Digital Integration Teaching Initiative

Northeastern University
NULab for Digital Humanities and Computational Social Science

Python Fundamentals

Click to schedule a meeting with the DITI Team

Python is a computer programming language that is widely used in data science and the digital humanities. This handout provides a summary of the introductory Python fundamentals relevant to writing computational poetry.

Strings

A string is a piece of text that can include letters, numbers, and other characters. A string can be written with single or double quotes, but the quotes need to match each other and they need to be the "straight" version, not “curly/smart” quotes. Example string: `"This is a string."`

Lists

Lists in Python can store anywhere from zero to millions of values, and lists store these values in order. Example list: `['duck','turtle','cat'].` Items in a list can be referenced by their index number, or position in the list. The first item in a list has the index number 0, the second item has index number 1, etc. For example, the item at index 0 in the above list is `'duck'`.

Dictionaries

Like lists, dictionaries can hold many values within a single variable. In a dictionary, each value is stored in relation to a descriptive key forming a key/value pair. Below is an example of a dictionary (definitions quoted from Cambridge Dictionary):

```
{'duck':'a bird that lives by water and has webbed feet, a short neck, and a large beak',
'turtle':'a reptile that lives in the ocean and has a thick shell covering its body into which it can move its head and legs for protection'}
```

Items in a dictionary can be referenced using their key. For example, the key `'duck'` would reference the value `'a bird that lives by water and has webbed feet, a short neck, and a large beak'`

Variables

Variables in Python point to specific information. We create a variable with an assignment statement that gives the variable an initial value. For example, assigning the variable `color` the value of `'pink'`: `color = 'pink'`.

Conditional Statements

We can use conditional statements to run a portion of code if a condition is true, run a different portion of code if a different condition is true, or run no code. To see if a condition is true we use `==` to compare two things. To tell the computer what code we want to run under different conditions, we can use `if`, `elif`, and `else` statements. The below code will print out a different string of text depending on the value of the variable `color`:

```
if color == 'pink':
    print("That's my favorite color!")
elif color == 'blue':
    print("That's my second favorite color!")
else:
    print("That's not my first or second favorite color.")
```

Functions

A function is a collection of code which can be applied to complete a task. In general, functions can be thought of as a kind of recipe where you plug in a set of ingredients, or parameters, and the function follows the recipe to get the desired output. For example, the `print()` function prints out what is in the parentheses.

Libraries and Modules

To use functions that are not built into Python, we can import modules. A module is a Python file that includes function definitions. These modules can be collected into larger groups called packages and libraries. To access a function in a module, we use dot notation. Below is an example of importing a module and using dot notation to access a function by putting a dot (or period) between the module name and the function name.

```
import random
random_word = random.choice(word_list)
```