

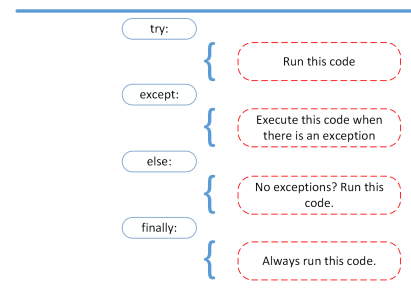


Exceptions

Ngoại lệ

Ngoại lệ là gì?

- ❑ Lỗi xảy ra trong lúc chạy chương trình: lỗi cú pháp, lỗi chia 0, lỗi biến/tập tin/hàm/object không tìm thấy, lỗi kiểu dữ liệu, v.v.
- ❑ Chương trình đang chạy sẽ sinh ra ngoại lệ khi một có lỗi xảy ra=> nếu không có đoạn **bắt ngoại lệ** nào trong module thì sẽ kết thúc và dừng chương trình. Ngoại lệ bên trong hàm, đối tượng có thể bắt ở bên ngoài do cơ chế ném ngoại lệ ra cấp cao hơn.
- ❑ Để chương trình có thể chạy tiếp khi có lỗi xảy ra thì cần đặt trong khối bắt ngoại lệ



Exception	Nguyên nhân gây ra exception
AssertionError	Trả về exception khi biểu thức so sánh của câu lệnh assert có kết quả là False
FloatingPointError	Xảy ra lỗi khi thao tác trên các giá trị số thực gặp lỗi
ImportError	Xảy ra khi import một module mà Python không tìm thấy
IndexError	Xảy ra khi truy xuất một phần tử có index nằm ngoài phạm vi cho phép
KeyError	Xảy ra khi một key không tìm thấy trong một Dictionary
KeyboardInterrupt	Khi người dùng gõ các interrupt key (Ctrl+C hoặc Delete)
MemoryError	Xảy ra khi bộ nhớ lưu trữ bị đầy
NameError	Xảy ra khi một biến (variable) không tìm thấy trong phạm vi (scope) cục bộ (local) hoặc toàn cục (global)
NotImplementedError	Xảy ra khi một phương thức trừu tượng (abstract method) của lớp cơ sở không được định nghĩa lại trong dẫn xuất kế thừa lớp cơ sở đó

Exception	Nguyên nhân gây ra exception
OSError	Đây là những lỗi hệ thống của hệ điều hành
OverflowError	Xảy ra khi kết quả của một số phép tính cho ra giá trị quá lớn để lưu trữ và biểu diễn
RuntimeError	Xảy ra lỗi là một exception không thuộc bất kỳ exception nào được xây dựng sẵn trong Python
SyntaxError	Xảy ra khi cú pháp của Python không đúng quy định
IndentationError	Xảy ra khi chúng ta sử dụng <u>indentation</u> không đúng quy định
TabError	Xảy ra khi <u>indentation</u> sử dụng tab hoặc khoảng cách không nhất quán
SystemError	Xảy ra khi trình thông dịch phát hiện một lỗi nội bộ (internal error) nào đó
TypeError	Xảy ra khi một phương thức (method) hoặc toán tử (operator) được áp dụng cho một đối tượng (object) có kiểu dữ liệu không phù hợp

Exception	Nguyên nhân gây ra exception
UnboundLocalError	Xảy ra khi gán giá trị (tham chiếu) cho một biến cục bộ trong một phương thức nhưng thật sự không có giá trị nào được biến đó tham chiếu đến
UnicodeError	Xảy ra khi mã hóa (encoding) hoặc giải mã (decoding) liên quan đến Unicode bị lỗi
ValueError	Một hàm có đối số truyền vào đúng kiểu dữ liệu nhưng giá trị của đối số lại không phù hợp
ZeroDivisionError	Chia một số cho 0

Bắt ngoại lệ

try:



Run this code

except:



Execute this code when
there is an exception

Bắt ngoại lệ

```
try:
    a="1"
    b=22
    x = 1/b
    y=a/2
    print(x)
except TypeError:
    print("Data type of variable is not suitable type!")
except ZeroDivisionError:
    print("Cannot divide by 0!")
except Exception as e:
    print("An exception occurred:",e)
print("still continue from here➡")
```



Bắt ngoại lệ

```
def function1():  
    a = int(input('Nhap so nguyen:'))  
    if a % 2 == 0:  
        print(a, " là số chẵn")  
    else:  
        print(A, " là số lẻ")  
  
    def f2():  
        pass  
    # Ngoại lệ xảy ra trong function1 bắt ở đây  
try:  
    function1()  
except Exception as e:  
    print("Có lỗi:", e)  
# cac lenh sau nua  
s = "python"  
print(s)
```



Bắt ngoại lệ

try:



Run this code

except:



Execute this code when
there is an exception

else:



No exceptions? Run this
code.

