1. Beginner

**What is Python**

* Python2 vs Python3
* Pip
* Interactive Python
* Interpreted vs compiled language
* Scripting language
* Strongly typed

**Development env 1.**

* VS Code
* Pip install xxx
* AWS CLI
* Git

**Lexical elements**

* Indentation
* Comments
* Naming convention
* Operators
* Module v Package v Library v Framework
* Import statement
* Main function
* Docstring

**Data structures 1**

* Print
* Variable
* .format()
* Primitive data structures
* Boolean
* Integer
* String
* Float
* Unicode strings
* Null and None
* Integer operations
* String methods
* Type()
* Files
* import

**Functions 1.**

* Arguments and Parameters
* Variable scope
* Global
* Return
* Continue
* break

**Data structures 2.**

* Non-primitive data structures
* Sequences
* List
* List methods
* Set
* Set methods
* Dictionary
* Dictionary methods
* Dictionary views
* Concatenation
* Slicing
* Mutable and Immutable

**Execution control**

* If else
* Logical operators
* any(), all()
* For loop
* While loop

**Modules and libraries 1.**

* Modules
* Libraries
* Object
* Methods vs Functions
* Import json (dump, dumps, load, loads)

**Development env 2.**

* VS Code plugins:
  + Git lens
  + Json
* Linting

1. Intermediate

**Generators**

* Iterable
* Iterator
* \_\_iter\_\_
* \_\_next\_\_
* Generators
* Yield
* List comprehension
* Set comprehension
* Dictionary comprehension
* Enumerate

**Functions 2.**

* Callables
* \*args
* \*kwargs
* Lambda experssion
* Tenary operatons
* Map
* Filter
* Reduce
* Zip
* Global
* Execution context
* Function annotations
* namespaces

**Decorators**

* Closures
* Free variables
* Inner function
* Annotation
* @Property
* Wrapper
* descriptor

**Data Structures 3**

* Set
* FrozenSet
* Specialized Dictionaries
* DefaultDict
* OrderedDict
* Counter
* UserDict
* ChainMap
* Tuples
* Shallow copy
* Deep copy
* Equality vs Identity
* Collections
* Namedtuple
* Deque
* Linked List

**Exception Handling**

* Try/Except/Finally
* raise

**Serialization**

* Pickling
* JSON
* JSON Encoder
* Custom JSON Encoder

**Asynchronous Python**

* Asynch in node.js
* Asynchio
* Await
* Awaitable
* Promise
* Future

**Built-in modules**

* Functools
* Itertools
* Beautiful Soup
* Marshmallow
* Requests
* Regular experssions (re)

**Object oriented python**

* Classes
* Instances
* Class variables
* Magic methods
* OOP Principles (general)
* Design patterns (general)
* Self

**Testing**

* Unittest
* Profiling
* Test coverage

1. Advanced

Generator comprehension

Factory Functions

**Object introspection**

**One liners**

**Parallel programming**

* Thread vs Coroutine
* Coroutines
* multithreading

**Context Managers**

* with

**Functional programming in python**

* Memorization
* Currying
* Monads

Python garbage collector

Python memory management

Function caching

**Optimization**

**Web Development**

**Data Science**

**Python bytecode semantics**

**Inspect.stack**

**Importlib**

**Namespaces**

**Code generation**

**Asd module**

**Parsing**

* Plex

**Extending python**

* Cython

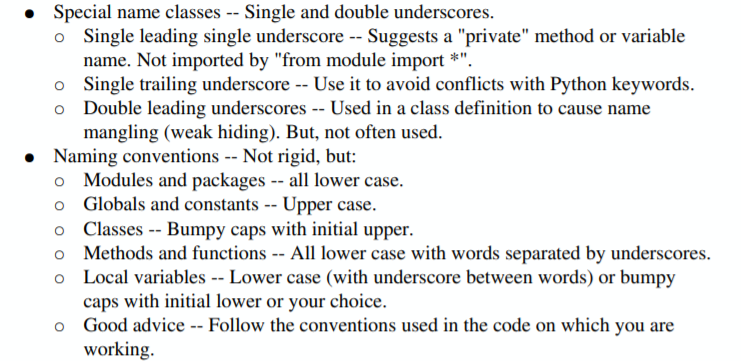
1. References
2. <https://www.davekuhlman.org/python_book_01.pdf>
3. <https://wiki.python.org/moin/FrontPage>
4. <https://readthedocs.org/projects/pythonguide/downloads/pdf/latest/>
   1. Courses

LinuxAcademy: Automating AWS with Lambda, Python, and Boto3

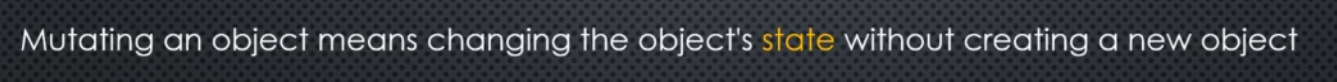
LinuxAcademy: Python 3 Scripting for System Administrators

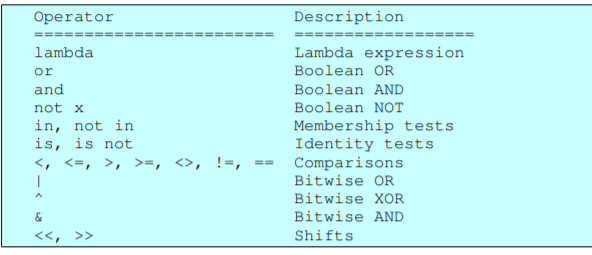
Corey Schafer – Python Tutorials (youtube) <https://www.youtube.com/watch?v=YYXdXT2l-Gg&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU>

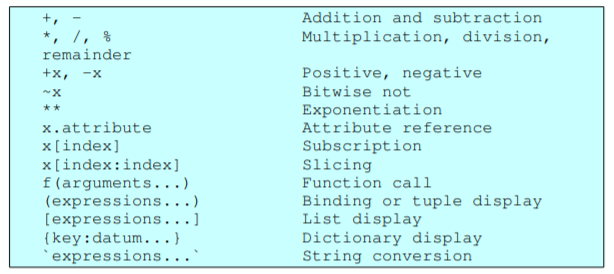
1. Notez
   1. Naming convention



* 1. Operators







* 1. Sequences and iterators

