Errors

Table of Contents

1

Introduction

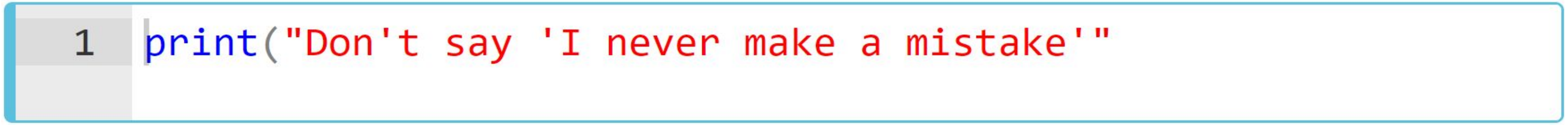
What can you say about this lesson?

Summarize what you've learned from pre-class content.

Students, write your response!

Introduction (review)

- Sometimes you can see that the codes with the simplest syntax can give an error when you never expected.
- Consider the following example :



```
print("Don't say 'I never make a mistake'"
```

What is the output? Try to figure out in your mind…

Students, write your response!

Introduction (review)

- Sometimes you can see that the codes with the simplest syntax can give an error when you never expected.
- Consider the following example :

```python
print("Don't say 'I never make a mistake'"
```

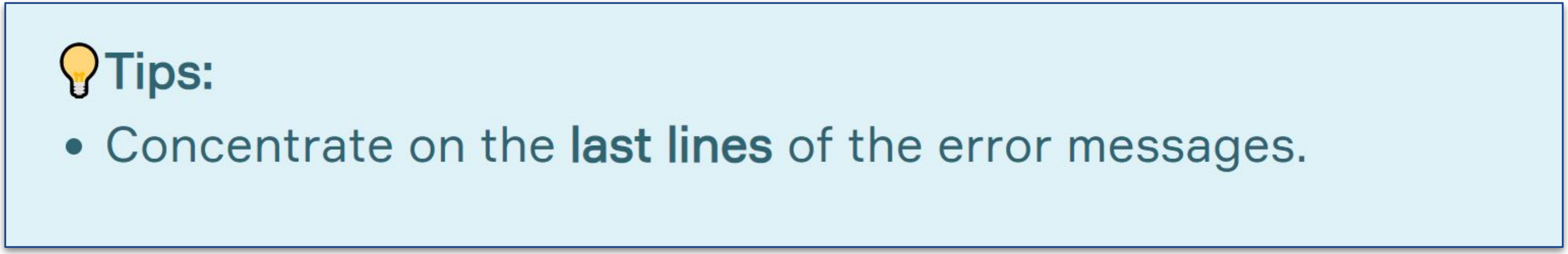
```
Traceback (most recent call last):
File "code.py", line 1
    print("Don't say 'I never make a mistake'"
                                              ^
SyntaxError: unexpected EOF while parsing
```

Introduction (review)

- In the error message appeared on the screen, there is the term **Traceback**.
- It is actually **a module** of *614 lines of Python code*.
- This module provides a standard interface to extract, format and print stack traces of Python programs.

Introduction (review)

- It exactly mimics the behavior of the Python interpreter when it prints a stack trace.
- In this way, it allows you to follow the line and character of the error and trace it.