Bắt ngoại lệ

try:



Run this code

except:



Execute this code when
there is an exception

else:



No exceptions? Run this
code.

finally:



Always run this code.

Ta có thể ném 1 ngoại lệ nếu cần

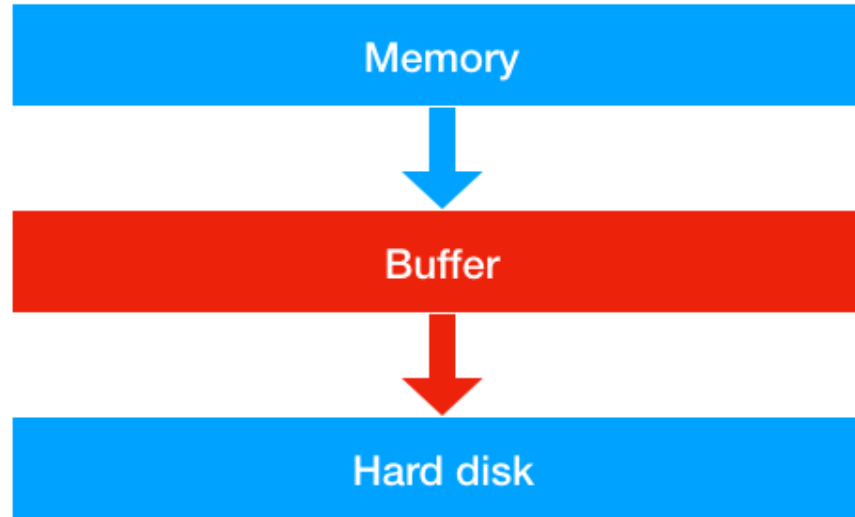
```
if condition:  
    raise Exception(string_message)
```

Nếu condition is True
→ Exception

```
assert condition, string_message
```

Nếu condition is False
→ AssertionError

Làm việc với tập tin



- ❑ Buffer (vùng nhớ đệm) là lớp trung gian khi dữ liệu bộ nhớ được ghi vào đĩa cứng → tăng tốc độ, giảm số lần ghi vào đĩa cứng.
- ❑ Ví dụ, khi chúng ta tải xuống một tập tin 500 MB, nếu chúng ta ghi 1MB một lần vào đĩa cứng, chúng ta phải ghi 500 lần. Nếu chúng ta ghi 50MB một lần một lần, chúng ta chỉ ghi 50 lần vào đĩa cứng.

```
file_object = open(file_name [, access_mode][, buffering])
```

Làm việc với tập tin

```
file_object = open(file_name [, access_mode][, buffering])
```

- ❑ `file_name` tên_tệp mà bạn muốn làm việc
- ❑ `access_mode` xác định chế độ mở tệp để làm gì?: đọc, ghi, thêm, v.v. mặc định là đọc (r)
- ❑ `buffering` : 0 là không dung bộ đệm; 1 bộ đệm dòng text cho text file; Số nguyên dương N, bộ đệm N byte; Số âm, kích thước bộ đệm là mặc định của hệ thống



Làm việc với tập tin

Modes	Description
r	Mở tệp chỉ được đọc dạng text. Con trỏ tệp được đặt ở đầu tệp, mặc định. Lỗi xảy ra nếu file không tồn tại
rb	Mở tệp chỉ để đọc ở định dạng nhị phân. Lỗi xảy ra nếu file không tồn tại
r+	Mở tệp đọc và ghi dạng text
rb+	Mở tệp đọc và ghi dạng nhị phân
w	Mở tệp để ghi dạng text, Ghi đè tệp nếu tệp tồn tại. Nếu tệp không tồn tại tạo mới để ghi.
wb	Mở tệp để ghi dạng nhị phân, Ghi đè tệp nếu tệp tồn tại. Nếu tệp không tồn tại tạo mới để ghi.
w+	Mở tệp để ghi và đọc dạng text. Ghi đè tệp nếu tệp tồn tại. Nếu tệp không tồn tại tạo mới để ghi.



