Python Basics

Markus Dale, 2018

Python

python™

Python

- Guido van Rossum, late 1980s
- High-level scripting language
- Python interpreter/virtual machine
- all objects
- dynamically typed (variable name points to object)
- strongly typed (object's carry type information)
- Mark Lutz, Learning Python, 5th Edition, O'Reilly

Python Numbers - learning/types/numbers.py

- integers, floats
- int grows - no max value

```
# I'm a comment line - use me often
int1 = 12

#       2147483647 Java int max value
#       9223372036854775807 Java long max value
# Look ma, no overflow
int_bigger_than_long = 123456789101112131415161

float1 = 1.2
...
```

Strings - learning/types/strs.py

- Python Library Reference strs
- Immutable!

```
s1 = 'hello, I am a String and a sequence'

# concat and repetition
foo = 'f' + 'oo'
foobar = foo + 'bar'
whole_lotta_foos = foo * 8

# Immutability - 'str' object has no item assignment
# foo[0] = 'b'   # throws TypeError
...
```

Lists - learning/types/lists.py

- Mutable (but some operators return new list)

```python
py_invent = ['Guido', 'van', 'Rossum']

# List as sequence/iteration, index, slices... - new list
py_invent[0]        # 'Guido'
py_invent[:-1]      # ['Guido', 'van']

# size/count number of elements in list
len(py_invent)
...
```

Dictionaries - learning/types/dictionaries.py

- unsorted key-value store (think maps or hashtables)

```
# No order! Mutable! Keys, values of different types.
# keys must be hashable (__hash__(self))
ping_pong_wins = {'Chang': 98, 'Steve': 82}

ping_pong_wins['Chang']

# He won another - update value
ping_pong_wins['Chang'] += 1

# Add an element if key does not exist
ping_pong_wins['Troy'] = 70
...
```

Sets - learning/types/sets.py

- unique elements, mutable

```python
set1 = {'a', 'b', 'c'}

set2 = {'d', 'e', 'c'}

# union of sets as a new set; also | operator
set_union = set1.union(set2)
...
```

Tuples - learning/types/tuples.py

- immutable, positional lists

```python
 # Immutable
tup = ('foo', 100, 42)

'foo' in tup # true

word = tup[0]
word_count = tup[1]
doc_count = tup[2]

# unpacking a tuple
w, w_count, d_count = tup
...
```

Booleans - learning/types/booleans.py

- True, False
- Short-circuit evaluation
- Truth values

```
# two constant objects: True or False
true_val = 'a' == 'a'

false_val = 'a' == 'b'

# Boolean operations: and, or, not
not_true = not True

# No IndexError thanks to short-circuit evaluation
short_list = [1,2]
short_circuit_eval_or = (10 != 1) or short_list[99]
...
```

None - learning/types/none.py

- Python's null value, truth value 'False'

```python
def no_return_none():
    a = 100

def explicit_none_return():
    return None

none_means_no_return = no_return_none()
explicit_none = explicit_none_return()

# evaluates to truth value of false
if explicit_none:
    print('Test evaluated to True')
else:
    print('Test evaluated to False')
```

Files - learning/types/files.py

- input/output devices from file system and more

```
fin = open('io/yawl.txt', encoding='utf-8')
wordsStr = fin.read()
fin.close()
...
fout = open('foo.txt', mode='w', encoding='utf-8')
fout.write('this is foo\n')
fout.close()
...
```

Built-in documentation dir/help

```
# Run dir on object type or object variable
# lists all methods (in a list)
dir(str)
['__add__',...'isalnum', 'isalpha'...]

# help on specific method
help(str.center)
Help on method_descriptor:
center(...)
    S.center(width[, fillchar]) -> str

    Return S centered in a string of length width...
```

Python documentation and online resources

- Python 3 documentation home
- Python Tutorial
- Python 3 API
- Python HowTos
  - Logging
  - Regular Expressions
- Python FAQ
- Dr. Google
- Stackoverflow

References and Acknowledgements

- Python Logo: By www.python.org - https://www.python.org/community/logos/, GPL, https://commons.wikimedia.org/w/index.php?curid=34991637
- Learning Python 5th edition
- Python 3 documentation