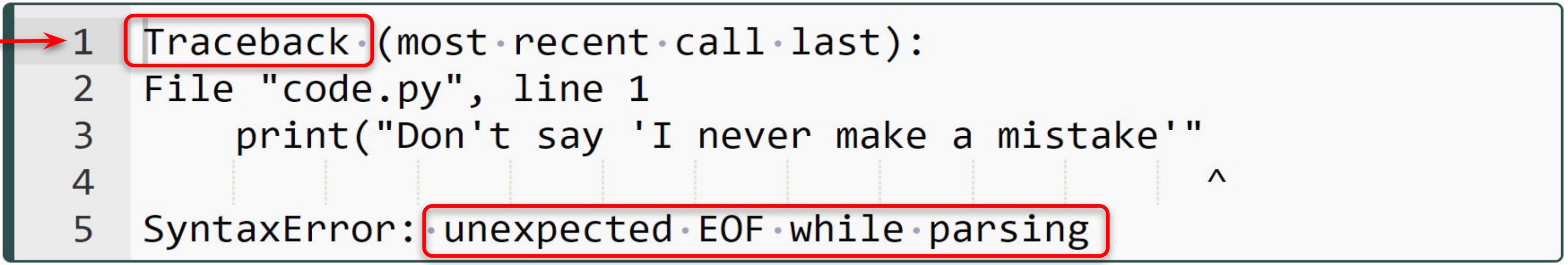


💡**Tips:**

- Concentrate on the **last lines** of the error messages.

Introduction (review)

- The name of module - ***Traceback*** - appears when your code causes an error and it reports detailed information on that specific error, demonstrating the particular files in which the error occurred.

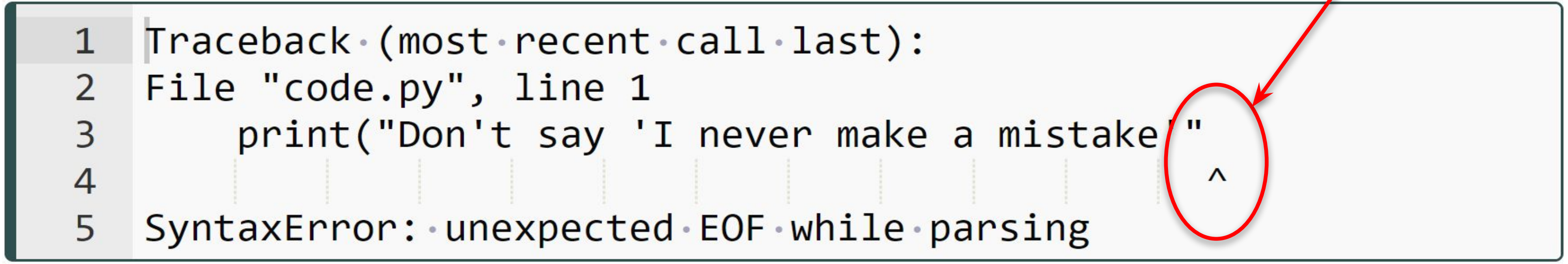


```
Traceback (most recent call last):
File "code.py", line 1
    print("Don't say 'I never make a mistake'"
                                               ^
SyntaxError: unexpected EOF while parsing
```

- In these error messages, the most important thing that a programmer should be interested in is **the last lines** in the most cases.

Introduction (review)

- In this example, the last two lines indicate that this error type is a Syntax error and it also indicates in which line and in which character (with the ^ sign) the error raised.



```
Traceback (most recent call last):
File "code.py", line 1
    print("Don't say 'I never make a mistake'"
                                              ^
SyntaxError: unexpected EOF while parsing
```

