

Python basics

Xiao-jiang Li, Ph. D Candidate
Department of Geography, UCONN

Introduction of Python

- **Python is ideal for non-professional programmers**

Easy to learn, and powerful.

- **Python is Scripting language**

Python interpreter can interpret (rather than compile) and automate the execution of tasks

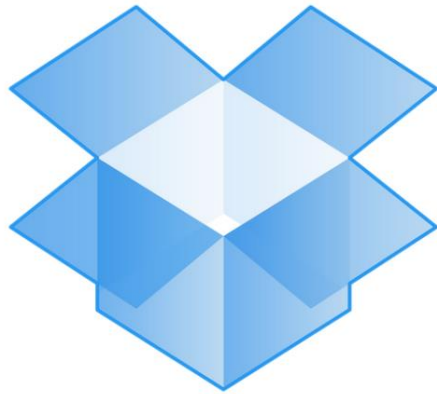
- **Python is for automation**

Work with software applications, Microsoft Office

- **Python is for science**

Numpy, Pandas, opencv...

Python applications

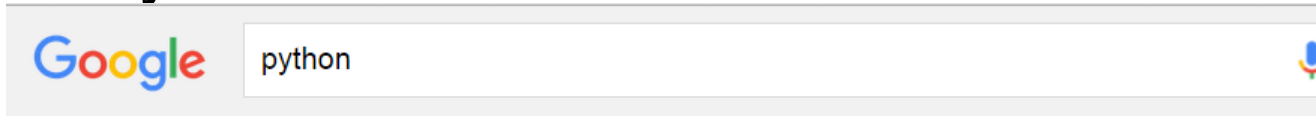


Dropbox



Python IDE

Python official website



[Web](#) [Images](#) [News](#) [Videos](#) [Books](#) [More ▾](#) [Search tools](#)

About 161,000,000 results (0.73 seconds)

