Python: The Easy Way

Lecture 3

Modules

To make your code more modular

Intro

module1.py

module2.py

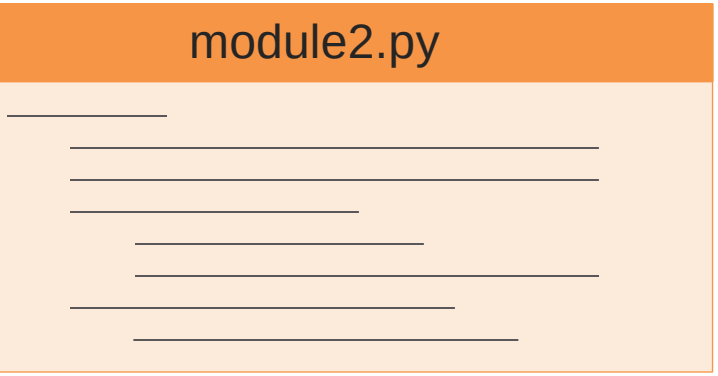
module3.py

Intro

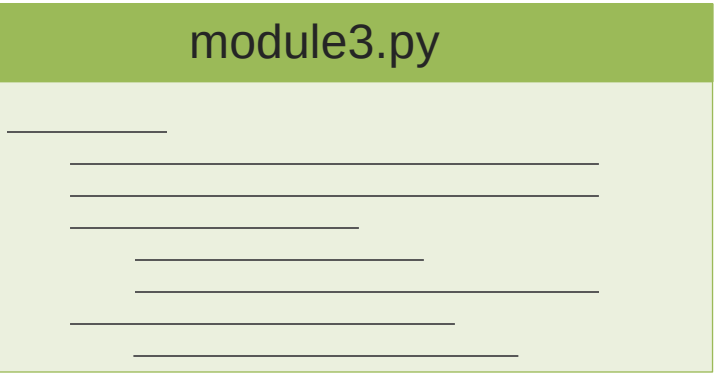
project.py

module1.py

module2.py



module3.py



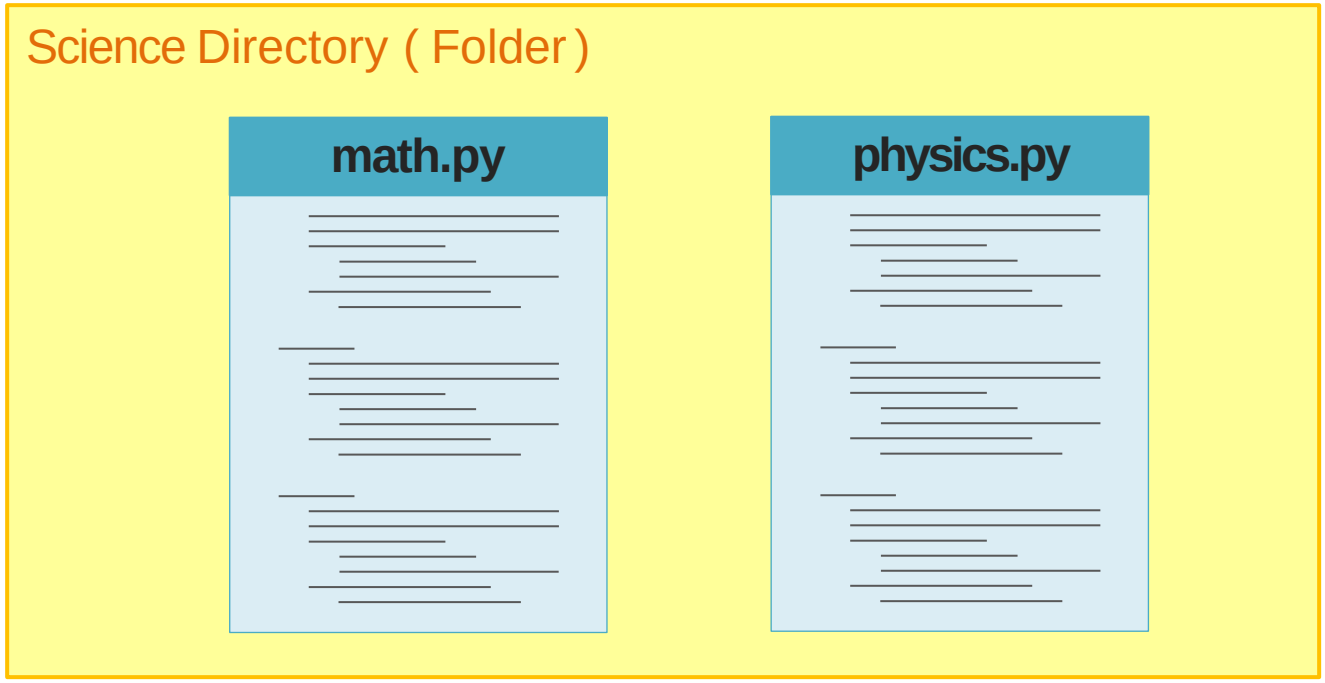
How To

```
from module_name import block_name
```