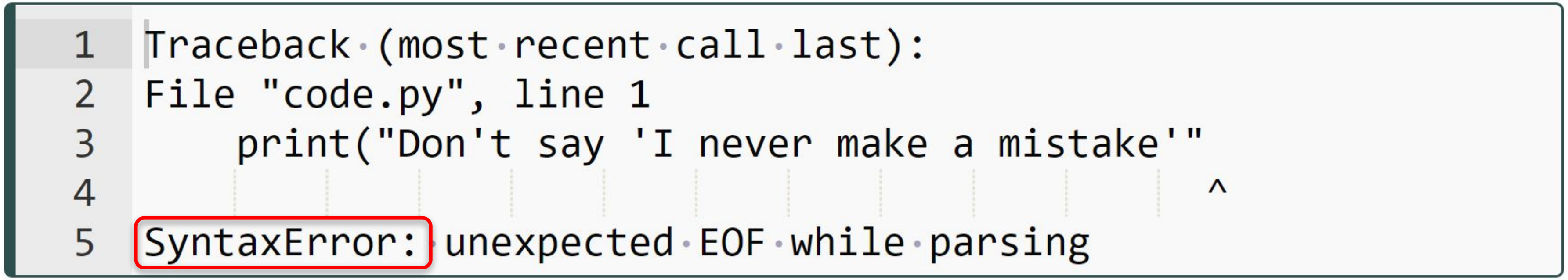
⚠ **Attention:**

- Do not panic when you see those error lines. Do not hesitate to read carefully what they are saying to you.

2 Syntax Errors

Syntax Errors (review)

- In the previous example (shown below), you must have seen the mysterious word **SyntaxError**, which you will likely encounter frequently during your time in Python.

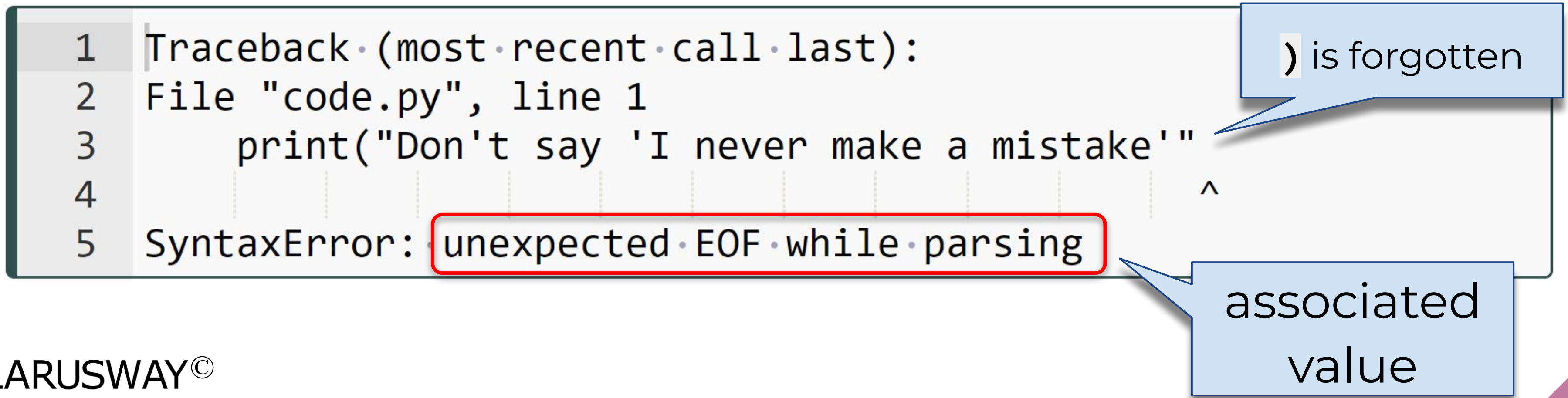


```
Traceback (most recent call last):
File "code.py", line 1
    print("Don't say 'I never make a mistake'"
                                              ^
SyntaxError: unexpected EOF while parsing
```

- A **wide variety of errors** in Python are called `SyntaxError`. Typically, they indicate a problem that Python encountered when trying to compile your program, or that your code could not be run.

