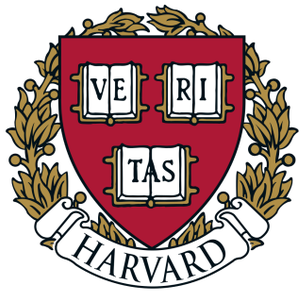


Lecture 4 - Becoming self sustainable at programming - II

Spandan Madan

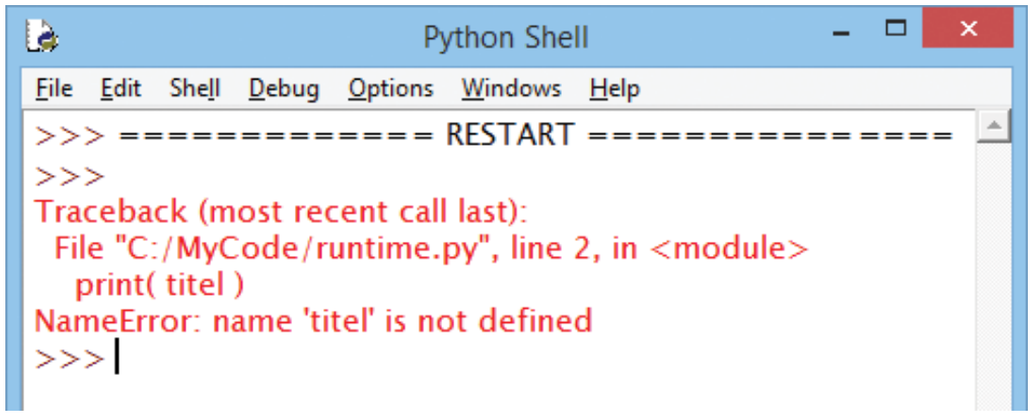
The guiding principle behind these 4 lectures

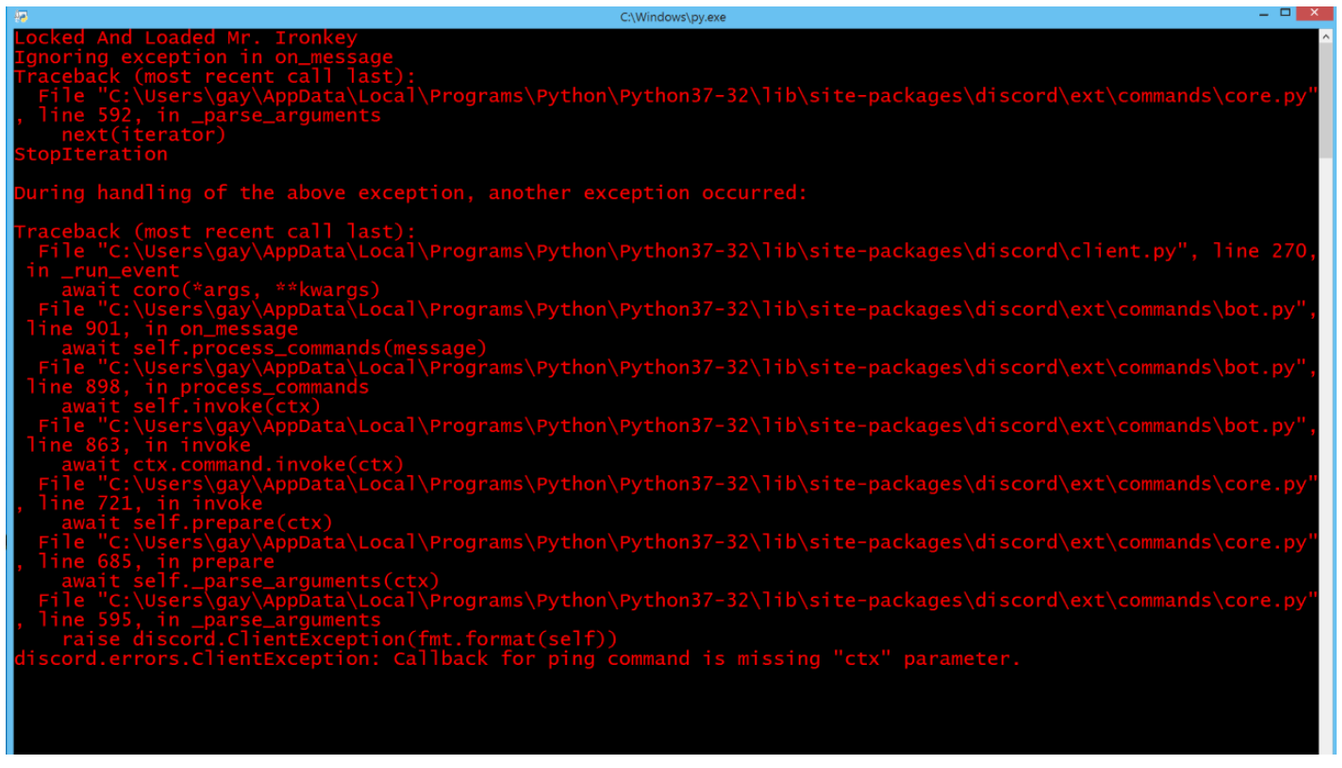
- Becoming self sustainable (Level 0): Learn the basics. The building blocks of python. Lists, Functions, Dictionaries, File read/write, Print, Objects, so on.

