

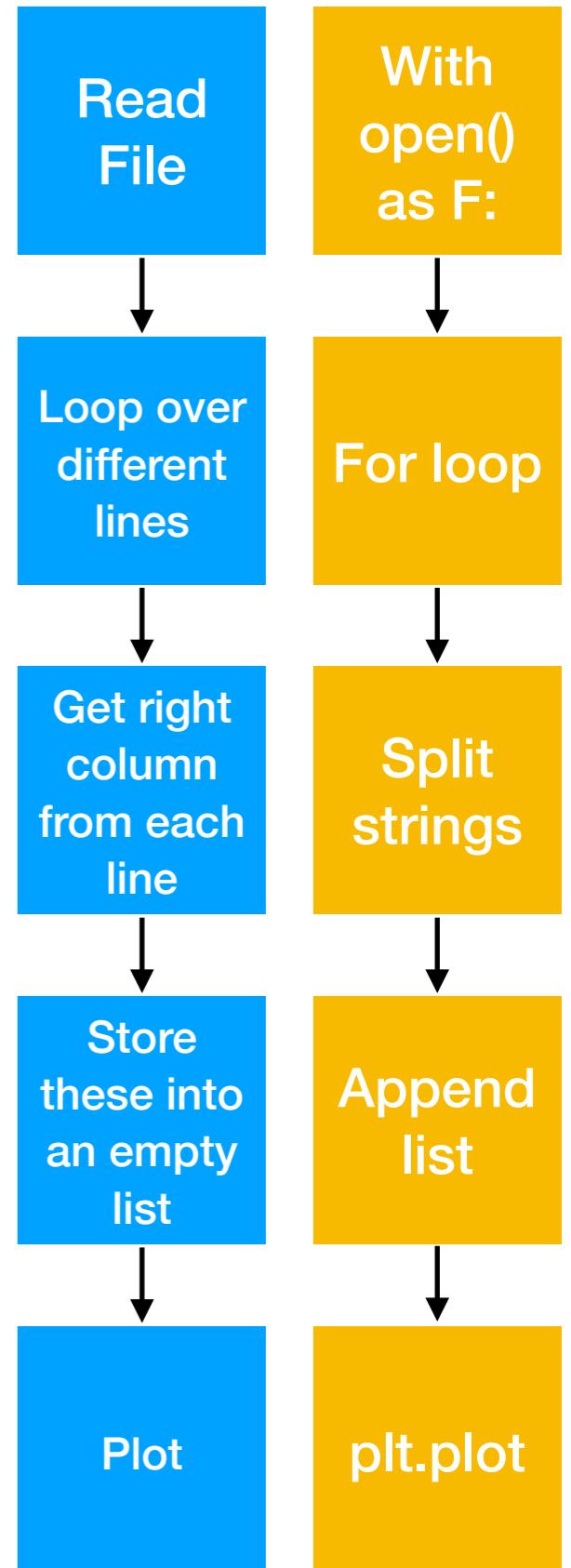
Putting it all together

Spandan Madan

Thinking Computationally

“First, solve the problem. Then, write the code.” – John Johnson

- What blocks do I need (**objects**), and how do I put them together (**algorithm**).
- Think this BEFORE going into the details of the blocks.



Important elements in Python programming

- Variables ($a = 1, b = 2$)
- Variables are mostly objects - lists, dicts, and so on.
- Operators ($a+b, a-b$)
- In-built functions - sorted()
- Modules
- custom functions
- If-else : Controlling the flow of code I
- Loops : Controlling the flow of code II

Variable Data Types

- Numbers. Come with operations like +,-,*,/
- Strings. Come with functions like split(), replace()
- Booleans : True or False. Often used in if-else conditions.
- Type casting : converting type. Ex: float('1') = 1.0

We saw a lot of objects

- Lists (called arrays in other languages)
- Tuples : *immutable* lists, you probably won't need to use them.
- Dictionaries
- File handler objects
- Sets
-

Making decisions based on some condition - if else

```
a = 1000
if a < 2000:
    print('less than 2000')

>> 'less than 2000'
```