Syntax Errors (review)

- Every syntax error has a text value (**known as associated value**) that describes the error in detail.
- In this example, the message "**SyntaxError: unexpected EOF while parsing**" means that something else had been expected by the interpreter after your statement, but you didn't pass it to the interpreter.



```
Traceback (most recent call last):
File "code.py", line 1
    print("Don't say 'I never make a mistake'"
                                              ^
SyntaxError: unexpected EOF while parsing
```

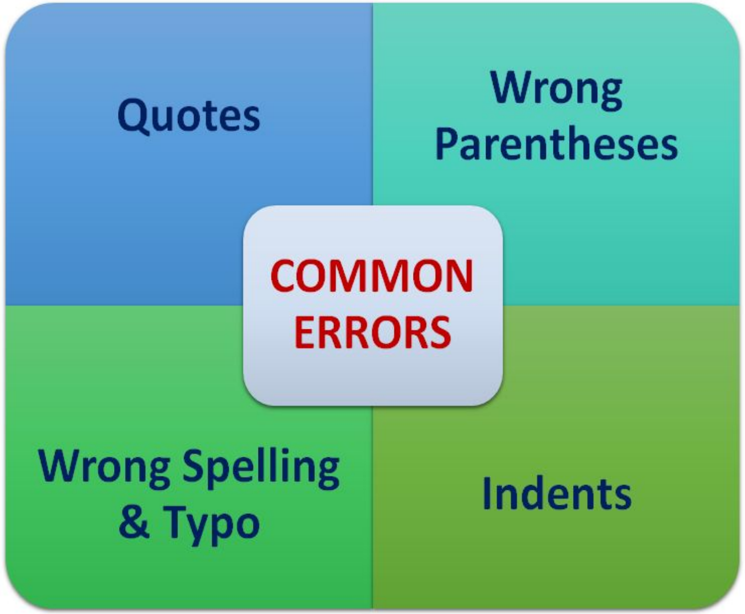
3

Common Errors

Summarize what you've learned from pre-class content about common errors:

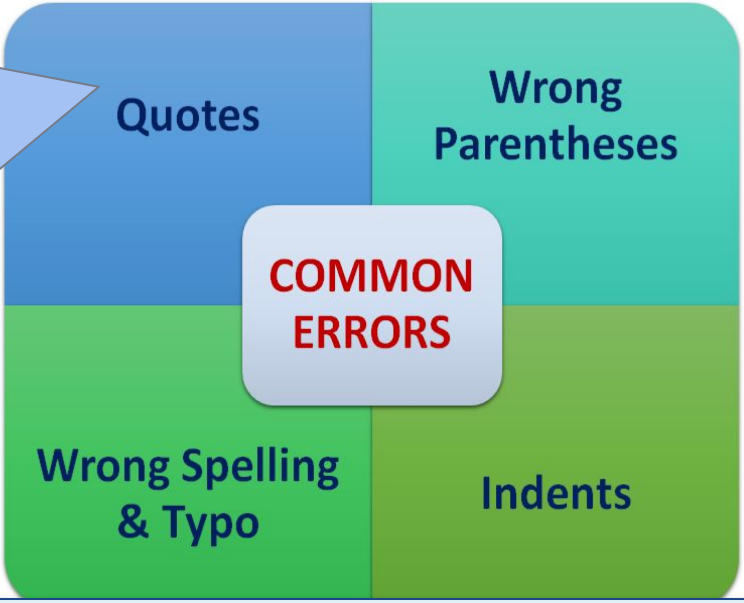
Students, write your response!

Common Errors (review)



Common Errors (review)