Làm việc với tập tin

Modes	Description
wb+	Mở tệp để ghi và đọc dạng nhị phân. Ghi đè tệp nếu tệp tồn tại. Nếu tệp không tồn tại tạo mới để ghi.
a	Mở một tệp để thêm vào tệp dạng văn bản. Con trỏ tệp ở cuối tệp nếu tệp tồn tại. Nếu tệp không tồn tại, nó sẽ tạo một tệp mới để ghi.
ab	Mở một tệp để thêm vào tệp dạng nhị phân. Con trỏ tệp ở cuối tệp nếu tệp tồn tại. Nếu tệp không tồn tại, nó sẽ tạo một tệp mới để ghi.
a+	Mở một tệp để đọc và thêm vào tệp dạng văn bản. Con trỏ tệp ở cuối tệp nếu tệp tồn tại. Nếu tệp không tồn tại, nó sẽ tạo một tệp mới để ghi.
ab+	Mở một tệp để đọc và thêm vào tệp dạng nhị phân. Con trỏ tệp ở cuối tệp nếu tệp tồn tại. Nếu tệp không tồn tại, nó sẽ tạo một tệp mới để ghi.

Làm việc với tập tin

```
# Open a file
fo = open("foo.txt", "wb")
print ("Name of the file: ", fo.name)
print ("Closed or not : ", fo.closed)
print ("Opening mode : ", fo.mode)
fo.close()
```



Name of the file: foo.txt

Closed or not : False

Opening mode : wb



Làm việc với tập tin

```
# Open a file
fo = open("foo.txt", "wb")
print ("Name of the file: ", fo.name)
print ("Closed or not : ", fo.closed)
print ("Opening mode : ", fo.mode)
fo.close()
```



Name of the file: foo.txt

Closed or not : False

Opening mode : wb



Làm việc với tập tin

Phương thức	Mô tả
<code>fileObject.write(string)</code>	Ghi chuỗi vào file đang mở, không thêm ký tự xuống dòng
<code>fileObject.read([count])</code>	Đọc số lượng ký tự (count) nếu còn
<code>fileObject.tell()</code>	Trả về vị trí con trỏ file
<code>fileObject.flush()</code>	Ghi toàn bộ dữ liệu trong vùng đệm vào đĩa
<code>fileObject.close()</code>	Ghi toàn bộ dữ liệu trong vùng đệm vào đĩa và đóng tập tin

Nếu không gọi `close()` hay `flush()`, dữ liệu vùng đệm sẽ không ghi vào đĩa ➔ **mất dữ liệu**



Làm việc với tập tin

```
# Open a file
fo = open("foo.txt", "w")
fo.write( "Python is a great language.\nYeah its
great!!\n")
```

```
# Close opened file
fo.close()
```

```
# Open a file
fo = open("foo.txt", "r+")
str = fo.read(10)
print ("Read String is : ", str)
```

```
# Close opened file
fo.close()
```

→ Read String is : Python is



Làm việc với tập tin

```
with open("test.txt", 'w', encoding = 'utf-8') as f:  
    f.write("my first file\n")  
    f.write("This file\n\n")  
    f.write("contains three lines\n")
```



test - Notepad

File Edit Format View Help

my first file

This file

contains three lines

Làm việc với tập tin

```
# Open a file
fo = open("foo.txt", "r+")
str = fo.read(10)
print ("Read String is : ", str)
# Check current position
position = fo.tell()
print ("Current file position : ", position)
# Reposition pointer at the beginning once again
position = fo.seek(0, 0)
str = fo.read(10)
print ("Again read String is : ", str)
# Close opened file
fo.close()
```

➔ Read String is : Python is
Current file position : 10
Again read String is : Python is



Line buffering (text mode)

```
f = open('line_buffer.txt', 'w', buffering=1)
# text mode, line buffering
f.write('this is line\n')
# \n means we finish a line, so this line is written to
hard disk
f.write('this not a line')
# no \n, so it won't be written to hard disk
f.write('\n')
# with \n, write to hard disk
```