Welcome to Python.org

<https://www.python.org/> ▾ Python ▾

The official home of the Python community

Search python.org

Download

Python 2.7.10 - Python 3.4.3 - Python 3.5.0

The Python Tutorial

1. Whetting Your Appetite
2. Structures - 9. C

Python 3.5.0

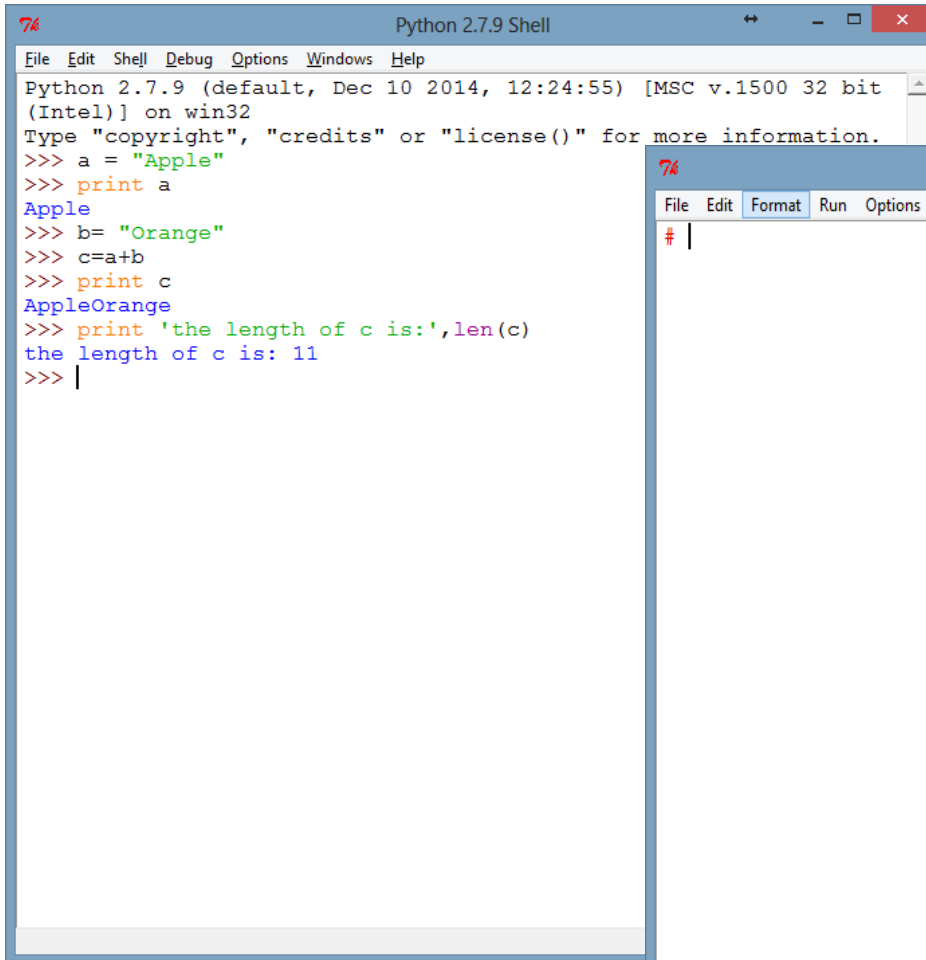
The Python Tutorial
Language Reference

A screenshot of the Python.org website homepage. The page has a dark blue header with the Python logo and the word "python" in white. Below the header is a navigation menu with links for "About", "Downloads", "Documentation", "Community", "Success Stories", "News", and "Events". The main content area features a large heading "Download the latest version for Windows" and two yellow buttons: "Download Python 3.5.0" and "Download Python 2.7.10". Below these buttons are links for "Here's more about the difference between Python 2 and 3.", "Python for Windows, Linux/UNIX, Mac OS X, Other", and "Pre-releases". The background of the main content area shows a blue sky with clouds and two yellow and white striped parachutes carrying cardboard boxes. At the bottom of the page, there is a section titled "Looking for a specific release?" with a link to "Python releases by version number".

Looking for a specific release?

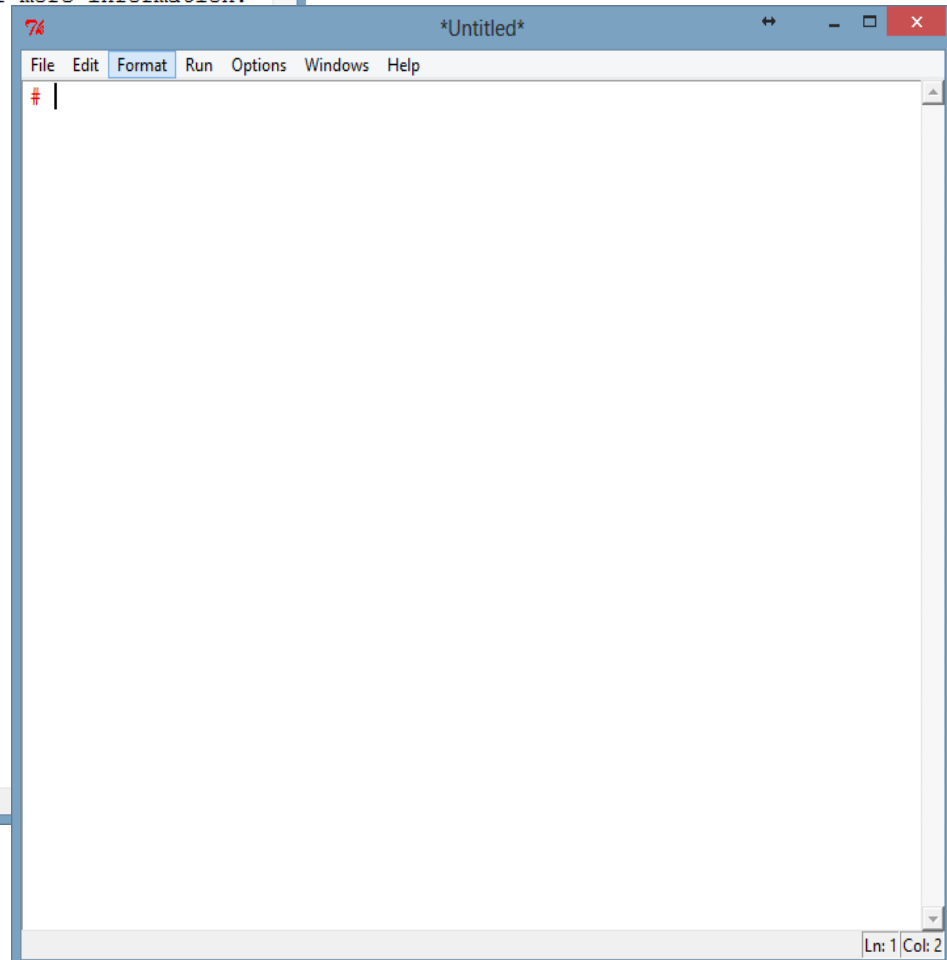
[Python releases by version number](#)

