COMP1005 Week 2 Cheat Sheet

Lisa Luff

8/12/2020

Contents

Strings

- Characters are alphabetical (upper and lower), numbers, symbols and spaces
- The order of the characters matters, so a string is referred to as a sequence

Defining Strings

- Quotes indicate a string rather than a number
- Single or double quotation marks can be used, but they must be matching

Escaping Characters

- If you need a quotation mark or apostrophe within the string, you can "escape" it (or make it so it is not considered one of the enclosing quotation marks)
  - This is done with \
    - Eg. 'He\'s'
- Or you can use double quotes with single inside

Special Characters

- You can also use \ for special characters
  - \n - new line
  - \t - tab
- If you want to print these you need to use double \

Long Strings

- Python style suggests limiting each line of code to 79 characters
- If you want a long string that would cover multiple lines, use \ to split the string
  - Eg. "I'm sorry to have" \ "kept you waiting."
  - Style guide also advises to line up the opening quotes on each line
- If you're inside brackets you don't need \
  - Eg. print("Now you listen here!" (enter) "He's not the messiah.")
- You can use triple quotes to create an extra long string, that wraps across multiple lines.
  - Eg. ""“This parrot is no more. \nIt has ceased to be.”""
  - You don't need to escape single or double quote marks with triple quote marks, so this can be easier for actual quotes.

Style

Python Style

- Python is a community development, and "Python Enhancement Proposals (PEPs) used to define and pitch changes/standards
- PEP-8 provides a style guide, which we will use in the unit
  - https://www.python.org/dev/peps/pep-0008
- They are not rules, just guidelines to help consistency and readability

Style In This Unit

- We follow PEP-8
- We will write a README file for each practical, test and for the assignment
- We also require comments at the start of each program

Indexing

- As a sequence, we can assign a number to each element in a string
  - Commas, space, periods, etc. count as elements too
- Counting starts at 0
- Escaped characters don't count the \

Accessing Individual Characters

- With numbers assigned to each character position, we can pick out individual characters
- Element 0 is the first character, and so on
- We can use indexing in combination with the length of the string to loop through the characters
- Range() can give a sequence of numbers in a range going forwards, backwards or skipping
  - Not inclusive of stop value
- You can work back from the end of a string using negative numbers
  - For a sequence with 10 characters, the 10th character will be element 10, and element -1
- It can be helpful to consider the element numbers as being between the elements

Working with strings

Building Strings

- The main operator for string expressions is + or concatenate
  - This adds the strings together one after the other with no spaces
    - Eg. "John" + "Cleese" = "JohnCleese" and "John" + "" + "Cleese" = "John Cleese"

Printing Strings

- When printing strings, you can use a separator which will assign a character to be printed between each variable
- You can also add a character to be printed at the end of each line, which can be useful to keep loops together

Lists

- If you need lots of data in one place, then you can put it into a list
- Lists can contain numbers, strings, other lists or a combination
- The items in a list are kept in order
- You can access the elements with an index (like with strings)
- You can change, delete or add to the items in a list at any point
- Lists are flexible and dynamic
- Their index also starts at 0
- Duplicates are ok
- Strings can be split into a list of strings

Slicing

- We can slice strings and lists to access parts of them
- This is done the same as with the range function
  - If either side of a range is missing, it will default to 0 or the end
    - You can use positive and negative element numbers

Using External Programs

- The is a massive range of programs you can download and use in python
- To use, before calling on them, you need to import them everytime
  - import *program* as *callerID*
  - You can call it what you want for ease of use
- Then if you want to use a function that is part of the program, you need to call it and the function
  - *callerID.function*()

(Pseudo)Random Numbers

Generating Random Numbers

- The random program provides a random number generator for python
  - Call the random() function, and it will return the next random floating point values from the generated sequence
  - All returned values will be between $0 \leq n < 1.0$
- There is actually no such thing as random number generation, the proper term is pseudorandom number generator(PRNG)
- Properties of a good PRNG:
  - Very long period
  - Uniformly distributed
  - Reproducible
  - Quick and easy to compute
- Good PRNGs
  - Serial codes:
    - Mersenne twister - used in python
    - GSL (GNU Scientific Library), has many available
      - http://www.gnu.org/software/gsl/
  - For parallel codes:
    - SPRNG, considered the leading
      - http://sprng.cs.fsu.edu/
- Random.org
  - Offers true random numbers
  - Based on atmospheric noise

Seeding

- random() produces different values each time it is called
- There is a long wait before it repeats
- So if you want the experiment to be repeatable, you can use a seed value
    - This will mean the same numbers come up every code

Random Integers

- random() generates random floating point numbers
- You can get random intergers with randint()
    - The argument for randint() is an inclusive range
- randrange gives the ability to assign a step argument

Picking random items

- The choice() function makes a random selection from a sequence

The Random Module

- Can also support:
    - Saving state
    - Permutations
    - Sampling
    - Mulitple simultaneous generators
    - Non-uniform distributions

Monte Carlo Techniques

- Monte Carlo techniques model systems by simulating them through random sampling
- It is powerful, flexible and very direct
- Often the simplest, or even only feasible way of solving a problem
- Used in all quantitative areas of study