Unbuffered (binary mode)

```
In [10]: f = open('binary_unbuffered.txt', 'wb', buffering=0)
```

```
In [11]: binary_unbuffered.txt
```

```
In [13]: f.write(b'hello') # data will be written to hard disk no matter the size  
Out[13]: 5
```

```
In [14]: binary_unbuffered.txt  
hello
```

```
In [15]: f.write(b' world')  
Out[15]: 6
```

```
In [16]: binary_unbuffered.txt  
hello world
```



buffered (binary mode)

```
In [18]: f = open('binary_buffer_size.txt', 'wb', buffering=10)
```

```
#buffer size is 10 bytes
```

```
In [19]: binary_buffer_size.txt
```

```
In [21]: f.write(b'hello') # write 5 bytes data
```

```
Out[21]: 5
```

```
In [22]: binary_buffer_size.txt
```

```
In [23]: f.write(b'helloworld') # write more
```

```
Out[23]: 10
```

```
In [24]: binary_buffer_size.txt # buffer is full, write to hard disk  
hello
```

```
In [25]: f.write(b'helloworld111')
```

```
Out[25]: 13
```

```
In [26]: binary_buffer_size.txt
```

```
hellohelloworldhelloworld111
```



Tập tin csv

```
Name,Hire Date,Salary,Sick Days remaining
Graham Chapman,03/15/14,50000.00,10
John Cleese,06/01/15,65000.00,8
Eric Idle,05/12/14,45000.00,10
Terry Jones,11/01/13,70000.00,3
Terry Gilliam,08/12/14,48000.00,7
Michael Palin,05/23/13,66000.00,8
```

Các bước để viết tệp CSV Để ghi dữ liệu vào tệp CSV:

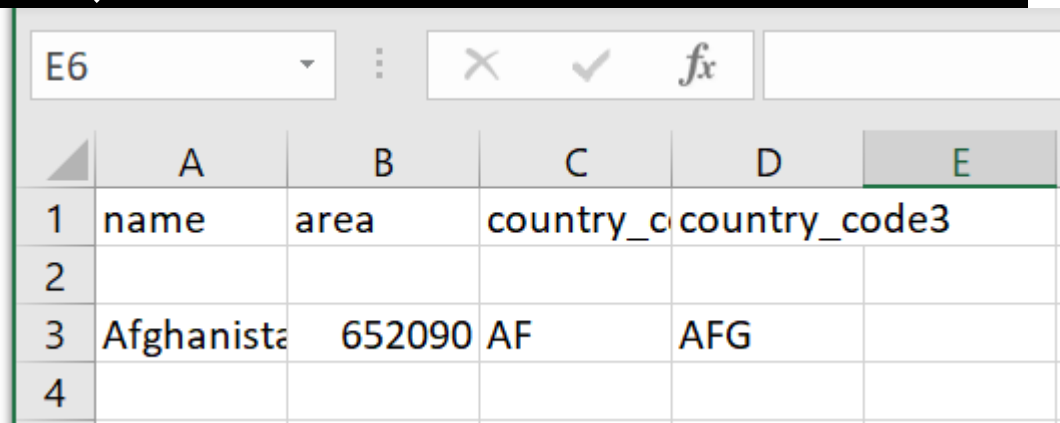
1. Mở tệp CSV để ghi (chế độ w) bằng cách sử dụng hàm open ().
2. Tạo CSV writer object (writer() function of the csv module)
3. Ghi dữ liệu vào tệp CSV bằng cách gọi phương thức writerow () hoặc writerows () của đối tượng
4. Đóng tệp sau khi bạn hoàn tất việc ghi dữ liệu

Tập tin csv

```
import csv

header = ['name', 'area', 'country_code2',
          'country_code3']
data = ['Afghanistan', 652090, 'AF', 'AFG']

with open('countries.csv', 'w', encoding='UTF8') as f:
    writer = csv.writer(f)
    # write the header
    writer.writerow(header)
    # write the data
    writer.writerow(data)
```



The screenshot shows a spreadsheet application window with a menu bar (E6, X, checkmark, fx) and a toolbar. The spreadsheet has columns A, B, C, D, and E. Row 1 contains the header: 'name', 'area', 'country_code2', 'country_code3'. Row 2 is empty. Row 3 contains the data: 'Afghanistan', '652090', 'AF', 'AFG'. Row 4 is empty.

