PYTHON 3

Functions

```
def hello(name):
    '''
    Prints greeting.
    '''
    print('Hello ' + name + '!')
    return None
```

python hello_world.py 'World'

```
import sys
def main():
    hello(sys.argv[1])
if __name__ == '__main__':
    main()
# Hello World!
```

Strings

Basics

```
s = 'Hi'
print(s[1]) # i
print(len(s)) # 2
print(s + ' there') # Hi there
print('pi = ' + str(3.14)) # pi = 3.14
```

Raw

```
raw = r'this\t\n and that'
print(raw) # this\t\n and that
```

Multiline

```
multi = '''It was the best of times.
It was the worst of times.'''
print(multi)
# It was the best of times.
# It was the worst of times.
```

Methods

```
print(s.lower()) # hi
print(s.upper()) # HI
print(' H i '.strip()) # H i
print(s.isalpha()) # True
print('123'.isdigit()) # True
print(' '.isspace()) # True
print(s.startswith('Hi')) # True
print(s.endswith('Hi')) # True
print(multi.find('best')) # 11
print(multi.find('Hi')) # -1
print(multi.replace('times','cake'))
# It was the best of cake.
# It was the worst of cake.
print('H i'.split()) # ['H','i']
print('-'.join(['H','i'])) # H-i
```

Slices

```
s = 'Hello'
print(s[1:4]) # ell
```

Strings (Continued)

```
print(s[1:]) # ello
print(s[:]) # Hello
print(s[1:100]) # ello
print(s[-1]) # o
print(s[-4]) # e
print(s[:-3]) # He
print(s[-3:]) # llo
print(s[:2] + s[2:]) # Hello
```

Substitution

```
t = ('int=%d, str=%s, num=%f' %
(1, 'one', 1.0))
print(t) # int=1, str=one, num=1.0
```

Operators

Arithmetic

```
a, b = 10, 20
print(a + b) # 30
print(a - b) # -10
print(a * b) # 200
print(b / a) # 2
print(b % a) # 0
print(a**b) # 10 to the power 20
print(9//2) # 4
print(9.0//2.0) # 4.0
print(-11//3) # -4
print(-11.0//3.0) # -4.0
```

Comparison

```
print(a > b) # False
print(a < b) # True
print(a == b) # False
print(a != b) # True
print(a >= b) # False
print(a <= b) # True
```

Logical

```
print(True and True) # True
print(False or True) # True
print(not False) # True
```

If Statements

```
score, pet = 40, True
if score >= 90 and pet:
    letter = 'A'
elif score >= 80:
    letter = 'B'
elif score >= 70:
    letter = 'C'
elif score >= 60 or pet:
    letter = 'D'
else:
    letter = 'F'
print(letter) # D
```

Lists

```
colors = ['red', 'blue', 'green']
print(colors[0]) # red
b = colors # b now points to colors
print(len(b)) # 3
```

For (works on strings too)

```
nums = [1, 4, 9, 16]
sum = 0
for num in nums:
    sum += num
print(sum) # 30
```

In

```
names = ['larry', 'curly', 'moe']
if 'curly' in names:
    print('hooray!') # hooray!
```

Range

```
for i in range(10):
    print(i)
# print 0 through 9
for i in range(5,10):
    print(i)
# print 5 through 9
```

While

```
nums = [1, 4, 9, 16]
i = 0
while i < len(nums):
    print(nums[i])
    i = i + 2 # print 1 and 9
```

List Methods

```
nums.append(0)
print(nums) # [1, 4, 9, 16, 0]
nums.insert(0, 3)
print(nums) # [3, 1, 4, 9, 16, 0]
nums.extend([1, 2])
print(nums)
# [3, 1, 4, 9, 16, 0, 1, 2]
print(nums.index(2)) # 7
nums.remove(4)
print(nums) # [3, 1, 9, 16, 0, 1, 2]
nums.sort()
print(nums) # [0, 1, 1, 2, 3, 9, 16]
nums.reverse()
print(nums) # [16, 9, 3, 2, 1, 1, 0]
print(nums.pop()) # 0
print(nums.pop(1)) # 9
print(nums) # [16, 3, 2, 1, 1]
```

List Slices

```
list = ['a', 'b', 'c', 'd']
print(list[1:-1]) # ['b', 'c']
list[0:2] = 'z'
# replace ['a', 'b'] with ['z']
print(list) # ['z', 'c', 'd']
```

Lists (Continued)

Delete

```
list = ['a', 'b', 'c', 'd']
del(list[0])
del(list[-2:])
print(list) # ['b']
```

