4 is end of July 2019

**Working environment:**

* MS Notebooks + Slack ?

**Module 1 theme: Basic Python concepts for corpus processing (3 + 1 weeks in total)**

Objectives of Module 1: (i) Understand the basic concepts of Python programming using Jupyter Notebooks in the Microsoft Azure environment (‘*Azure Notebooks*’). This will include variables, collection types, loops, control structures and functions. Along the way students will learn to (ii) Write efficient regular expressions to solve text-based extraction tasks such as sentence segmentation, part-of-speech tagging and building a simple ELIZA-like chatbot; (iii) Apply the edit distance algorithm to text sequence problems; (iv) Work with corpus data to calculate statistics using loops, dictionaries and counting; (v) Consolidate understanding of commonly used evaluation metrics such as accuracy, precision, recall and F-score.

**Module 1.1 (week 1) – Programming with Python in a Jupyter Notebook**

Number in brackets show estimated time to complete (needs testing). **Text in bold** highlights points of self-assessment or formal assessment. **Text in blue** corresponds to topics covered in the lectures.

Module 1.2 Week 1 (2 hours total)

**Short pre-module quiz** (<-- activate knowledge about Python and Jupyter from Module 1.1)

Getting started with Python notebooks

Basic Python expressions

1. Using Python as a calculator (<-- introduces variables, integers)
2. Working with strings (<-- introduces strings, string operations)
3. Introducing lists, dictionaries, tuples and sets
4. **Practice Quiz:** basic python expressions (5 multiple choice questions that the students can use for formative assessment)

First steps towards programming

1. Statements and loops (<-- introduces loops and control flow)
2. **Practice Quiz:** statements and loops (5 multiple choice questions that the students can use for formative assessment)

Reading text data from files

Debugging (<-- introduce the concept of errors, e.g. type errors. Errors are inevitable. Strategies for helping, e.g. printing out variables in the body of the code and how the Python debugger can help us)

Putting it all together (<-- read a file into a data structure and process it using basic Python expressions and loops)

**Quiz 1:** Basic Python assessment (assessed, 3 questions to be handed in on Week 4)