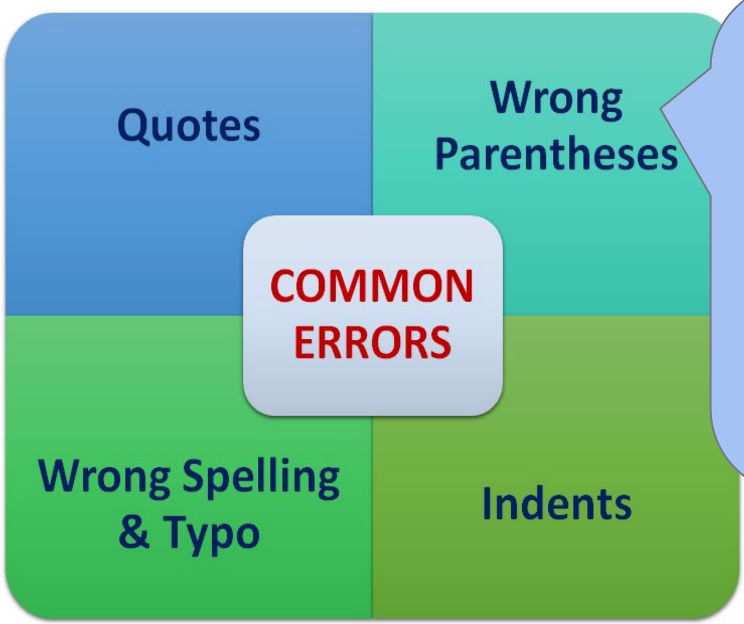
In the **Matter of Quotes** lesson, we have elaborated on how sensitive programming language Python is to **quotation** marks. So it's critical not to forget to enclose a string in quotes of the same type.



💡**Tips:**

- Keep this simple advice in your mind; triple quotes for multi-line strings, double or single quotes for ordinary strings.

Common Errors (review)

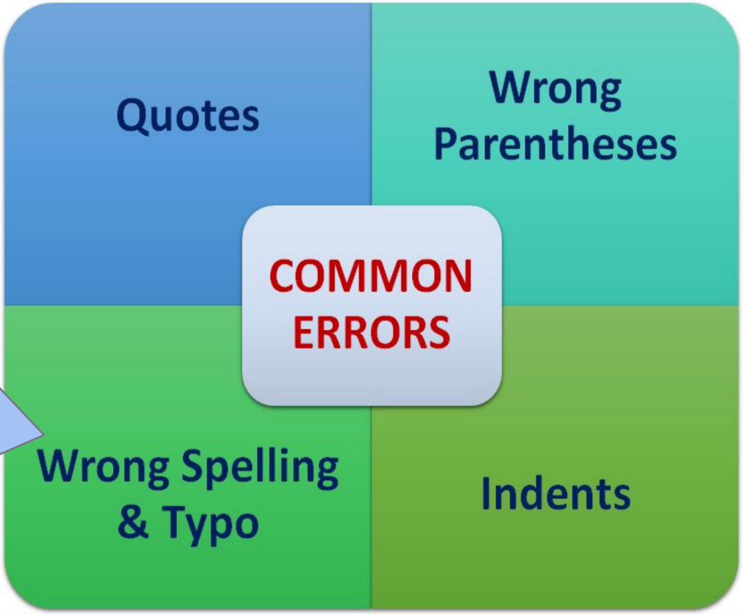


One of the most common syntax errors is the wrong number of parentheses in function calls. e.g.

```
print(len(my_list[2:])
```

Common Errors (review)

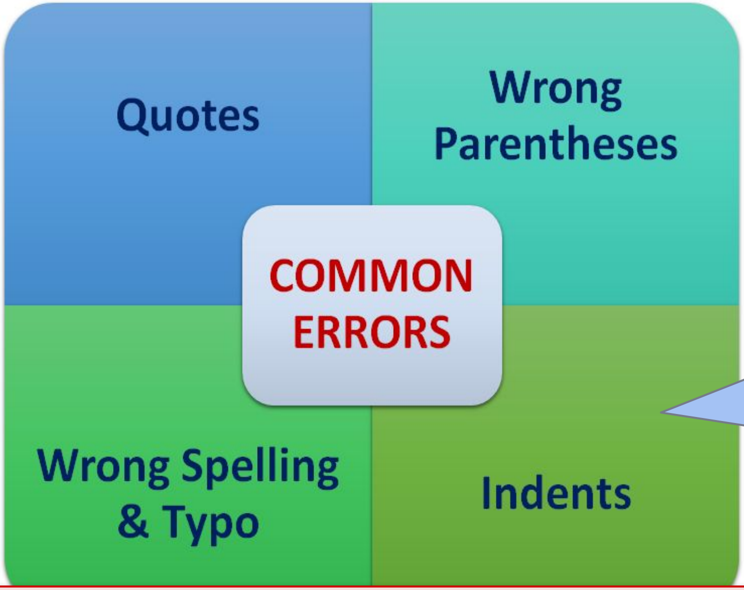
Yes, it may sound strange to you, but the **most common mistake** made by programmers is the wrong spelling keywords, function names, and variable names. e.g. `True` and `true`, `print` and `prit` or `pirnt`.