	A	B	C	D	E
1	name	area	country_code2	country_code3	
2					
3	Afghanistan	652090	AF	AFG	
4					

Tập tin csv

```
import csv

header = ['name', 'area', 'country_code2', 'country_code3']
data = ['Afghanistan', 652090, 'AF', 'AFG']

with open('countries.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)

    # write the header
    writer.writerow(header)

    # write the data
    writer.writerow(data)
```

Tập tin csv

```
import csv

header = ['name', 'area', 'country_code2', 'country_code3']
data = [
    ['Albania', 28748, 'AL', 'ALB'],
    ['Algeria', 2381741, 'DZ', 'DZA'],
    ['American Samoa', 199, 'AS', 'ASM'],
    ['Andorra', 468, 'AD', 'AND'],
    ['Angola', 1246700, 'AO', 'AGO']
]

with open('countries.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)

    # write the header
    writer.writerow(header)

    # write multiple rows
    writer.writerows(data)
```

Tập tin csv

```
import csv

with open('countries.csv') as csv_file:
    csv_reader = csv.reader(csv_file,
                             delimiter=',')
    for row in csv_reader:
        print(row)
```

```
['name', 'area', 'country_code2', 'country_code3']
['Albania', '28748', 'AL', 'ALB']
['Algeria', '2381741', 'DZ', 'DZA']
['American Samoa', '199', 'AS', 'ASM']
['Andorra', '468', 'AD', 'AND']
['Angola', '1246700', 'AO', 'AGO']
```



Tập tin csv

```
import csv

with open('countries.csv') as csv_file:
    csv_reader = csv.reader(csv_file,
delimiter=',')
    next(csv_reader)
    for row in csv_reader:
        print(row)
```

```
['Albania', '28748', 'AL', 'ALB']
['Algeria', '2381741', 'DZ', 'DZA']
['American Samoa', '199', 'AS', 'ASM']
['Andorra', '468', 'AD', 'AND']
['Angola', '1246700', 'AO', 'AGO']
```



Tập tin csv

```
import csv

with open("countries.csv", "r") as csvfile:
    csvreader = csv.DictReader(csvfile)
    for row in csvreader:
        print(row["name"], row["area"],
row["country_code2"], row["country_code3"])
```

Albania 28748 AL ALB
Algeria 2381741 DZ DZA
American Samoa 199 AS ASM
Andorra 468 AD AND
Angola 1246700 AO AGO



Tập tin csv

```
import pandas
df = pandas.read_csv('countries.csv')
print(df)
```

```
C:\Users\dangq>python test.py
```

	name	area	country_code2	country_code3
0	Albania	28748	AL	ALB
1	Algeria	2381741	DZ	DZA
2	American Samoa	199	AS	ASM
3	Andorra	468	AD	AND
4	Angola	1246700	AO	AGO

```
import pandas
df = pandas.read_csv('countries.csv', skiprows=1)
print(df)
```

```
C:\Users\dangq>python test.py
```

	name	area	country_code2	country_code3
0	Algeria	2381741	DZ	DZA
1	American Samoa	199	AS	ASM
2	Andorra	468	AD	AND
3	Angola	1246700	AO	AGO



Tập tin xls

<https://xlsxwriter.readthedocs.io/>

```
# import pandas as pd
import pandas as pd
# Create some Pandas dataframes from some data.
df1 = pd.DataFrame({'Data': [11, 12, 13, 14]})
df2 = pd.DataFrame({'Data': [21, 22, 23, 24]})
df3 = pd.DataFrame({'Data': [31, 32, 33, 34]})
df4 = pd.DataFrame({'Data': [41, 42, 43, 44]})
# Create a Pandas Excel writer object using XlsxWriter as the
engine.
writer = pd.ExcelWriter('pandas_positioning.xlsx', engine
='xlsxwriter')
# write and Positioning the dataframes in the worksheet.
# Default position, cell A1.
df1.to_excel(writer, sheet_name='Sheet1')
df2.to_excel(writer, sheet_name='Sheet1', startcol = 3)
df3.to_excel(writer, sheet_name='Sheet1', startrow = 6)
# It is also possible to write the dataframe without the header
and index.
df4.to_excel(writer, sheet_name='Sheet1', startrow = 7, startcol
= 4, header = False, index = False)
# Close the Pandas Excel writer object and output the Excel file.
writer.save()
```