These conditions are booleans - True or False

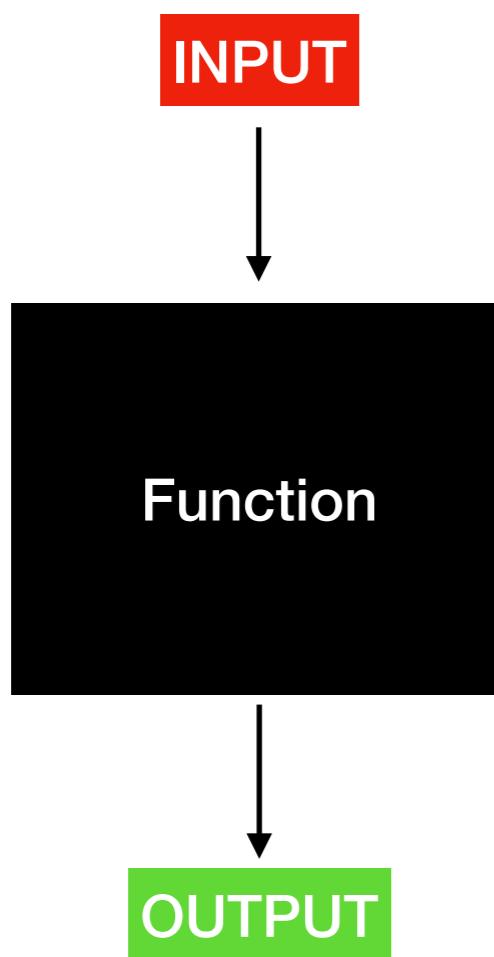
We saw combinations : Lists + for loop + if else + dictionary

```
# Finding employees with salary > $4,000 per month
# Dictionary employee_salaries already given to us

more_than_4000 = []
less_than_4000 = []

for name in employee_salaries.keys():
    salary = employee_salaries[name]
    if salary > 4000:
        more_than_4000.append(name)
    else:
        less_than_4000.append(name)
```

Custom Functions

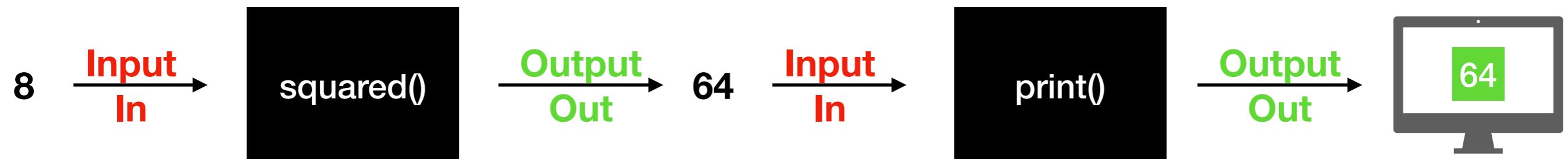


```
def pythagorus(side_1,side_2):
    hyp_squared = squared(side_1) + squared(side_2)
    hypotenuse = np.sqrt(hyp_squared)
    return hypotenuse

print(pythagorus(6,8))
>> 10
```

Nested Functions

`print(squared(64))`



FUNCTION CALL STACK

Nested For Loops

```
prefixes = ['multi', 'uni']
words = ['form', 'dimensional', 'lingual', 'lateral']

for i in prefixes:
    for j in words:
        print('%s-%s'%(i, j))
```

```
multi-form
multi-dimensional
multi-lingual
multi-lateral
uni-form
uni-dimensional
uni-lingual
uni-lateral
```

While loop

```
for a in range(10):  
    print(a)
```

Automatic increment

```
a = 0  
while(a<10):  
    print(a)  
    a += 1
```

increment yourself

$a = a + 1$

Python 3.6

```
1 prefixes = ['multi','uni']
2 words = ['form','dimensional','lingual','lateral']
3
4 for i in prefixes:
5     for j in words:
6         print('%s-%s'%(i,j))
```

[Edit this code](#)