Sorting

```
a = [5, 1, 4, 3]
print(sorted(a)) # [1, 3, 4, 5]
print(a) # [5, 1, 4, 3]
strs = ['aa', 'BB', 'zz', 'CC']
print(sorted(strs))
# ['BB', 'CC', 'aa', 'zz']
print(sorted(strs, reverse=True))
# ['zz', 'aa', 'CC', 'BB']
```

Custom Sorting

```
strs = ['ccc', 'aaaa', 'd', 'bb']
print(sorted(strs, key=len))
# ['d', 'bb', 'ccc', 'aaaa']
strs = ['aa', 'BB', 'zz', 'CC']
print(sorted(strs, key=str.lower))
# ['aa', 'BB', 'CC', 'zz']
strs = ['xc', 'zb', 'yd' ,'wa']
def last(s):
    return s[-1]
print(sorted(strs, key=last))
# ['wa', 'zb', 'xc', 'yd']
```

Tuples

```
tuple = (1, 2, 'hi')
print(len(tuple)) # 3
print(tuple[2]) # hi
tuple = (1, 2, 'bye')
print(tuple[2]) # bye
tuple = ('hi',)
print(tuple[0]) # hi
(x, y, z) = (42, 13, 'bye')
print(z) # bye
```

List Comprehension

Square Elements

```
nums = [1, 2, 3, 4]
print([n * n for n in nums])
# [1, 4, 9, 16]
```

Elements to Uppercase

```
strs = ['o', 'm', 'g']
print([s.upper() for s in strs])
# ['O', 'M', 'G']
```

Elements Less Than

```
nums = [2, 8, 1, 6]
print([n for n in nums if n < 3])
# [2, 1]
```

Elements Containing

```
strs = ['o', 'm', 'g']
s = [s.upper() for s in strs if 'm' in s]
print(s) # ['M']
```

Dicts

Creating

```
dict = {}
dict['o'] = 'oh'
dict['m'] = 'my'
dict['g'] = 'gosh'
print(dict)
# {'o': 'oh', 'm': 'my', 'g': 'gosh'}
```

Get (Unsafe/Safe)

```
print(dict['o']) # 'oh'
if 'm' in dict:
    print(dict['m']) # my
print(dict.get('z')) # None
```

Iterating Over

```
for key in dict:
    print(key) # m g o
for key in dict.keys():
    print(key) # m g o
print(dict.keys()) # ['m', 'g', 'o']
print(dict.values())
# ['my', 'gosh', 'oh']
for key in sorted(dict.keys()):
    print(key, dict[key])
# ('g', 'gosh') ('m', 'my') ...
print(dict.items())
# [('m', 'my'), ('g', 'gosh'),...]
for k, v in dict.items():
    print(k, v)
# ('m', 'my') ('g', 'gosh') ('o', 'oh')
```

Substitution

```
print('%(o)s %(m)s %(g)s' % dict)
# oh my gosh
```

Delete

```
del(dict['m'])
print(dict)
# {'g': 'gosh', 'o': 'oh'}
```

Sets

Creating

```
s = set('Help me')
print(s)
# {'m', 'H', ' ', 'e', 'l', 'p'}
s = set([1,2,2,3])
print(s)
# {1,2,3}
```

Sets (Continued)

```
s = set(('r','g','g','b'))
print(s)
# {'g', 'b', 'r'}
```

Set Methods

```
s = {'r','g','b'}
s.add('a')
print(s) # {'g', 'a', 'b', 'r'}
ss = s.copy()
print(ss) # {'g', 'a', 'b', 'r'}
s.clear()
print(s) # set()
x = {'a','b','c','d','e'}
x.discard('a')
print(x)
# {'e', 'd', 'c', 'b'}
x.remove('b')
print(x)
# {'c', 'd', 'e'}
# remove throws KeyError
# discard does not
```

Files

Create/Write

```
f = open('foo.txt', 'w')
f.write('oh\nmy\ngosh')
f.close()
```

Readlines

```
f = open('foo.txt', 'r')
print(f.readlines())
# ['oh\n', 'my\n', 'gosh']
f.close()
```

Read

```
f = open('foo.txt', 'r')
print(f.read())
# oh my gosh
f.close()
```

Incremental Read

```
f = open('foo.txt', 'rU')
for line in f:
    print(line,)
# ('oh\n',) ('my\n',) ('gosh',)
f.close()
```

Open/Append/Close using With

```
with open('foo.txt', 'a') as f:
    f.write('\nyo')
```