⚠ **Avoid ! :**

- Do not confuse uppercase and lowercase letter of the keywords. Keep in your mind that Python is a case-sensitive programing language.

Common Errors (review)



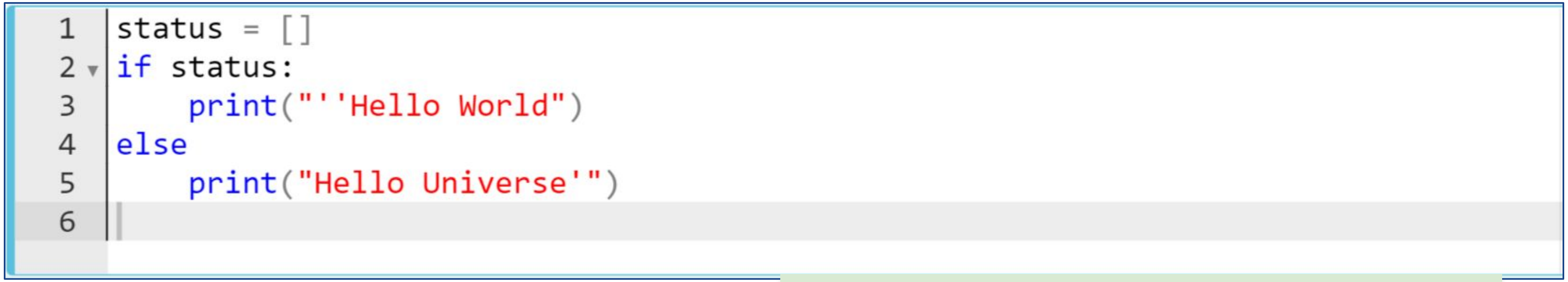
Indents are also very common errors for programmers.

⚠ **Avoid ! :**

- Do not forget to put the appropriate indent where necessary. Keep in your mind that Python is a indent-sensitive programing language.

Common Errors

- Let's find some errors in the codes :

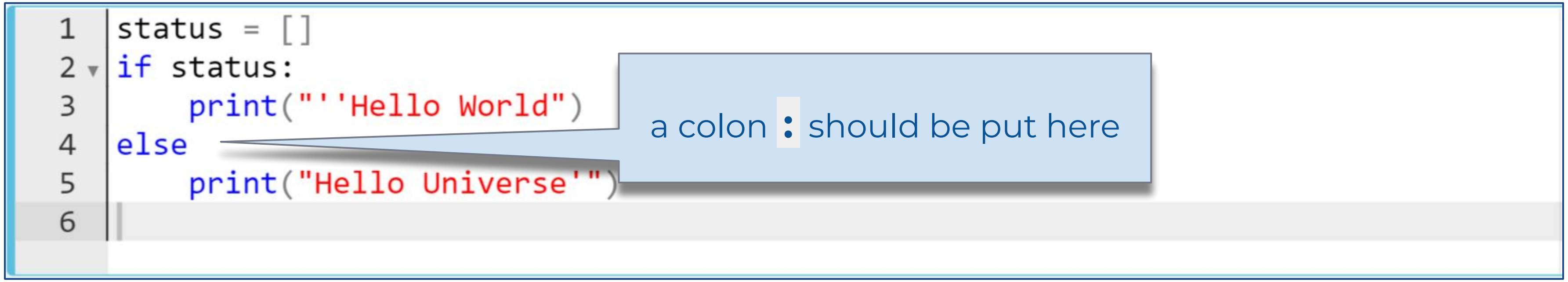


```
status = []
if status:
    print("''Hello World")
else
    print("Hello Universe''")