→ line that just executed
→ next line to execute

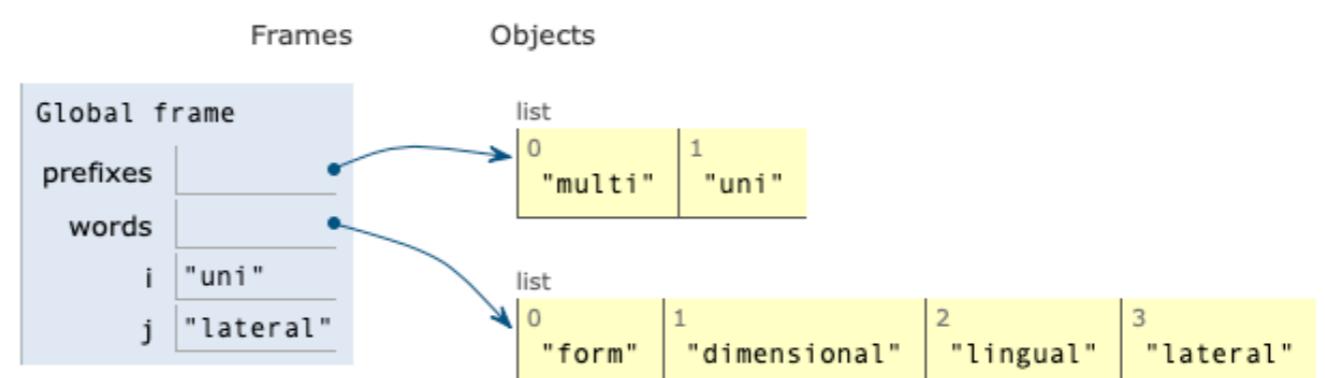
<< First < Prev Next > Last >>

Step 14 of 23

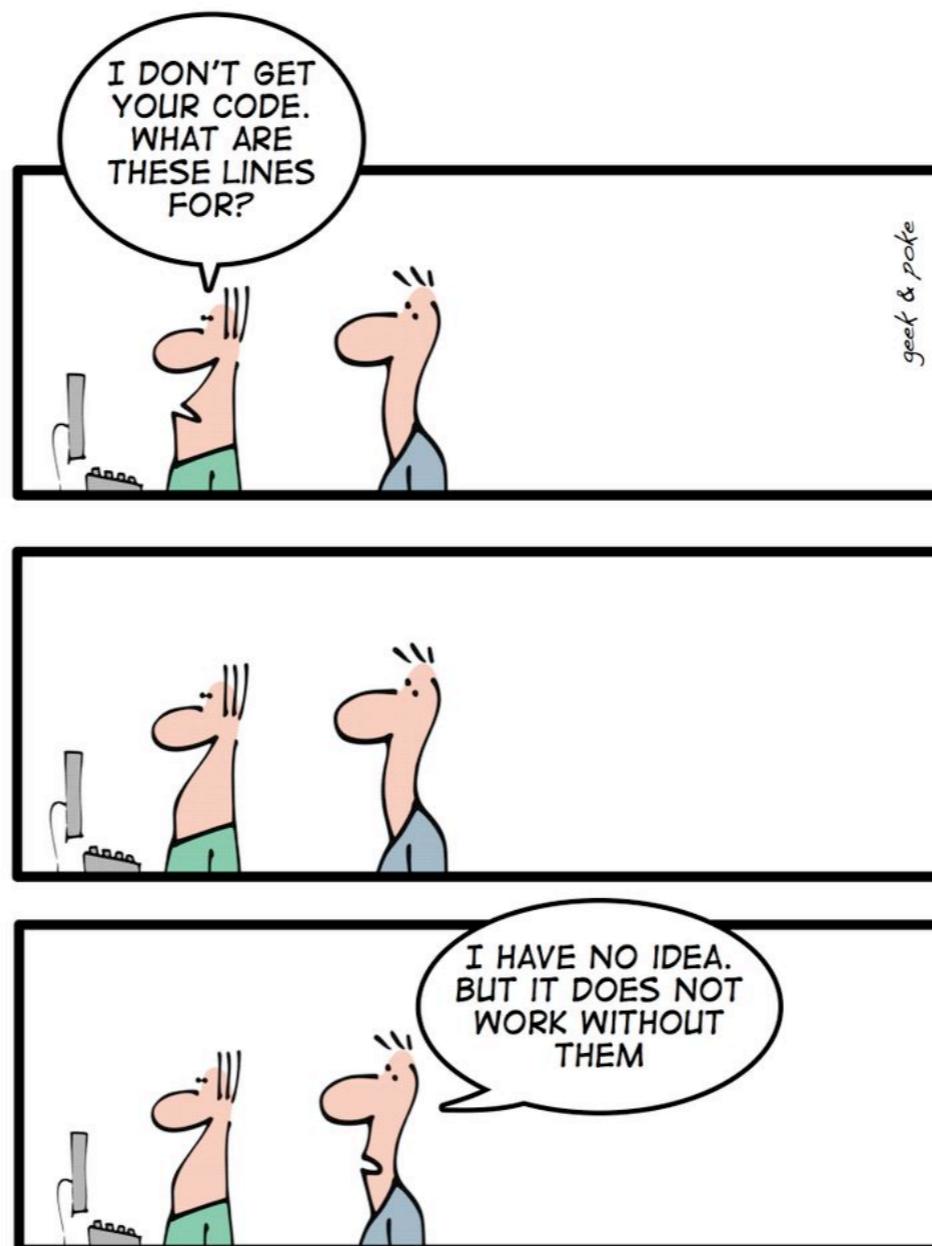
[Customize visualization \(NEW!\)](#)

