

INTRODUCTION TO PYTHON

DEPARTMENT OF ECONOMICS

Nicholas Jackson

June 2018

- Originally developed by Guido van Rossum in 1991.
- Name inspired by *Monty Python's Flying Circus*.
- Open source software, community development model, overseen by the Python Software Foundation.
- Dynamic types, supports procedural and object-oriented paradigms.
- Core language is small, with 31 keywords:

and	as	assert	break	class	continue	def	del
elif	else	except	exec	finally	for	from	global
if	import	in	is	lambda	not	or	pass
print	raise	return	try	while	with	yield	

- Many specialist libraries available.

HELLO WORLD!

```
print("Hello world!")
```

- `int` : integer numeric, eg 1, -23, 42, 10191
- `float` : floating point numeric, eg 2.718, -3.0, 452.34
- `str` : string, eg "Hello", 'Python'
- `bool` : boolean, either True or False
- `list` : sequence of values, eg [1,2,3], ["Mary", "Mungo", "Midge"],
- `tuple` : a read-only list, eg (4,5,2)
- `set` : unordered list with no duplicate elements, eg {1,2,2,3,4,4} = {1,2,3,4}
- `dict` : associative array of key/value pairs, eg {1:'one', 2:'two', 3:'three'}

```
>>> type(3)
<type 'int'>
>>> type(3.0)
<type 'float'>
```

```
>>> type("Hello")
<type 'str'>
>>> type([1,2,3])
<type 'list'>
```

```
def square(x):  
    s = x*x  
    return s
```

```
def square(x):  
    if isinstance(x,int) or isinstance(x,float):  
        return x*x  
    else:  
        print("Argument must be numeric.")  
        return None
```

IF/ELIF/ELSE

```
if x < 0:
    print("x is negative")
elif x > 0:
    print("x is positive")
else:
    print("x is zero")
```

WHILE

```
k = 0
while k < 10:
    print(k)
    k += 1
```

FOR

```
for i in [1,2,3,4,5]:
    print(i, square(i))
```

```
l[3]           # fourth element of l
l[-1]         # last element of l
l[2] = 32     # set third element of l to 32
l.append('hello') # append 'hello' to the end of l
l.remove(4)   # remove first occurrence of 4 from l
l.pop(3)      # remove fourth element of l
l.insert(2, 'abc') # insert 'abc' before third element of l
len(l)       # number of elements in l
l[2:5]       # third to sixth elements of l
l[3:]        # fourth to last elements of l
l[:5]        # first to sixth elements of l
```


NUMERICAL LISTS

```
range(5)           # [0,1,2,3,4]
range(1,5)         # [1,2,3,4]
range(1,5,2)       # [1,3]
```

LIST COMPREHENSIONS

```
[x*x for x in range(1,5)] # [1,4,9,16]
```

```
d = {'one' : 1, 'two' : 'II', 'three' : 'drei'}

d['four'] = 'quatre' # add a new key/value pair to d
d['three'] = 'gamma' # modify an element of d
del d['two']         # delete a key/value pair from d
print(d['one'])     # print the value of an element of d
d.keys()            # ['one','three','four']
d.values()          # [1,'gamma','quatre']
```

```
f = open('file.txt', 'r')
for line in f:
    print(line.rstrip())
f.close()
```

Or:

```
with open('file.txt', 'r') as f:
    for line in f:
        print(line.rstrip())
```

```
with open('file.txt', 'w') as f:
    f.write('Hello world!\n')
```

```
>>> import math
>>> print(math.sqrt(2))
1.41421356237
```

Or:

```
>>> from math import sqrt
>>> print(sqrt(2))
1.41421356237
```

Or:

```
>>> from math import *
>>> print(sqrt(2))
1.41421356237
```