Python for Data Science

IISER-Kolkata ML4HEP Pre-school Lecture Series
May 14, 2025 - Lecture 5

Subir Sarkar, SINP
subir.sarkar@cern.ch

int type

```
>>> a,b = 1,2
>>> type(a)
<class 'int'>
>>> type(b)
<class 'int'>
>>> c = a + b
>>> d = a.__add__(b)
>>> c,d
(3, 3)
>>> type(c), type(d)
(<class 'int'>, <class 'int'>)
```

Module/Package

- 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

Idiomatic Python

Idiomatic Python

- Write better, more readable/maintainable code
- Use the language's strengths to the fullest extent

Python Idiom - swapping 2 numbers

Novice

```
>>> a = 5
>>> b = 7
>>> tmp = a
>>> a = b
>>> b = tmp
>>> a,b
7 5
```

Idiomatic

```
>>> a,b = 5,7
>>> b,a = a,b
>>> a,b
7 5
```

Python Idiom - List operation

```
>>> p_list = ['Perl', 'Python', 'Ruby']
```

Novice

```
>>> for i in range(len(p_list)):
...     print(p_list[i]), # C Way
...
Perl Python Ruby
>>> index = 0
>>> while index < len(p_list):
...     print (p_list[index])
...     index += 1
```

Idiomatic

```
>>> for item in p_list:
...     print(item)
>>> for (index,item) in enumerate(p_list):
...     print(index, item)
```

Python Idiom – dictionary from lists

```
>>> given = ['Sachin','Rahul','Sourav','VVS']
>>> family = ['Tendulkar','Dravid',
'Ganguly','Laxman']
```

Novice

```
>>> players = {}
>>> for (i, n) in enumerate(given):
...     players[n] = family[i]
...
>>> players
{'Sachin': 'Tendulkar', 'Rahul': 'Dravid', 'Sourav':
'Ganguly', 'VVS': 'Laxman'}
```

Idiomatic

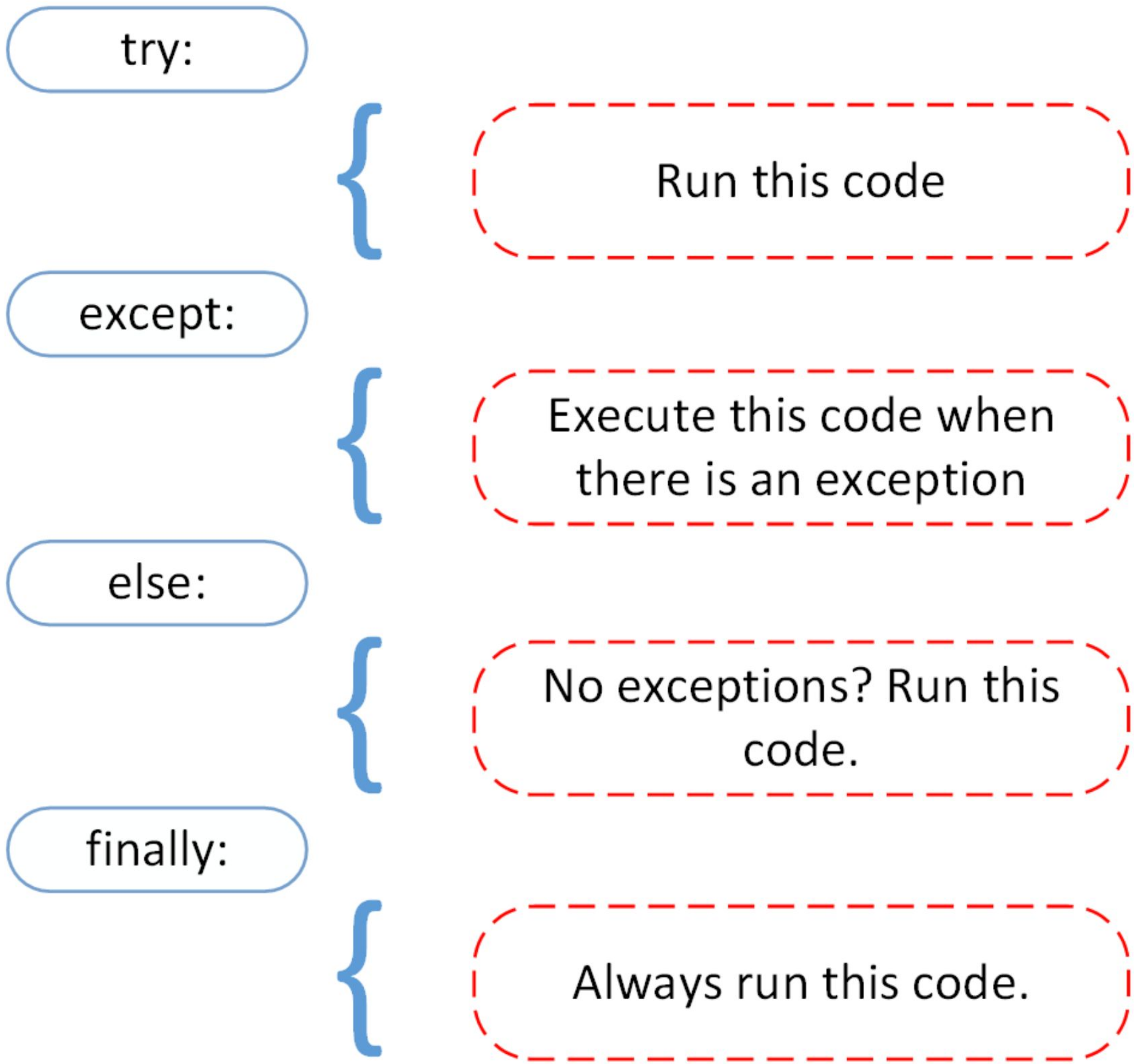
```
>>> players = dict(zip(given, family))
>>> players
{'Sachin': 'Tendulkar', 'Rahul': 'Dravid', 'Sourav':
'Ganguly', 'VVS': 'Laxman'}
```

Python Idiom: LBYL vs EAFP

```
import traceback
players = {'Sachin':'Tendulkar','Rahul': 'Dravid',
'Sourav': 'Ganguly', 'VVS': 'Laxman'}

name = 'Anil'

# Look Before You Leap (LBYL)
if name in players:
  f = players[name]
else:
  print 'key %s does not exist' % name

# Easier to Ask Forgiveness than Permission (EAFP)
try:
  f = players[name]
except KeyError:
  traceback.print_exc()
```

Exceptions in Python

https://www.datacamp.com/community/tutorials/exception-handling-python



Object Serialization

pickle – object serialization

```
#!/usr/bin/env python
import pickle as pk

a = {'key1': (1,2,3, 'world', [1,2,3]),
     'key2': {'b': 0, 'c': -1, 'd': 11}}
b = {'a': 1, 'b': 2}

f = open('store.pk', 'wb')
pk.dump(a,f)
pk.dump(b,f)
f.close()

f = open('store.pk', 'rb')
a = pk.load(f)
print a['key1']
print b
```

Python specific format

cPickle – speed up

```
try:
    import cPickle as pk
except:
    print('cPickle is not available!')
    import pickle as pk
a = {'key1': (1,2,3, 'world', [1,2,3]),
     'key2': {'b': 0, 'c': -1, 'd': 11}}
b = {'a': 1, 'b': 2}

f = open('store.pk', 'wb')
pk.dump(a,f)
pk.dump(b,f)
f.close()

f = open('store.pk', 'rb')
a = pk.load(f)
print (a['key1'])
print (b)
```