```

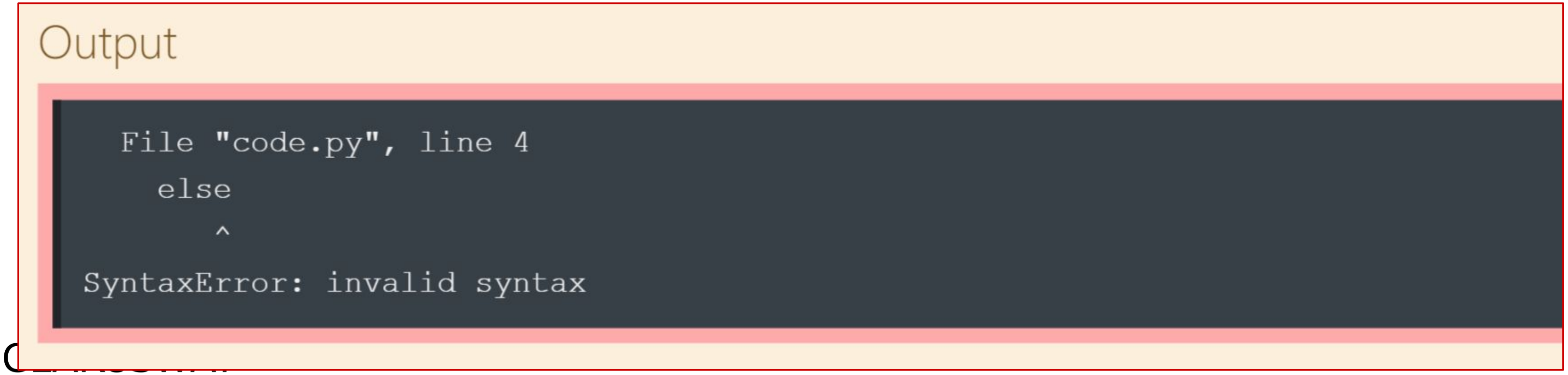
What is the error? Try to figure out in your mind...

Common Errors

- There is a typo (missing colon) :



```
1  status = []
2  if status:
3      print("''Hello World")
4  else
5      print("Hello Universe''")
6  
```

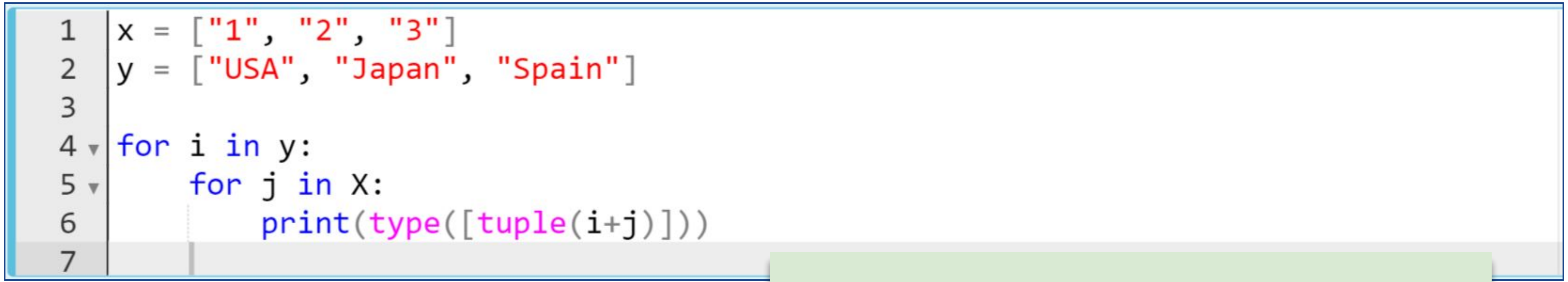


Output

```
   File "code.py", line 4
     else
        ^
 SyntaxError: invalid syntax
```

