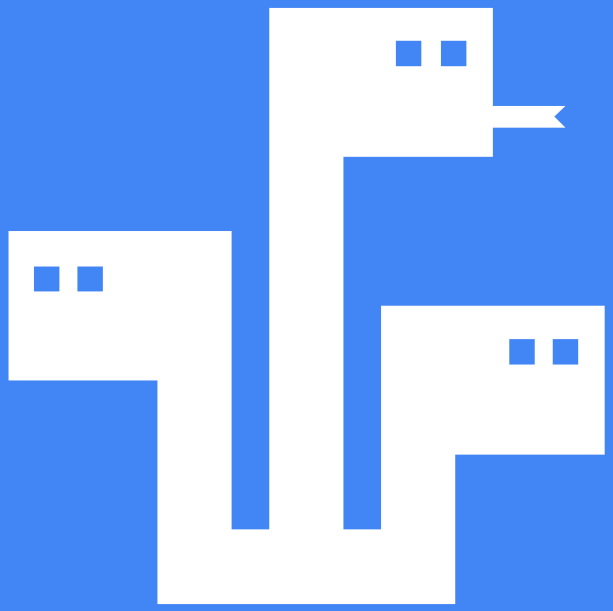
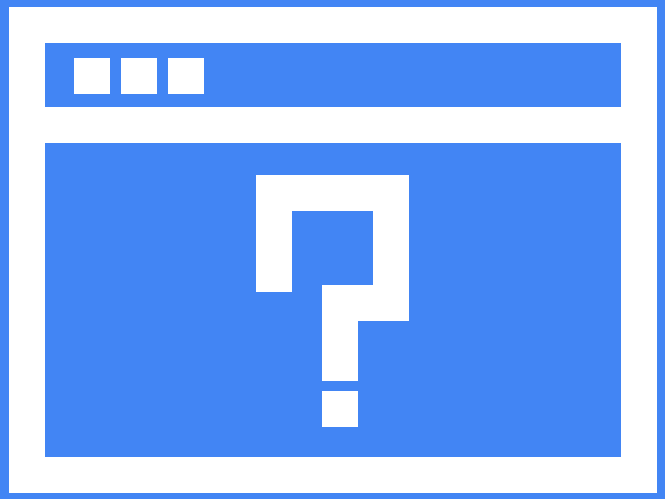
Basics

Python - Nick Reynolds
September 4, 2017



Who are you?



10 Weeks

- Development
- Concepts and Logic
- Real world applications

Schedule

Week 01 - Basics	
Week 02 - Conditionals and Functions	
Week 03 - Lists and Loops	Assignment 1 Released
Week 04 - Dictionaries and Revision	
Week 05 - Classes	Assignment 1 Due
Week 06 - File Input / Output	Assignment 2 Released
Week 07 - Testing and Regular Expressions	
Week 08 - Decorators and Design Patterns	
Week 09 - Revision	Assignment 2 Due
Week 10 - Exam	

What's a program?

- Input
- Output
- Math
- Conditions
- Repetition

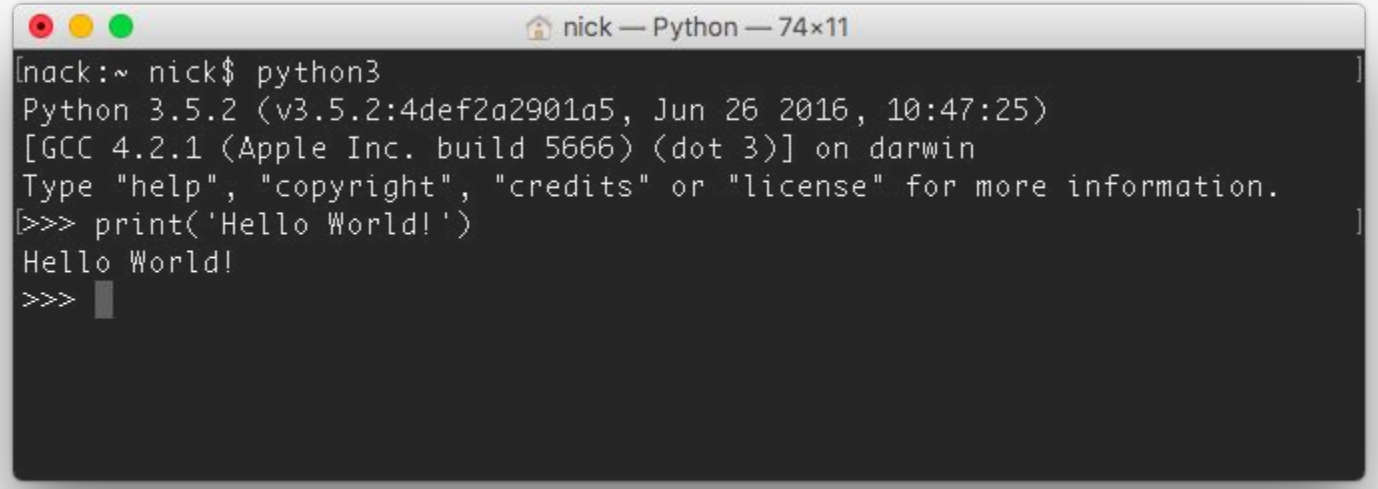
Python

- High level language
- General purpose language
- Low overhead

Dropbox

reddit

Accessing Python



Through the terminal or an Integrated Development Environment (IDE) like PyCharm

Variables

Variables & Types

```
>>> x = 5                    # number
>>> x
5
>>> x = 'Jack'               # string
>>> x
'Jack'
>>> x = [5, 'Jack']          # list
>>> x
[5, 'Jack']
```