Python IDE



A screenshot of a Python 2.7.9 Shell window. The title bar reads "Python 2.7.9 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following code and output:

```
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit  
(Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> a = "Apple"  
>>> print a  
Apple  
>>> b= "Orange"  
>>> c=a+b  
>>> print c  
AppleOrange  
>>> print 'the length of c is:',len(c)  
the length of c is: 11  
>>> |
```



A screenshot of an "Untitled" editor window. The title bar reads "*Untitled*". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area contains a single line of code:

```
# |
```

The status bar at the bottom right indicates "Ln: 1 Col: 2".

Data types - Number

Integer

Number with no decimal part, 1, 2, 4

```
>>> a = 1
>>> print a
1
```

Float

Decimal number, like 1.10, 10.243

```
>>> b = 1.0
>>> print b
1.0
>>> c = 10.243
>>> print c
10.243
```

Data types – number operations

Number operations

Add

```
>>> 3 + 2
5
```

Subtract

```
>>> 3 - 2
1
```

Negative

```
>>> -2 + 3
1
```

Division

```
>>> 11.0 / 3
3.6666666666666666
```

Division (truncate)

```
>>> 11.0 // 3
3.0
```

Division remainder

```
>>> 11.0 % 3
2.0
```

Multiply

```
>>> 3 * 3
9
```

Exponent

```
>>> 3 ** 3
27
```

Exponent

```
>>> pow(3, 3)
27
```

Data types – number operations

Number operations of float and integer

```
>>> a = 1
>>> c = 10.243
>>> print a+c
11.243
```

Conversion between float and integer

```
>>> fa = float(a)
>>> print fa
1.0
>>> ic = int(c)
>>> print ic
10
```


Data types – string

Strings are text, anything that is enclosed by quotes.

Single or double quotes are fine but **consistency** is needed for a given string.

```
>>> a = 'Apple'
```

```
>>> b = "Apple"
```

```
>>> c = 'Apple"
```

```
SyntaxError: EOL while scanning string literal
```

Data types – string

String operations

1. Print strings

```
>>> a = 'c:\desktop\good'
>>> print a
c:\desktop\good
>>> b = 'c:\desktop\notgood'
>>> print b
c:\desktop
otgood
```

Why? How to print 'c:\desktop\notgood' correctly

Add a 'r' before the string

```
>>> b = r'c:\desktop\notgood'
>>> print b
c:\desktop\notgood
```

Data types – string

String operations

2. Connect two strings

```
>>> a = "Apple"  
>>> print a  
Apple  
>>> b= "Orange"  
>>> c=a+b  
>>> print c  
AppleOrange
```

3. The length of a string

```
>>> print c  
AppleOrange  
>>> print 'the length of c is:',len(c)  
the length of c is: 11  
>>> |
```

Data types – string

4. Other string operations

Lowercase, Uppercase, split ...

```
>>> a = 'Apple'
```

```
>>> print a
```

```
Apple
```

```
>>> print a.lower()
```

```
apple
```

```
>>> print a.upper()
```

```
APPLE
```

```
>>> print a.split('l')
```

```
['App', 'e']
```

Data types – string

5. Variable substitution in strings

The substitute is used to create a string dynamically.

