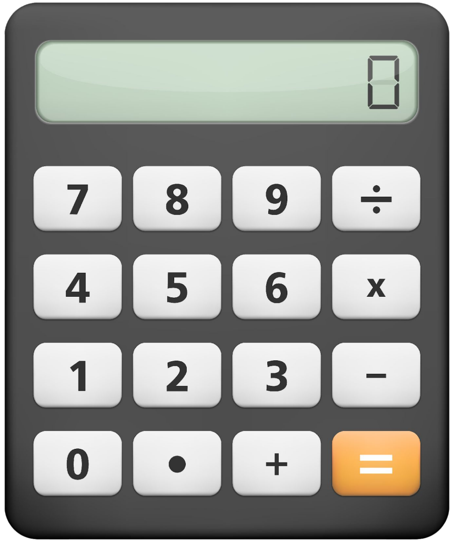
Chapter 2: Python Data Types

Resource: Introduction to Computing Using Python by Ljubomir Perkovic

Topics to cover:

- Python interpreter
- 2.1 Expressions, variables and assignments
- 2.2 Strings
- 2.3 Lists & Tuples
- 2.5 Math module (briefly)


7 8 9 ÷
4 5 6 x
1 2 3 −
0 • + =

==

Python Shell

Python: The Great Calculator

Algebraic Expressions

- Integer or int
- Floating point or float
- Remember **PEDMAS** (parenthesis, exponential, division, multiplication, addition, subtraction) & **Left to Right (L2R)**

Boolean Expressions

- Boolean (True, False)
- Boolean expressions (and, or, not)

Variables

Variable Names:

- Lowercase (a-z)
- uppercase (A-Z)
- underscore (_)
- digits (0-9) EXCEPT THE FIRST CHARACTER!!!
- CANNOT be reserved keywords

Examples:

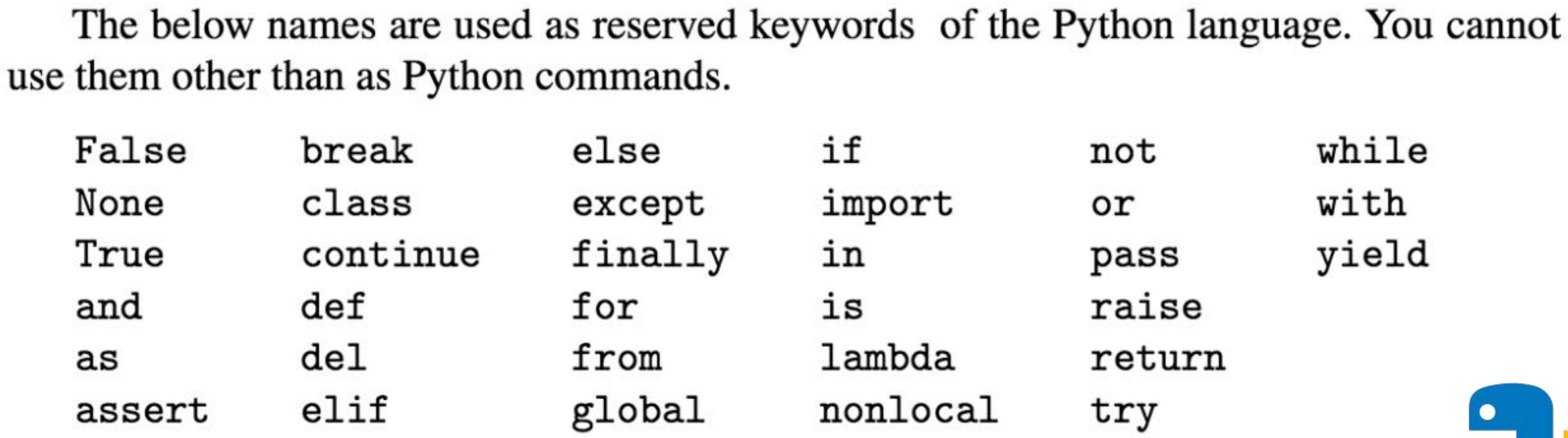
- myList, _list, list6, l_2 OK
- 51list, list-3 NOT OK
- O, l (el), I (eye) NOT OK
- ***** note: mylist & myList are different!!! Case sensitive! *****

Good practice:

- Be explicit!
  - Name \`price\` better than name \`p\`
- Multiple word name
  - Use underscore or camelCase
  - Interest_rate, InterestRate
- Pick one style and be consistent!
- Shorter meaningful names better than longer ones!
  - user_name, better
  - name_input_from_user, less better

Reserved keywords

PEP8 style guide: https://www.python.org/dev/peps/pep-0008/#naming-conventions



The below names are used as reserved keywords of the Python language. You cannot use them other than as Python commands.