i.e. from math import tan

Packages

```
from pkge_name.module_name import block_name
```

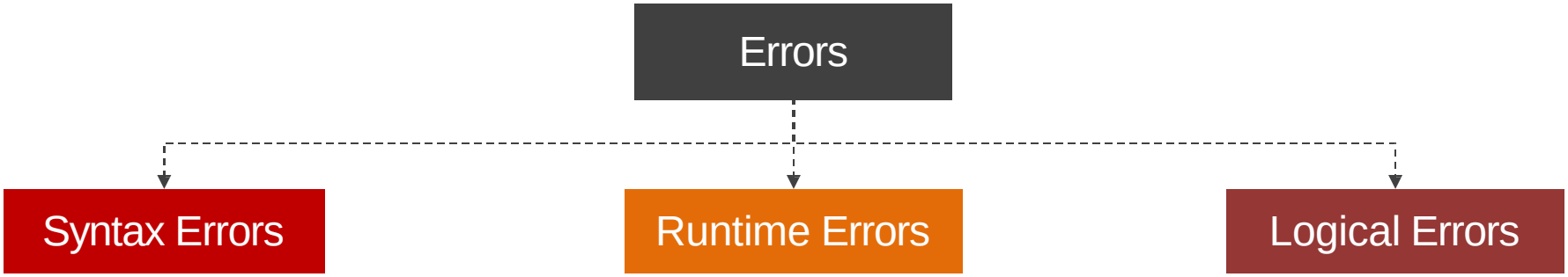


i.e. from science.math import tan

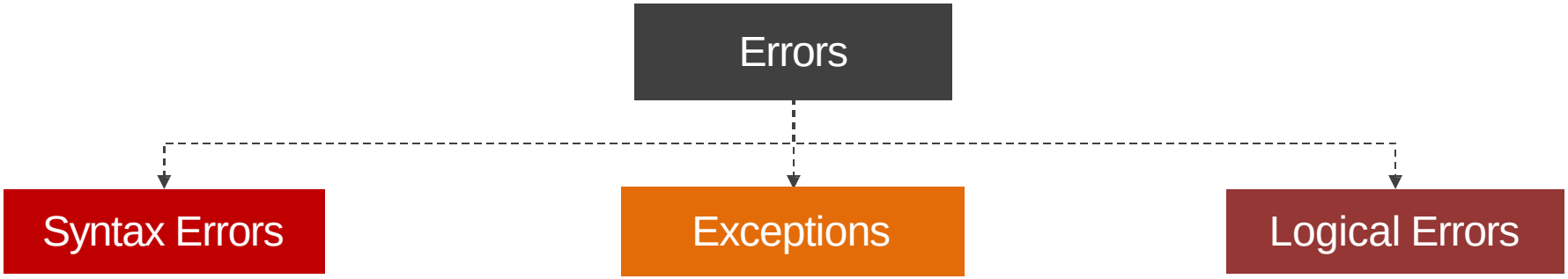
Errors & Exceptions

Gotta catch ‘em all

Intro



Intro



Syntax Errors

Errors that will show up if you doesn't follow Python Syntax Rules

```
print("You missed the closing round braces "
```

```
print("You missed the closing round braces "
                                            ^
SyntaxError: invalid syntax
```

Exceptions

Errors detected during execution are called **Exceptions**

```
print(firstname);
```

```
NameError: name 'firstname' is not defined
```

Handling Exceptions

```
try:
    doTry()
except:
    doExcept()
finally:
    doFinally()
```

Put the code that you want to handle its exceptions

Handle the exception if it raised in the try clause

Put the code that you want to run always if there is an exception or not.

Handling Exceptions

```
try:
        for i in range(3):
            print(3/i)
except ValueError:
        print(“Value Error")
except ZeroDivisionError:
        print("you divided by 0")
finally:
        print("this will print no matter what")
```

Raising Exceptions

```
raise ErrorName(error_message)
```

i.e. raise NameError(“It’s Not a name”)

File Input & Output

File Authoring

Open Files

```
open(file_name, mode)
```

mode	Job description
r	Open Files for reading only
w	Open Files for writing only *
a	Open Files for appending *
r+	Open Files for reading and writing *
rb	Open Files for reading binary files
rb+	Open Files for reading and writing binary files *

* If the file not exist , It will create it.

Read Files

```
fl = open("some_file.txt", 'r')
fl.read()
#output: Some text on line 1.
         Other text on line 2.
fl.read(4)
#output: Some
fl.readline()
#output:  text on line 1.
fl = open("some_file.txt", 'r')
for line in f1:
       print(line)
#output: Some text on line 1.
         Other text on line 2.
```

some_file.txt

Some text on line1.

Other text on line2.

Write on Files

```
fl = open("some_file.txt", 'w')
```

some_file.txt

Some text on line 1.

Other text on line 2.

