Week 2: Python Primer

Week 2 Outcomes

1. Python basics
    1. Triple-quoted strings, etc
    2. Defining functions, parameters
    3. Importing, etc
    4. Lists, Tuples
    5. Pass by reference
2. Sequences: Lists, Tuples
3. Dictionaries, Sets
4. Slicing, Two dimensions, etc

Lesson Overview (Agenda)

We will look at the following sections in the Jupyter notebooks for the textbook:

- 2.5 triple quoted strings
- 2.6 input and converting string to int as in int('7')
- 2.7 docstrings
- 3.8 keyword arguments
- 3.13 range function
- 3.14 decimal.Decimal import example
- 4.2 defining functions

- 4.2 None
- 4.2 <symbol>? and <symbol>??
- 4.9 default parameter values
- 4.10 keyword arguments
- 4.11 arbitrary argument lists and *<iterable> unpacking
- 4.13 scope and global keyword in functions
- 4.13 block vs suite
- 4.14 importing
- 4.15 Passing arguments by reference

- 5.2 negative list indices
    append to a list with += [sublist]
    concatenate lists with +
    using for and range
- 5.3 Tuples (immutable)
    one-element tuple ('red',) vs ('red') which equals 'red'
    Tuples may contain mutable objects like lists
- 5.4 Unpacking Sequences
    enumerate function
    tuple function
    sequence * value returns new sequence repeated value times

Time to check your learning!

Let’s see how many key concepts from Topic 1 you recall by answering the following questions!

- What do triple-quoted strings do?
- How can we turn a string '78' into an integer?
- What range function call would return a sequence from 3 to 8
- How are arguments passed in Python? (by-value or by-reference)?
- Are python tuples mutable or not?

5.5 sequence slicing

slicing with steps

modifying lists with slices

- 5.6 del statement
- 5.8 sorting lists mylist.sort() or mylist.sort(reverse=True)

sorted(mylist) returns a new sorted list

- 5.9 Searching sequences

mylist.index(value) searches mylist for value, and returns index

operators in and not in

functions any and all

-

5.10 built-in list methods: insert, append, extend, remove, clear, count, reverse, copy,
5.11: pop

- 5.12 list comprehension

    filtering with if

- 5.13 Generator Expressions

    use parentheses

- 5.14 lambda expressions

- 5.15 zip

- 5.16 two dimensional lists

6.2 Dictionaries

- 6.2.1 Creating a dictionary {}
- 6.2.2 Iterating through a dictionary
- 6.2.3 access value with dict[key]
- 6.2.3 del dict[key]
- 6.2.3 dict.get('key')
- 6.2.3 boolean test 'key' in dict
- 6.2.4 dict.keys() and dict.values()
- 6.2.4 converting keys, values to lists with list(dict.items())
- 6.2.9 Dictionary comprehensions

- 6.3 Sets
- 6.3 creating a set from a list: set(list)
- 6.3 frozenset creates an immutable set from an iterable
- 6.3.1 Comparing Sets, subsets
- 6.3.2 Union and intersection set type's methods
- 6.3.3 add remove discard

Time to check your learning!

Let’s see how many key concepts from Python Ecosystem you recall by answering the following questions!

- What's the difference between a Dictionary and a Set?
- How can we check whether a key is present in a dictionary?
- Can we convert a dictionary to a list?
- How can we add a single value to a set?
- How do we compare whether two sets are equal?