Tập tin docx

<https://python-docx.readthedocs.io/en/latest/>

```
# import docx NOT python-docx
import docx
# create an instance of a word document
doc = docx.Document()
# add a heading of level 0 (largest heading)
doc.add_heading('Heading for the document', 0)
# add a paragraph and store the object in a variable
doc_para = doc.add_paragraph('Your paragraph goes here, ')
# add a run i.e, style like bold, italic, underline, etc.
doc_para.add_run('hey there, bold here').bold = True
doc_para.add_run(', and ')
doc_para.add_run('these words are italic').italic = True
# add a page break to start a new page
doc.add_page_break()
# add a heading of level 2
doc.add_heading('Heading level 2', 2)
# pictures can also be added to our word document
# width is optional
doc.add_picture('path_to_picture')
# now save the document to a location
doc.save('path_to_document')
```

Tập tin json

- ❑ JSON (JavaScript Object Notation)
- ❑ Dữ liệu trong JSON được lưu trữ theo cặp thuộc tính (attribute) – giá trị (value)
- ❑ Truyền dữ liệu Client/Server, tập tin cấu hình .v.v



```
order1.json  + X
Schema: <No Schema Selected>
1  {
2    "size": "medium",
3    "price": 15.67,
4    "toppings": ["mushrooms", "pepperoni", "basil"],
5    "extra_cheese": false,
6    "delivery": true,
7    "client": {
8      "name": "Nguyen Van Ty",
9      "phone": null,
10     "email": "nvty@email.com"
11   }
12 }
```

Tập tin json

```
import json

with open('order1.json', 'r') as f:
    data = json.load(f)

print(type(data))
print(data)
print(data["client"]["name"])
```

```
<class 'dict'>
{'size': 'medium', 'price': 15.67, 'toppings': ['mushrooms',
'pepperoni', 'basil'], 'extra_cheese': False, 'delivery': True, 'client':
{'name': 'Nguyen Van Ty', 'phone': None, 'email':
'nvty@email.com'}}
Nguyen Van Ty
```



```
import json
# python object(dictionary) to be dumped
dict1 = {
    "emp1": {
        "name": "Lisa",
        "designation": "programmer",
        "age": "34",
        "salary": "54000"
    },
    "emp2": {
        "name": "Elis",
        "designation": "Trainee",
        "age": "24",
        "salary": "40000"
    },
}

# the json file where the output must be stored
out_file = open("myfile.json", "w")
json.dump(dict1, out_file, indent = 6)
out_file.close()
```



Tập tin json

Python	JSON
dict	object
list, tuple	array
str	string
int, float, int- & float-derived Enums	number
True	true
False	false
None	null

Lưu object ra file

<https://joblib.readthedocs.io/en/latest/generated/joblib.dump.html>

- ❑ Lưu đối tượng ra file lưu trữ (Persist object)

```
import joblib
joblib.dump(object_name, filename, compress=0,
protocol=None, cache_size=None)
```

- ❑ Tạo đối tượng từ file lưu trữ đối tượng (Persist object)

```
import joblib
joblib.load(filename, mmap_mode=None)
```

WARNING: `joblib.load` relies on the `pickle` module and can therefore execute arbitrary Python code. It should therefore never be used to load files from untrusted sources.

```
import joblib
class Student:
    # constructor
    def __init__(self, name):
        self.name = name
        self.courses = []

    def show(self):
        print('Hello, my name is', self.name)
        print(self.courses)
    def add_course(self, subject, mark):
        self.courses.append((subject, mark))
#####
student1 = Student("Nguyễn Văn Tý")
student1.add_course("Python", 10)
student1.add_course("C++", 9)
student1.add_course("Database", 9.5)
student1.show()
joblib.dump(student1, 'student.pk')
student2=joblib.load('student.pk')
student2.show()
```

```
Hello, my name is Nguyễn Văn
Tý
[('Python', 10), ('C++', 9),
('Database', 9.5)]
Hello, my name is Nguyễn Văn
Tý
[('Python', 10), ('C++', 9),
('Database', 9.5)]
```



Q & A

Thank you!