Print output (drag lower right corner to resize)

```
multi-form
multi-dimensional
multi-lingual
multi-lateral
```

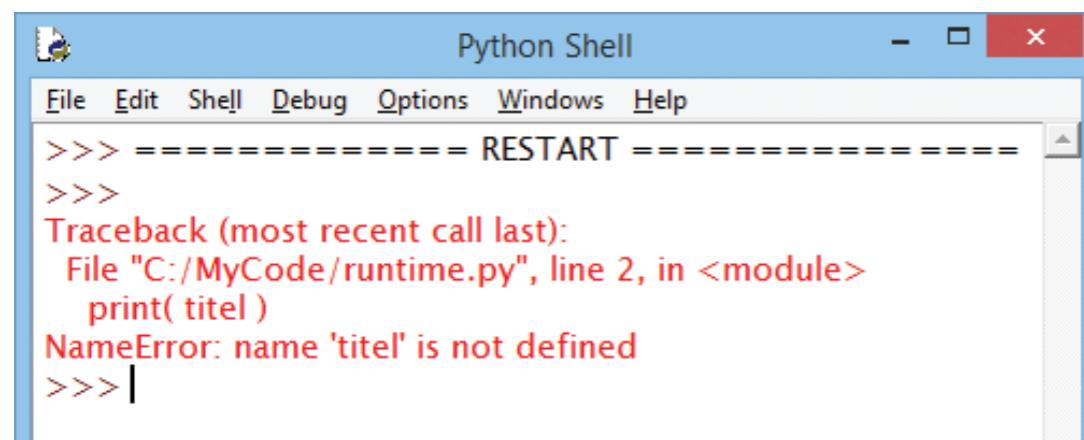


The struggles of syntax

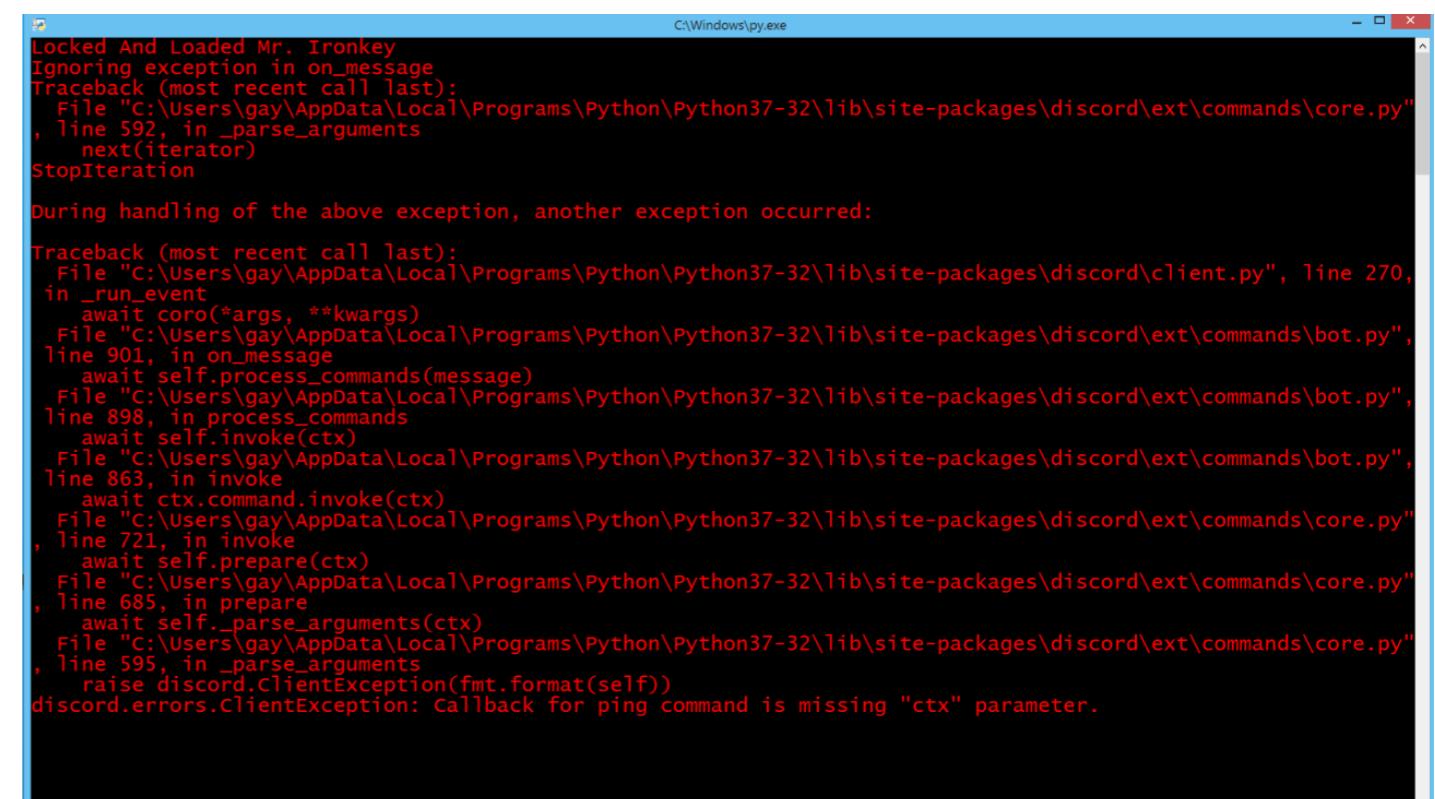


THE ART OF PROGRAMMING - PART 2: KISS

Errors



