

Python Introduction

Meiying Qin

Python

Interpreted language (vs compiled language)

No type declaration - BUT, it is a typed language

help() + google!

Example: help("string")

```
Help on module string:

NAME
  string - A collection of string operations (most are no longer used).

FILE
  /usr/lib64/python2.7/string.py

DESCRIPTION
  Warning: most of the code you see here isn't normally used nowadays.
  Beginning with Python 1.6, many of these functions are implemented as
  methods on the standard string object. They used to be implemented by
  a built-in module called strop, but strop is now obsolete itself.

  Public module variables:

  whitespace -- a string containing all characters considered whitespace
  lowercase -- a string containing all characters considered lowercase letters
  uppercase -- a string containing all characters considered uppercase letters
  letters -- a string containing all characters considered letters
  digits -- a string containing all characters considered decimal digits
  hexdigits -- a string containing all characters considered hexadecimal digits
  octdigits -- a string containing all characters considered octal digits
  punctuation -- a string containing all characters considered punctuation
  printable -- a string containing all characters considered printable

CLASSES
  builtin__.object
    Formatter
    Template

  class Formatter(builtin__.object)
    | Methods defined here:
    |
    | check_unused_args(self, used_args, args, kwargs)
    |
    | convert_field(self, value, conversion)
    |
    | format(*args, **kwargs)
```

More advanced



Introduction

Types

If statements

Loops

Print

Function

Classes

Types - int, float

```
>>> a = 5
>>> a += 6
>>> a
11
>>> a = 5.0
>>> a += 6
>>> a
11.0
```

```
>>> a = int(0.5)
>>> a
0
>>> a = float(3)
>>> a
3.0
```

Operator: +, -, *, /, //, %, **, ...

Types - int, float

Different from python 2, but still it is good to keep in mind the type

```
>>> a = 1 / 2
>>> a
0.5
```

Types - bool

True, False

Types - string

```
>>> a = "this is a string"
>>> a = 'this is a string'
>>> a[0]
't'
>>> a[:]
'this is a string'
>>> a[0:3]
'thi'
>>> a[-4:-1]
'rin'
>>> a[0] = "b"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
```

Types - string

```
>>> "1 + 2 = {:0d}".format(1 + 2)
'1 + 2 = 3'
```

Types - list

```
>>> a = ['a', 'b', 'c']
>>> a[0]
'a'
>>> a[:-1]
['a', 'b']
>>> a + a
['a', 'b', 'c', 'a', 'b', 'c']
>>> a.append('d')
>>> a
['a', 'b', 'c', 'd']
>>> a[0] = 'z'
>>> a
['z', 'b', 'c', 'd']
>>> a = ["one", "two", "three"]
>>> "one" in a
True
>>> "four" in a
False
```

Types - list

```
>>> a = ['g', 'a', 'm']
>>> b = a
>>> b
['g', 'a', 'm']
>>> a
['g', 'a', 'm']
>>> b.sort()
>>> b
['a', 'g', 'm']
>>> a
['a', 'g', 'm']
```

Types - list

```
>>> a = ['g', 'a', 'm']
>>> b = a[:]
>>> a
['g', 'a', 'm']
>>> b
['g', 'a', 'm']
>>> b.sort()
>>> b
['a', 'g', 'm']
>>> a
['g', 'a', 'm']
```

Types - list

```
>>> a = ['a', 'b', 'c']  
>>> ";".join(a)  
'a;b;c'
```

```
>>> a = [1, 2, 3, 6]  
>>> b = [7, 1, 4, 2]  
>>> list(map(lambda x, y: x + y, a, b))  
[8, 3, 7, 8]
```

```
>>> a = list(range(10))  
>>> a  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
>>> list(filter(lambda x: x % 3 == 0, a))  
[0, 3, 6, 9]
```

Types - tuple

```
>>> a = ('a', 'b', 'c')
>>> a[0]
'a'
>>> a[-1]
'c'
>>> b = a
>>> b
('a', 'b', 'c')
>>> a
('a', 'b', 'c')
>>> b[0] = "d"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
```

Types - dictionary

```
>>> a = {}
>>> a["name"] = []
>>> a
{'name': []}
>>> a["name"].append("pikachu")
>>> a
{'name': ['pikachu']}
```