`False`	`break`	`else`	`if`	`not`	`while`
`None`	`class`	`except`	`import`	`or`	`with`
`True`	`continue`	`finally`	`in`	`pass`	`yield`
`and`	`def`	`for`	`is`	`raise`	
`as`	`del`	`from`	`lambda`	`return`	
`assert`	`elif`	`global`	`nonlocal`	`try`	

Strings

String, denoted `str`

- Represent/manipulate text data (sequence/string of characters)
  - Blanks
  - Punctuation
  - Symbols
- Enclosed within matching quotes; either single quotes (‘) or double quotes (“). If it’s a huge string then triple quotes; triple single quotes (‘’’) or triple double quotes (“””)

Strings

Concatenate (+)

Multiply (*)

in/boolean operator

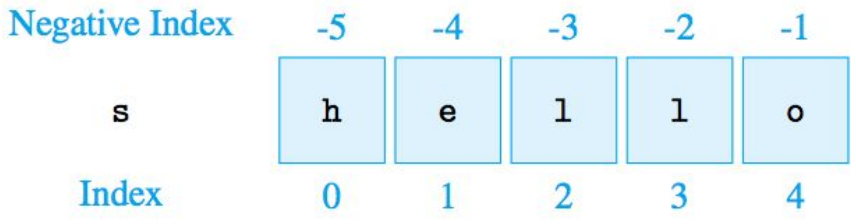
Substring

len

Indexing Operator [ ]

Index:

- Index of character in string = character’s offset (position in string) with respect to first character
- Count starts at 0….n
    - First character has index 0
    - Second character has index 1 (one away from first character)
- Going backwards:
    - Use negative indexes to access characters from the back
    - Last character has index -1



Lists [ ]

Organization

Certain order

List = sequence of objects of any type (numbers, strings, lists, etc)

- Comma separated sequence of objects enclosed within square brackets

Mutable unlike strings (immutable) - can change

Indexing operators

in/boolean operator

Concatenate

max/min/sum

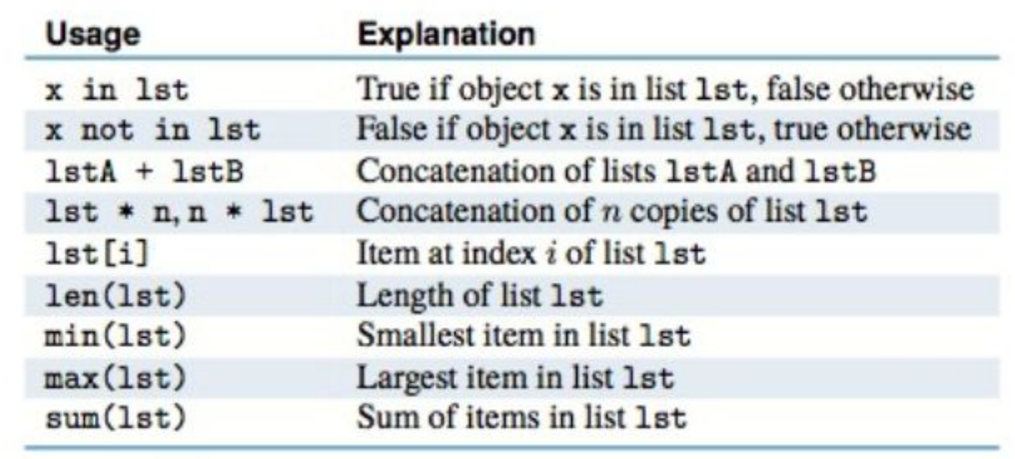
Append/count/remove/reverse/sort

Read the docs: https://docs.python.org/3/tutorial/datastructures.html

Tuples ( ) - immutable lists

Lists except tuples are immutable - cannot be changed

Contains a sequence of values enclosed by parentheses **( )** instead of brackets **[ ]**



Usage	Explanation
`x in lst`	True if object `x` is in list `lst`, false otherwise
`x not in lst`	False if object `x` is in list `lst`, true otherwise
`lstA + lstB`	Concatenation of lists `lstA` and `lstB`
`lst * n, n * lst`	Concatenation of n copies of list `lst`
`lst[i]`	Item at index i of list `lst`
`len(lst)`	Length of list `lst`
`min(lst)`	Smallest item in list `lst`
`max(lst)`	Largest item in list `lst`
`sum(lst)`	Sum of items in list `lst`

Objects and Classes

Objects - container for values (int, list...)

Every object has a **type** and a **value**

Object’s type - what kind of values object can hold and what kind of operations can be performed on object. Ex: int, float, bool, str, list

Python is **object-oriented** because values are always stored in objects vs other languages values explicitly in memory

Because every value in Python stored in object, python type is a class. Class = Type.

Python Modules (Libraries)

No need to rebuild what is out there

Import the module

https://docs.python.org/3/library/math.html

Use the documentation