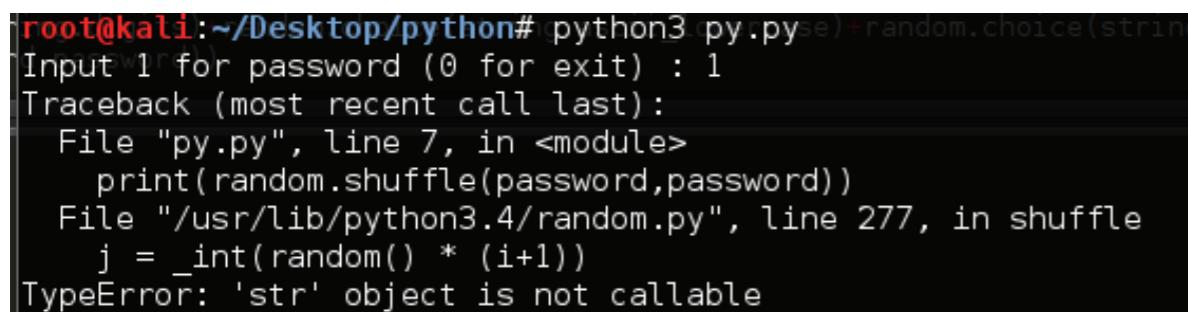
```
Python Shell
File Edit Shell Debug Options Windows Help
>>> ====== RESTART ======
>>>
Traceback (most recent call last):
  File "C:/MyCode/runtime.py", line 2, in <module>
    print( titel )
NameError: name 'titel' is not defined
>>> |
```