Introduction

Types

If statements

Loops

Print

Function

Classes

if

```
a = 3

if a == 3:
    a += 5
elif a == 4:
    a += 4
else:
    a += 1
```

```
>>> [evaluate sample.py]
>>> a
8
>>>
```

if

```
a = []
```

```
if a:  
    a.append("b")  
else:  
    a.append("c")
```

```
>>> [evaluate sample.py]  
>>> a  
['c']
```

if

```
a = []
```

```
if a:  
    a.append("b")  
else:  
    a.append("c")
```

```
>>> [evaluate sample.py]  
>>> a  
['c']
```

```
>>> bool([])  
False  
>>> bool(["a"])  
True  
>>> bool("")  
False  
>>> bool("a")  
True
```

if

```
a = 1 == 1
```

```
if a == True:
```

```
    a = False
```

```
else:
```

```
    a = True
```

if

```
a = 1 == 1
```

```
if a == True:
```

```
    a = False
```

```
else:
```

```
    a = True
```

if

```
a = 1 == 1
```

```
if a == True:  
    a = False  
else:  
    a = True
```

if

```
a = 1 == 1
```

```
if a:
```

```
    a = False
```

```
else:
```

```
    a = True
```


if

```
a = ["a", "b", "c"]
```

```
if a == ["a", "b", "c"]:  
    print("YAY!")  
else:  
    print(":(")
```

```
>>> [evaluate sample.py]  
YAY!
```

```
a = ["a", "b", "c"]
```

```
if a is ["a", "b", "c"]:  
    print("YAY!")  
else:  
    print(":(")
```

```
>>> [evaluate sample.py]  
:(
```

Introduction

Types

If statements

Loops

Print

Function

Classes

loops - for

```
for i in range(5):  
    print(i)
```

```
>>> [evaluate sample.py]  
0  
1  
2  
3  
4
```

loops - for

```
>>> [x ** 2 for x in [1, 2, 3, 4]]  
[1, 4, 9, 16]  
>>> [x ** 2 + y for y in [1, 2] for x in [1, 2, 3, 4]]  
[2, 5, 10, 17, 3, 6, 11, 18]
```

loops - while

```
i = 5
```

```
while i >= 0:  
    print(i)  
    i -= 1
```

```
>>> [evaluate sample.py]  
5  
4  
3  
2  
1  
0
```

Introduction

Types

If statements

Loops

Print

Function

Classes

print

```
>>> a = "abc"  
>>> print(a)  
abc
```

print

```
print("before")
```

```
# might did something wrong here
```

```
print("after")
```


Introduction

Types

If statements

Loops

Print

Function

Classes

functions

```
def add(x, y):  
    return x + y
```

functions

```
def add(x, y):  
    return x + y
```

```
a = [3, 6, 2]
```

```
b = [2, 7, 4]
```

```
print(list(map(add, a, b))) ***
```

```
>>> [evaluate sample.py]  
[5, 13, 6]
```

functions

```
def function1(x):
```

```
    x = 5
```

```
def function2(x):
```

```
    x[0] = 3
```

```
x = 3
```

```
function1(x)
```

```
print(x)
```

```
x = [1, 2, 3]
```

```
function2(x)
```

```
print(x)
```

```
>>> [evaluate sample.py]
```

```
3
```

```
[3, 2, 3]
```

functions

```
def addition(x, y):  
    return x + y
```

```
def subtraction(x, y):  
    return x - y
```

```
def multiplication(x, y):  
    return x * y
```

```
def division(x, y):  
    return x / y
```

```
operations = [addition, subtraction, multiplication, division]
```

```
for operation in operations:  
    print(operation(10, 5))
```

```
>>> [evaluate sample.py]  
15  
5  
50  
2.0
```

Introduction

Types

Print

If statements

Loops

Function

Classes

classes

```
human = Human(15, 100)
print("human's height is " + str(human.get_height()))
print("human's age is " + str(human.get_age()))
```

```
student = Student(5, 1, 4.0)
print("student's height is " + str(student.get_height()))
print("student's age is " + str(student.get_age()))
print("student's grade is " + str(student.get_grade()))
```

```
def get_height(self):
    return self._height
```

```
def get_age(self):
    return self._age
```

```
class Student(Human):
    def __init__(self, height, age, grade):
        Human.__init__(self, height, age)
        self._grade = grade
```

```
def get_grade(self):
    return self._grade
```

human's height is 15

human's age is 100

student's height is 5

student's age is 1

student's grade is 4.0

Resources

https://www.w3schools.com/python/python_modules.asp

<https://www.codementor.io/sheena/python-path-virtualenv-import-for-beginners-du107r3o1>

<https://www.pythonforbeginners.com/comments/comments-in-python>

<https://realpython.com/python-comments-guide/>

Resources

<https://wiki.python.org/moin/MovingToPythonFromOtherLanguages>

Cheat sheet:

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwiR6r3qvOLfAhVGxYMKHSd_ARIQFjAAegQICRAB&url=https%3A%2F%2Fgithub.com%2Fehmatthes%2Fpcc%2Freleases%2Fdownload%2Fv1.0.0%2Fbeginners_python_cheat_sheet_pcc_all.pdf&usg=AOvVaw3SuqxiGtfMRN69R8fASwR

<https://www.python.org/about/gettingstarted/>

<https://www.w3schools.com/python/>

<https://www.toptal.com/python/top-10-mistakes-that-python-programmers-make> ***