# Lesson 1:

* Environment Setup
  + Installing python \ pip (create doc as pre required)

https://www.tutorialspoint.com/python/python\_environment.htm

* + Pycharm \ Jupiter (create doc as pre required)
    - <https://ipython.org/notebook.html>
    - <https://www.jetbrains.com/pycharm/download>
* Type overview - **45 min**
  + Assignment and reassignment
  + Assign before access, multiple assignment
  + <https://realpython.com/python-variables/> - add to PP
  + Mixing operators between numbers and strings is not supported (3 + ‘a’)
  + Boolean: False = None, 0, [], {}, ‘’
  + Strings:
    - Functions: startswith, isalpha, islower. count, find, split
    - + and \*: ‘a’+’b’=’a b’, ‘a’\*3=’a a a’
    - Slicing: list[0:1], [start:stop:step] → [::-1]
    - ' '.join(['hello', 'world']) → 'hello world'
    - Format string (%s, format)
* Condition & loops (if, else, switch, for) - **30 min**
  + Object comparison using ‘is’ or ‘is not’
  + Boolean operations: ==, or, and, not
  + == vs is (https://www.learnpython.org/en/Conditions)
  + If-elif-else (one line)
  + For
  + Range
  + While, while-else
  + Break, continue, pass
* Functions - **45 min**
  + Declare & call
  + Args & kwargs (https://pythontips.com/2013/08/04/args-and-kwargs-in-python-explained/)
  + Default parameter value
  + Return (multi)
  + Scope and Lifetime of variables - add to PP
  + Adding main
* Containers (list, tuple, set, dict) - **30 min**
  + List:
    - list is a collection which is ordered and changeable
    - access the list items
    - change the value of a specific item
    - loop through the list items

## Check if Item Exists

* + - String as list of letters
    - Len, append, remove, pop, extend, reverse, sort
  + Tuple
    - ordered and unchangeable
    - access the tuple items
    - loop through the tuple items
    - Check if Item Exists
  + Set:
    - unordered and unindexed
    - [Difference](https://www.w3schools.com/python/ref_set_difference.asp), issubset
    - From list to set
  + Dictionary:
    - unordered, changeable and indexed
    - Keys as numbers\strings\mix
    - access the dict items
    - loop through the dict items
    - Check if Item Exists
    - Create (dict() vs {})
    - Add, remove, keys, items, values

# Lesson 2:

* Exception - **15 min**
  + Try-except-else
  + Try-except-except
  + try-except-finally
* Modules and packages
  + Module, package, import, dir
* Virtualenv, pip and requirements.txt
  + Pip install uninstall list
  + https://www.dabapps.com/blog/introduction-to-pip-and-virtualenv-python/
* Debug
  + Import pdb \ ipdb
* Classes and Objects:
  + Everything is an object
  + The \_\_init\_\_() Function
  + Self parameter
  + Inheritance (+multiple)
  + Overloading - https://www.programiz.com/python-programming/operator-overloading
* Decoration:
  + Decorator - <https://www.programiz.com/python-programming/decorator>
  + Decorator with params
  + Chaining decorators (?)
* Generators - https://www.programiz.com/python-programming/generator
* Context manager
  + With files
  + With lock

# Lesson 3:

* Functional programming
  + Having a function as a return value (https://www.programiz.com/python-programming/decorator)
  + Having function as method argument
  + Closures
  + Lamdba
* Playing with containers
  + all, any, sum, zip
  + Sorted (with\out key)
  + Map, filter, reduce
  + List\dict comprehension (<https://treyhunner.com/2015/12/python-list-comprehensions-now-in-color/>)
  + Iterator - https://www.programiz.com/python-programming/iterator
  + Testing
  + Unittest framework and conversion
  + Conftest
  + fixture
  + Patch
  + pytest.mark.parametrize
  + Monkeypatch (?)

# Rest Application

Writing a rest application

Git repo will include basic docker with swagger

Course

# Environment Setup

* virtualenv env --python=python3
* source env/bin/activate
* python3 -m pip install --upgrade pip
* python3 -m pip install jupyter
* jupyter notebook