
Basic Python

ML4HEP 2026 Pre-school Lectures

Subir Sarkar, SINP

Lecture 3, 12/05/2026

Course Outline

- Python module, package, and namespace
 - `subprocess` and `pickle` modules
- String

Python Module

Python module, package, namespace

- A Python *module* is a file containing Python code
- A *package* is a collection of modules with a common purpose
- A project usually integrates different packages
- Python *namespace* is a combination of modules/packages

module

Let us try to model a simple bank account

```
# bacc.py
def account () :
    return {'balance': 0} # dictionary

def deposit (account, amount) :
    account['balance'] += amount
    return account['balance']

def withdraw (account, amount) :
    account['balance'] -= amount
    return account['balance']
```

A module in use

```
>>> from bacc import account, deposit, withdraw
>>> a = account()
>>> b = account()
>>> deposit(a, 100)
100
>>> deposit(b, 50)
50
>>> withdraw(b, 10)
40
>>> withdraw(a, 10)
90
```

A module w/ test code

```
def account():  
    return {'balance': 0} # dictionary  
  
def deposit(account, amount):  
    account['balance'] += amount  
    return account['balance']  
  
def withdraw(account, amount):  
    account['balance'] -= amount  
    return account['balance']  
  
def balance(account):  
    return account['balance']  
  
if __name__ == '__main__':  
    a = account()  
    b = account()  
    deposit(a, 1000)  
    deposit(b, 500)  
    withdraw(b, 100)  
    withdraw(a, 100)  
    print('Balance -> A/C a:', balance(a), 'A/C b:', balance(b))
```

int type

```
>>> a,b = 1,2
```

```
>>> type(a)
<class 'int'>
```

```
>>> type(b)
<class 'int'>
```

```
>>> c = a + b
```

```
>>> d = a.__add__(b) # operator overloading
```

```
>>> c,d
(3, 3)
```

```
>>> type(c), type(d)
(<class 'int'>, <class 'int'>)
```

Python String

String

- A string in Python is a sequence of characters
- Strings are **immutable**

```
a = "Higgs boson"  
b = 'aka, the God particle'  
c = """ # span multiple lines  
was discovered  
at the Large Hadron Collider  
10 years ago  
"""
```

quoted string

```
print("He asked, \"Which is your favorite team?\"")  
print('He asked, "Which is your favorite team?"')
```

String

```
>>> import string # load the module in memory
```

```
>>> dir(string)
```

```
[... 'atof', 'atof_error', 'atoi', 'atoi_error',  
'atol', 'atol_error', 'capitalize', 'capwords',  
'center', 'count', 'digits', 'expandtabs', 'find',  
'hexdigits', 'index', 'index_error', 'join',  
'joinfields', 'letters', 'ljust', 'lower',  
'lowercase', 'lstrip', 'maketrans', 'octdigits',  
'printable', 'punctuation', 'replace', 'rfind',  
'rindex', 'rjust', 'rsplit', 'rstrip', 'split',  
'splitfields', 'strip', 'swapcase', 'translate',  
'upper', 'uppercase', 'whitespace', 'zfill']
```

String object operations

```
>>> s = 'Hello World!'
```

```
>>> s.isdigit()
```

```
False
```

```
>>> s.find('W')
```

```
6
```

```
>>> s.rfind('W')
```

```
6
```

```
>>> s.endswith('W')
```

```
False
```

```
>>> s.startswith('Hello')
```

```
True
```

```
>>> s.split(" ")
```

```
['Hello', 'World!']
```

```
>>> s.upper()
```

```
'HELLO WORLD!'
```

```
>>> s.swapcase()
```

```
'hELLO wORLD!'
```

```
>>> s.replace('World', 'Globe')
```

```
'Hello Globe!'
```

```
>>> s = 'Hello\tWorld!'
```

```
>>> s
```

```
'Hello\tWorld!'
```

```
>>> s.expandtabs()
```

```
'Hello  World!'
```

String operations

Convert a string

'1,2,3,4,5,6,7,8'

to

'1:2:3:4:5:6:7:8'

String operations : split & join

```
>>> astr = '1,2,3,4,5,6,7,8'
```

```
>>> list = astr.split(',')
```

```
>>> list
```

```
['1', '2', '3', '4', '5', '6', '7', '8']
```

```
>>> ':' .join(list)
```

```
'1:2:3:4:5:6:7:8'
```

all in one line

```
>>> ':' .join('1,2,3,4,5,6,7,8'.split(','))
```

```
'1:2:3:4:5:6:7:8'
```

String Formatting [C style]

```
>>> print("Today's price: %f" % 50.4625)
```

```
Today's price: 50.462500
```

```
>>> print("Today's price: %.2f" % 50.4625)
```

```
Today's price: 50.46
```

```
>>> print("Change since yesterday: %+ .2f" % 1.5)
```

```
Change since yesterday: +1.50
```

String Formatting [Modern]

```
>>> print("{:.3f}".format(math.pi))
```

```
3.142
```

```
>>> print("{0:d} - {0:x} - {0:o} - {0:b}".format(21))
```

```
21 - 15 - 25 - 10101
```

```
>>> s = "I prefer {0} over {1}".format('emacs', 'vi')
```

```
>>> s = "Lang: {lang}".format(lang="Python")
```

```
# defining format
```

```
>>> email_f = "Your email was {email}".format
```

```
>>> print(email_f(email="bob@example.com"))
```