- Becoming self sustainable (Level 1): Learning to learn, by searching online, reading other people's code.

- Becoming self sustainable (Level 2): Adapting it for your use. Finding errors, making it work for your use.

Errors



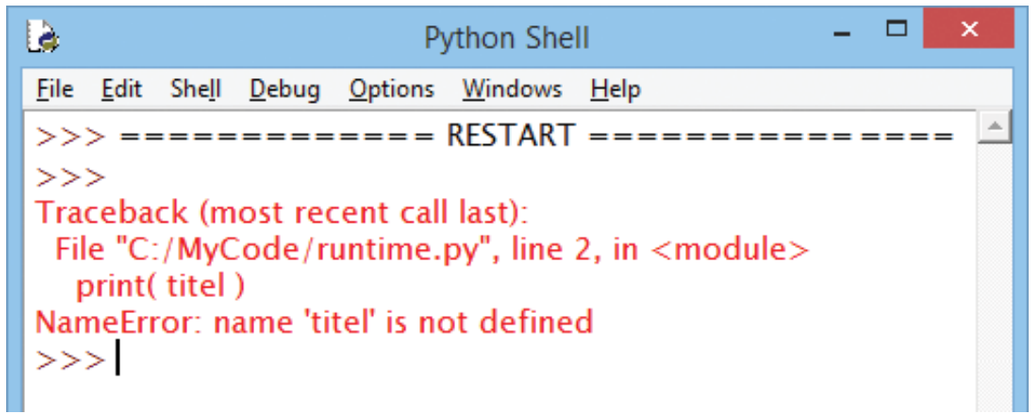



```
root@kali:~/Desktop/python# python3 py.py
Input 1 for password (0 for exit) : 1
Traceback (most recent call last):
  File "py.py", line 7, in <module>
    print(random.shuffle(password,password))
  File "/usr/lib/python3.4/random.py", line 277, in shuffle
    j = _int(random() * (i+1))
TypeError: 'str' object is not callable
```

```
~/projects/py/Environments/my_env 35s
my_env ❯ python example.py
Traceback (most recent call last):
  File "example.py", line 5, in <module>
    say('Michael')
  File "example.py", line 3, in say
    print('Hello, ' + nam)
NameError: name 'nam' is not defined
```

Errors are scary

$$scariness \propto redness$$



$$ease \propto scariness$$

titel —> title

Debugging

- Finding bugs in your code when there’s an error.
- Checking the logic of the code.
- Sanity checks
- Tests
- try-except

How to debug in python

- Print every little detail.
- Check each line for unexpected behavior
- try-except : left as an exercise

Common Error types

- SyntaxError
- IndexError, KeyError
- NameError
- TypeError
- IndentationError
- ModuleNotFoundError, ImportError

ModuleNotFoundError

- When you import a module you haven’t installed.

pip install <package_name>

- What is pip?
- Where do you run pip?
- How do you run it in your notebook?
- Must you run it in a notebook, though?

Going beyond colab notebook

- Linux users - congratulations, your life is easiest.
- Mac users - congratulations, your life will at least seem easy.
- Windows users - welcome to hell…

…for the first 2 days.

Installing python on your own computer

- linux, mac: open terminal app. Enter python.
- Windows users : install online, as easy as installing any other software :)
- https://realpython.com/installing-python/
- https://www.liquidweb.com/kb/install-pip-windows/

Where do I type everything?

- You write your code in an “editor” like Atom, Sublime Text (free to download). Notepad if you would like a death by syntax errors.

- Run your code in a “shell” - command prompt on windows, terminal on Mac/Linux.

Demo

- Go into python interpreter
- Quickly play around with files
- Run a python script

How does pip work?

- Everything is a file. A file's path must be given.
- Adding something to PATH (where should shell looks for things).
- Once pip is installed and added to PATH:

pip install <package_name>

- Modules are files too. Stored somewhere where you "import" them from.