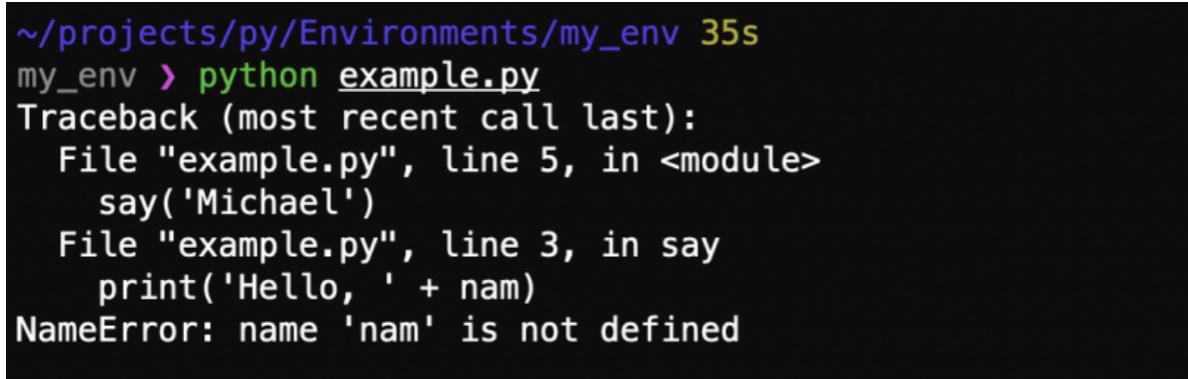
```
C:\Windows\py.exe
Locked And Loaded Mr. Ironkey
Ignoring exception in on_message
Traceback (most recent call last):
  File "C:\Users\gay\AppData\Local\Programs\Python\Python37-32\lib\site-packages\discord\ext\commands\core.py",
    line 592, in _parse_arguments
      next(iterator)
StopIteration

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Users\gay\AppData\Local\Programs\Python\Python37-32\lib\site-packages\discord\client.py", line 270,
    in _run_event
      await coro(*args, **kwargs)
  File "C:\Users\gay\AppData\Local\Programs\Python\Python37-32\lib\site-packages\discord\ext\commands\bot.py",
    line 901, in on_message
      await self._process_commands(message)
  File "C:\Users\gay\AppData\Local\Programs\Python\Python37-32\lib\site-packages\discord\ext\commands\bot.py",
    line 898, in process_commands
      await self.invoke(ctx)
  File "C:\Users\gay\AppData\Local\Programs\Python\Python37-32\lib\site-packages\discord\ext\commands\bot.py",
    line 863, in invoke
      await ctx.command.invoke(ctx)
  File "C:\Users\gay\AppData\Local\Programs\Python\Python37-32\lib\site-packages\discord\ext\commands\core.py",
    line 721, in invoke
      await self._prepare(ctx)
  File "C:\Users\gay\AppData\Local\Programs\Python\Python37-32\lib\site-packages\discord\ext\commands\core.py",
    line 685, in _prepare
      await self._parse_arguments(ctx)
  File "C:\Users\gay\AppData\Local\Programs\Python\Python37-32\lib\site-packages\discord\ext\commands\core.py",
    line 595, in _parse_arguments
      raise discord.ClientException(fmt.format(self))
discord.errors.ClientException: Callback for ping command is missing "ctx" parameter.
```



```
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
```