%s is for string

%f is for float number

%d is for integer number

```
polygon = 'rectangle'  
length = 12  
width = 3.145
```

```
print "This polygon is a %s" % polygon  
print "the length of the %s is %d" %(polygon,length)  
print "the width of the %s is %f" %(polygon,width)
```

```
This polygon is a rectangle  
the length of the rectangle is 12  
the width of the rectangle is 3.145000
```

Data structures

1. List

Stores numbers, strings, lists, dictionaries, etc.

```
>>> List=[10.0,20,'float',100]
>>> print List
[10.0, 20, 'float', 100]
```

Add a new element to the list

```
>>> List.append(0.23)
>>> print List
[10.0, 20, 'float', 100, 0.23]
>>> List.insert(2,'InsertElement')
>>> print List
[10.0, 20, 'InsertElement', 'float', 100, 0.23]
```

List sequences

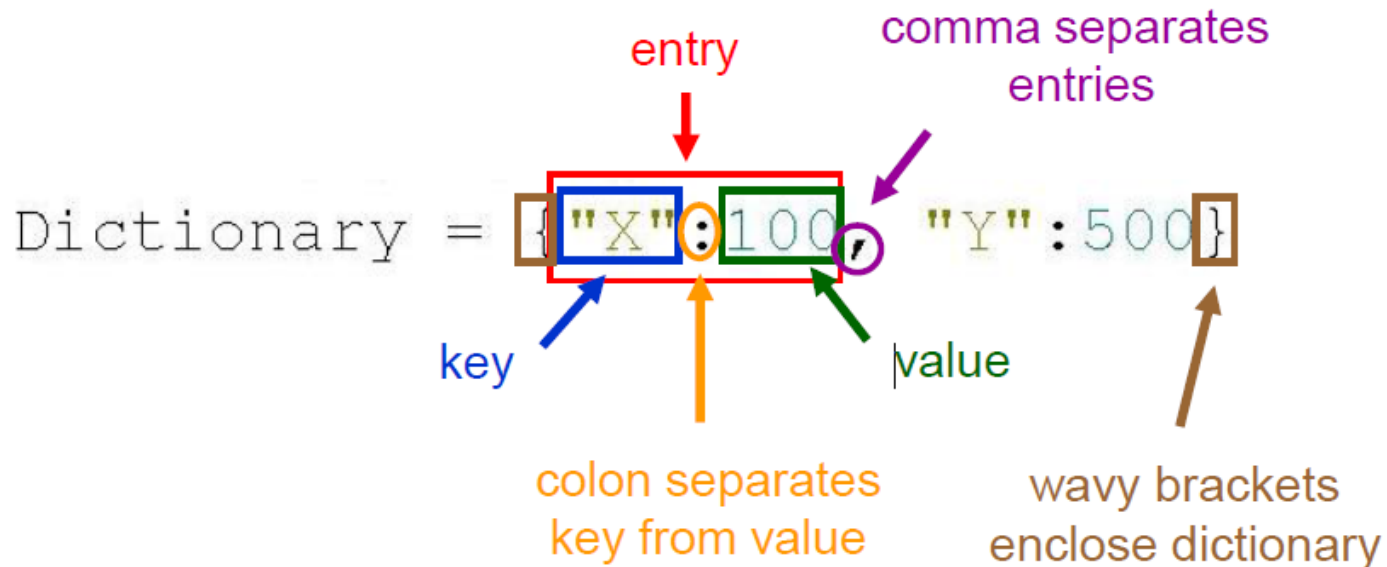
```
>>> print List[0],List[1],List[2],List[3],List[4]
10.0 20 float 100 0.23

>>> print List[-1],List[-2]    >>> print List[1:3]
0.23 100                       [20, 'float']
```

Data structures - list

2. Dictionary

Different with list, every element in a dictionary has two attributes - key and value

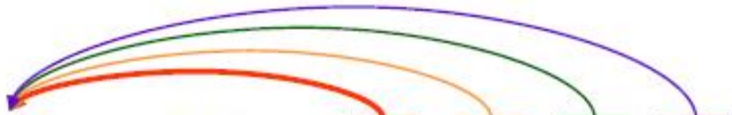


```
>>> a = {"X":10,"Y":20,"Z":30}
>>> print a
{'Y': 20, 'X': 10, 'Z': 30}
>>> print a{"X"}
SyntaxError: invalid syntax
>>> print a["X"]
10
>>>
```

For loop

for loop: example

```
for item in [1, 2, 3, 4] :  
    print item
```



Value of item	Output
item = 1	1
item = 2	2
item = 3	3
item = 4	4

For loop

For loop is used to do some repetitive operations.

```
# CALCULATE THE SUM OF 1 TO 100
num = range(101)
sum = 0
for i in num:
    sum = sum + i
print 'the sum of 1 to 100 is:', sum
```