ModuleNotFoundError

- When you import a module you haven’t installed.

pip install <package_name>

SyntaxError

- Missing commas, brackets, quotes
- = vs ==
- Mispelling name ex: pritn instead of print
- Look for line number, ^ symbol

```
print "hello"
np.sqrt(np.mean(x)
```

IndexError, KeyError

```
a=[1,2,3]

print(a[0])
>>1

print(a[1])
>>2

print(a[3])
```

```
country_number = {}
country_number['India':1]
country_number['USA':2]


print(country_number['India'])
>> 1

print(country_number['Canada'])
```

TypeError

```
a = 'hello'
b = 'bye'
c = 1

a+b
>> 'hellobye'

a+c
>> TypeError
```

IndentationError

- Python knows where things begin and end based on indentation - VERY IMPORTANT!

```
employee_name = 'ABC'
for i in range(10):
    print(i)
    print(employee_name)
```

```
for i in range(10):
    print(i)
print(employee_name)
```

NameError

```
def get_num_square(num)
    return num * num

print(get_num_squared)
```

```
x_var = 1
print(x)
```

Variable Scope

```
num_list = [12,311,3546]
def number_range(num_list):
    max_num = max(num_list)
    min_num = min(num_list)

    num_range = max_num - min_num
    return num_range

print(num_range)
```

```
>>> num_list = [12,311,3546]
>>> def number_range(num_list):
...     max_num = max(num_list)
...     min_num = min(num_list)
...
...     num_range = max_num - min_num
...     return num_range
...
>>> print(num_range)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'num_range' is not defined
>>>
```

- Variables declared in a function can’t be used outside. Unless passed appropriately.

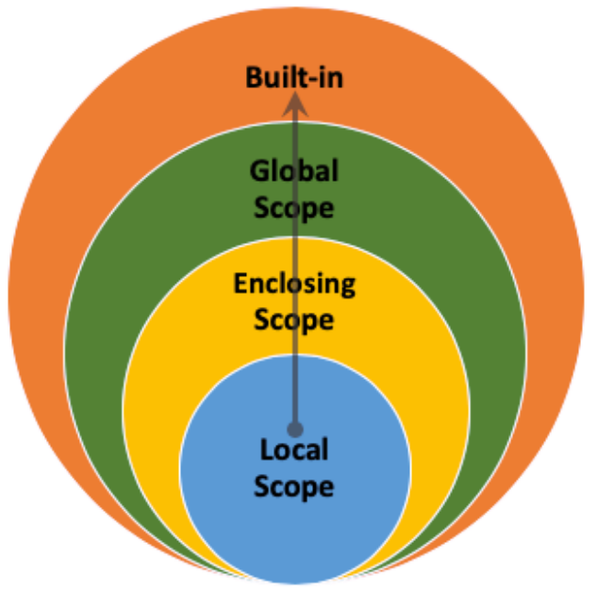
```
numbers_list = [12,311,3546]
def number_range(num_list):
    max_num = max(num_list)
    min_num = min(num_list)

    num_range = max_num - min_num
    return num_range

returned_num_range = number_range(numbers_list)
print(returned_num_range)
```

```
>>> numbers_list = [12,311,3546]
>>> def number_range(num_list):
...     max_num = max(num_list)
...     min_num = min(num_list)
...
...     num_range = max_num - min_num
...     return num_range
...
>>> returned_num_range = number_range(numbers_list)
>>> print(returned_num_range)
3534
```

Scope (access) hierarchy



- Local: Made in function, available in function ONLY.
- Enclosing: Nested functions
- Global: global x = 1
- Built in: sorted(), max(), float()

```
# Global scope
x = 0

def outer():
    # Enclosed scope
    x = 1
    def inner():
        # Local scope
        x = 2
```

Source: https://www.datacamp.com/community/tutorials/scope-of-variables-python

```python
def outer():
    first_num = 1
    def inner():
        second_num = 2
        # Print statement 1 - Scope: Inner
        print("first_num from outer: ", first_num)
        # Print statement 2 - Scope: Inner
        print("second_num from inner: ", second_num)
    inner()
    # Print statement 3 - Scope: Outer
    print("second_num from inner: ", second_num)

outer()
```

Source: https://www.datacamp.com/community/tutorials/scope-of-variables-python

Let’s debug

Conclusion

- Try setting up python and pip on your own computer - So that you can migrate out of colab over time.

- print, debug, print more.

Final Project

- This week - No assignment.
- Next week - No class.
- 2 weeks for revision + finalizing your project idea. Please make a 5 slide ppt on your project idea by June 6th.
- General Idea : Using what you learned for building something useful in your own work.