Number/Decimal Type

- Numbers and maths can be written out as you would expect
- Typical expressions:
  - + Plus
  - - Minus
  - / Divide
  - * Multiply
- Also supports equality checks:
  - < Less than
  - <= Less than or equal to
  - > Greater than
  - >= Greater than or equal to
  - == Equals
- Equality checks return True or False

```
>>> 1 + 2
3
>>> 3 * 3
9
>>> 3 - 1.5
1.5
>>> x = 3
>>> y = 5
>>> x * y
15
>>> x < y
True
>>> x < 1
False
```

String Type

- Strings are a collection of characters, like words or sentences
- Declared with a double or single quote
- Multiline strings are done using three quotes
- Lots of built in string manipulators!

```
>>> a = "Cat"
>>> b = 'Dog'
>>> print(a + b)
CatDog
>>> c = '''Wow this is a really long
Multiple line'''
>>> a.upper()
CAT
>>> len(a)
3
>>> a.replace("at", "up")
Cup
>>> a.find("t")
2
```

Indexes Start at Zero

Word	P	y	t	h	o	n
Index	0	1	2	3	4	5

String Type Continued

- Strings are like a list of characters
- You can access specific elements with square brackets
  - .split(*delimiter*) can be used to split your string into parts
  - *delimiter*.join(*list*) puts it back together

```
>>> d = "Catdog"
>>> d[4]
'o'
>>> d.find('t')
2
>>> d[-1].upper()
'G'
>>> d[2:4]
'td'
>>> words = d.split('d')
["Cat", "og"]
>>> 'Fr'.join(words)
'CatFrog'
```

List Type

- Lists are elements in succession
  - append(*value*) - Add items to the list
  - del - deleting an item
  - remove(*value*) - remove an item by value
  - insert(*index, value*) - add a value at index
  - index(*value*) - to get the index of a value
- Useful applications
  - *value* in *list* returns True or False
  - *value* not in *list* returns True or False

```
>>> Shopping = ["Apple", "Banana",
"Orange"]
>>> Shopping[2]
"Orange"
>>> Shopping[3]
IndexError: list index out of range
>>> del Shopping[0]
["Banana", "Orange"]
>>> Shopping.append("Pear")
["Banana", "Orange", "Pear"]
>>> Shopping.remove("Orange")
["Banana", "Pear"]
>>> "Pear" in Shopping
True
```

Variables: Tuples

- Like lists but can't be changed after being declared
- Useful if we don't want our list to be changed
- They are also faster to access and smaller in memory!

```
>>> bob = (1,2,3)
>>> bob = ("Apple", "Orange")
>>> bob[1]
'Orange'
>>> bob.append("Pear")
AttributeError: 'tuple' object has no
attribute 'append'
```

Variables: Dictionaries

- Data in lists is stored according to a number of key -> pairs
- Great for storing labeled information about things
- Useful methods:
  - .keys() - Lists out the keys of a dictionary
  - .clear()

```
>>> person = {
...     'name': 'Winston',
...     'age': 29
... }
>>> person['name']
'Winston'
>>> person['job'] = 'Scientist'
>>> person
{'age': 29, 'name': 'Winston',
'job': 'Scientist'}
>>> person.keys()
dict_keys(['age', 'name', 'job'])
>>> 'age' in person
True
```