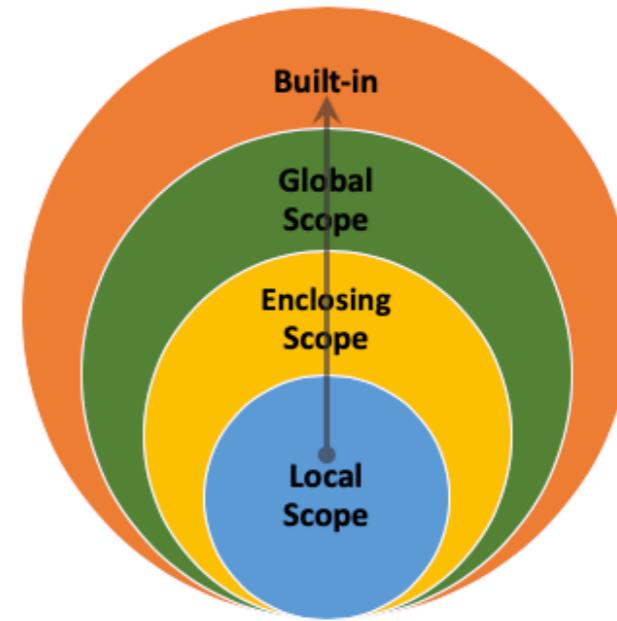


```
~/projects/py/Environments/my_env 35s
my_env > python example.py
Traceback (most recent call last):
  File "example.py", line 5, in <module>
    say('Michael')
  File "example.py", line 3, in say
    print('Hello, ' + nam)
NameError: name 'nam' is not defined
```

Debugging

- Find the line number where there's an error.
- Print everything above and below that line to scrutinize that line.
- SyntaxError
- IndexError, KeyError
- NameError
- TypeError
- IndentationError
- ModuleNotFoundError, ImportError

Scope - where can you access a variable?

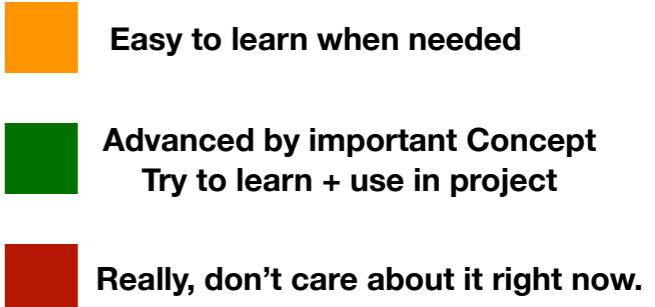


- Local: Made in function, available in function ONLY.
- Enclosing: Nested functions
- Global: `global x = 1`
- Built in: `sorted()`, `max()`, `float()`

```
# Global scope
x = 0

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

Concepts left out

- Custom Classes/Objects:
 - For ex: Employee object, inbuilt functions : get_salary(), get_address(), get_team_name(), get_role().
 - Inheritance : classes having access to parts of other classes
 - For ex: Accountant object inherits from Employee object
 - Lambda functions: fancy, shorthand python functions
 - JSON : format for reading/writing files used by people
 - RegEx : Ninja Level matching strings and parts of strings
 - Iterators : looping beyond while and for.
- 
- The legend consists of three colored squares with corresponding text:
Orange square: Easy to learn when needed
Green square: Advanced by important Concept
Try to learn + use in project
Red square: Really, don't care about it right now.

Learning to Learn - google and get going

- Using matplotlib (plt) to change titles etc in plots
- Using beatifulsoup for scraping the internet
- Some more today
- Lots more of it in your project

"Programming isn't about what you know; it's about what you can figure out." - Chris Pine

Case Study 1 - get a user input button with Python

Case Study 2 - Getting time in Python

Case Study 3 - Read and Display an Image

Case Study 4 - Pass argument to python script

Final Logistics

- Projects : One on one meetings + GitHub issues
- Final Exam
- A repository of computational tools you would love to have. We could continue to ideate on it, to learn to think computationally.

Putting it all together

You have all the pieces, let's see what you build with them :)



That's all Folks!

You have all the pieces, let's see what you build with them :)