Common Errors

- Let's find some errors in the codes :

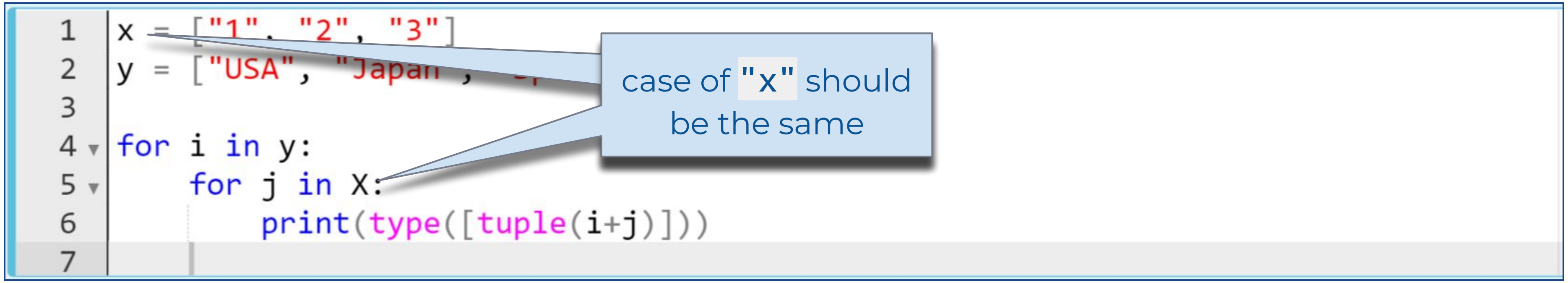


```
x = ["1", "2", "3"]
y = ["USA", "Japan", "Spain"]

for i in y:
    for j in X:
        print(type([tuple(i+j)]))

```

What is the error? Try to figure out in your mind...

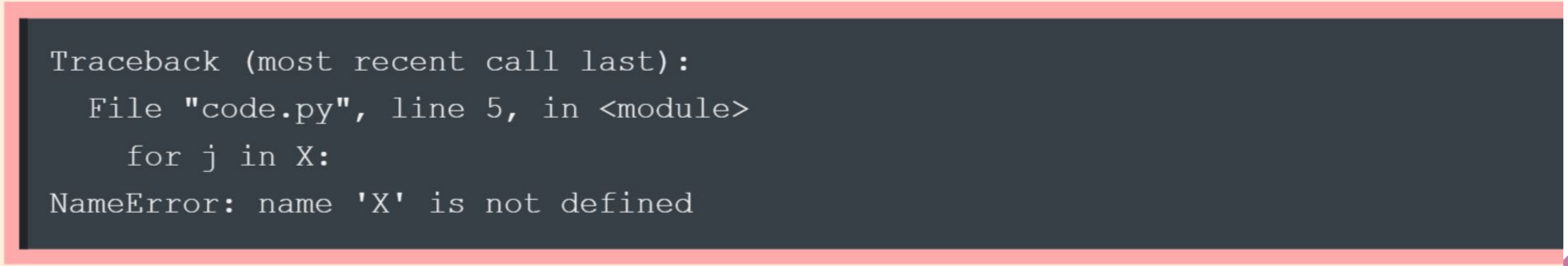
Common Errors

- Remember, Python is a case-sensitive language :



```python
x = ["1", "2", "3"]
y = ["USA", "Japan",

for i in y:
    for j in X:
        print(type([tuple(i+j)]))

```

Output



```
Traceback (most recent call last):
  File "code.py", line 5, in <module>
    for j in X:
NameError: name 'X' is not defined
```

THANKS!

Any questions?

You can find me at:

- @joseph
- joseph@clarusway.com