Write on Files

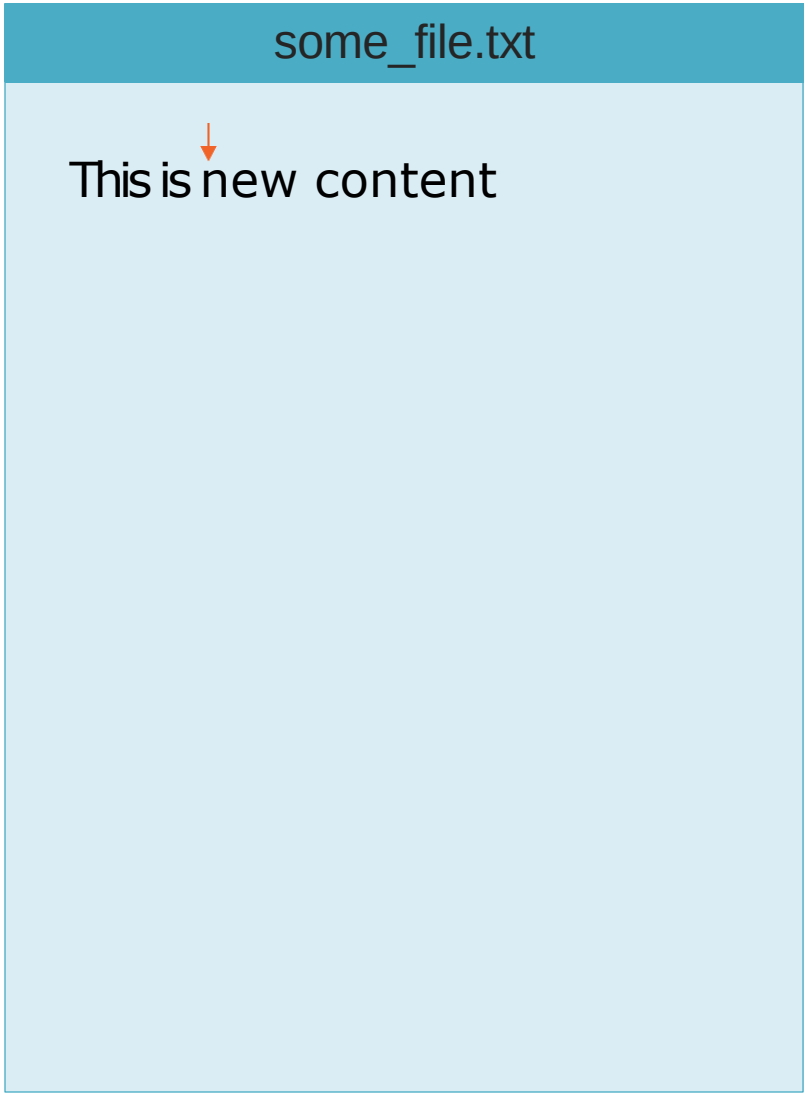
```
fl = open("some_file.txt", 'w')
fl.write("This is new content")
```

some_file.txt

This is new content

Write on Files

```
fl = open("some_file.txt", 'w')
fl.write("This is new content")
fl.seek(8)
```



Write on Files

```
fl = open("some_file.txt", 'w')
fl.write("This is new content")
fl.seek(8)
fl.write("old")
```

some_file.txt

This is old content

Write on Files

```
fl = open("some_file.txt", 'w')
fl.write("This is new content")
fl.seek(8)
fl.write("old")
fl.close()
fl = open("some_file.txt", 'a')
fl.write("\n content is appended")
```

some_file.txt

This is old content

content isappended

Python Standard Library

OS

os module provides functions for interacting with the operating system

```
import os

os.getcwd()                          # /usr/bin/python33

os.system("rmdir dir2")              # it will remove dir2

os.chdir("/home/ahmedmoawad")        # change the dir. to /home/…

os.getlogin()                        # "Ahmed Moawad"
```

math

math module provides access to the mathematical functions by the C standard

```
import math

math.ceil(3.2)          # 4

math.floor(3.6)         # 3

math.sqrt(9)            # 3

math.pi                 # 3.14
```

re

re provides regular expression matching operations

```
import re

re.match(pattern,string)

#match string with pattern from its starting

re.fullmatch(pattern,string)

#match full string with the pattern

re.search(pattern,string)

#scan the string finding the part that match the pattern
```

External Libraries

pip tool

pip

pip is a package management system used to install and manage software packages written in Python

```
pip install "some library"
```

i.e. pip install libcloud

Tips and Tricks

join and split

```
":".join(["1","Ali","grp"])     # colon is the separator
# '1:Ali:grp'
" ".join("ITI")                 # space is the separator
# 'I T I'
"Sara Mohamed".split(" ")       # space is the delimiter
# ["Sara" , "Mohamed"]
"django:flask".split(":")       # colon is the delimiter
# ["django" , "flask"]
```

is vs ==

```
True == 1                    # True

True is 1                    # False

list1 = [1,2,3]

list2 = [1,2,3]

list1 == list2               # True

list1 is list2               # False
```

Thank You