Pen and Paper

Checkpoint

Control Flow

Conditionals

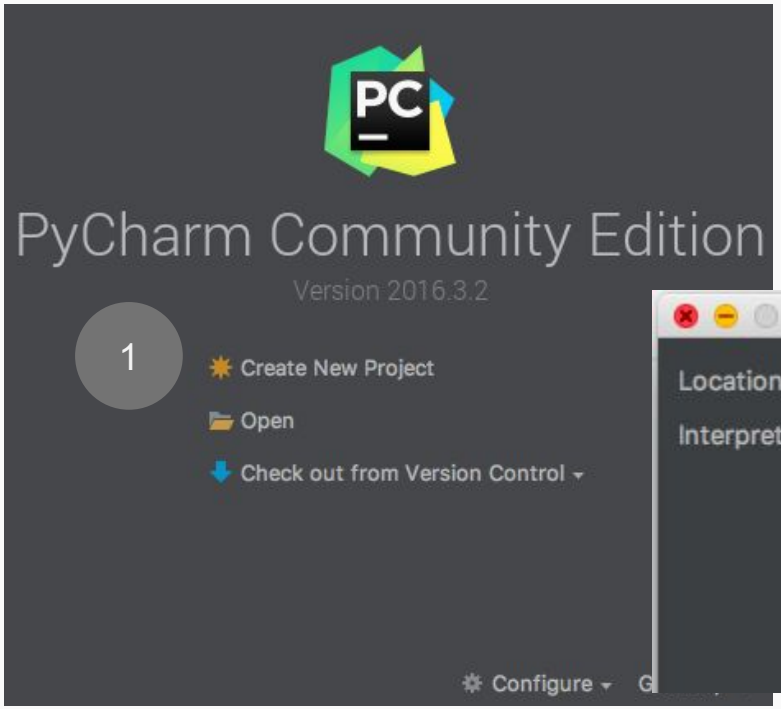
- Conditionals or IF statements control the flow of your application
- If conditions are satisfied then code is executed, else it is not
- Keywords:
  - and
  - or
  - not

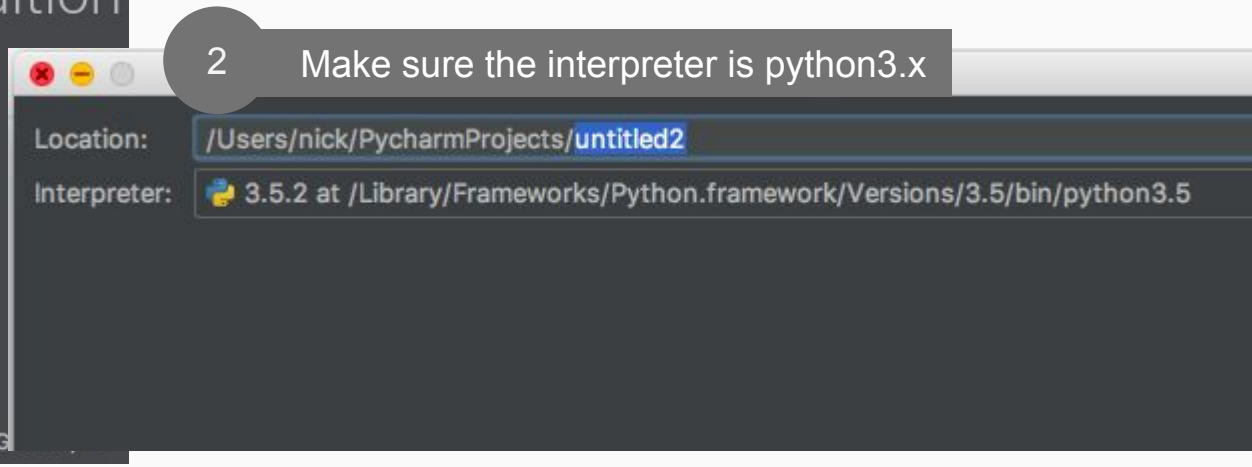
```
>>> a = 5
>>> if a < 10:
...     print('a is small!')
... else:
...     print('a is big!')
...
a is small!
>>> if a < 10 and a == 3:
...     print('a is 3!')
... else:
...     print('no luck!')
...
no luck!
```

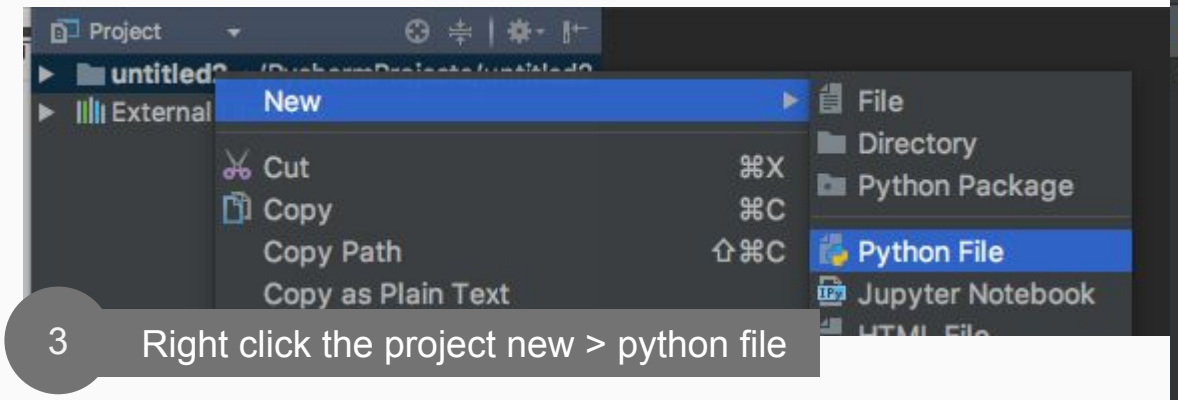
Conditionals Continued

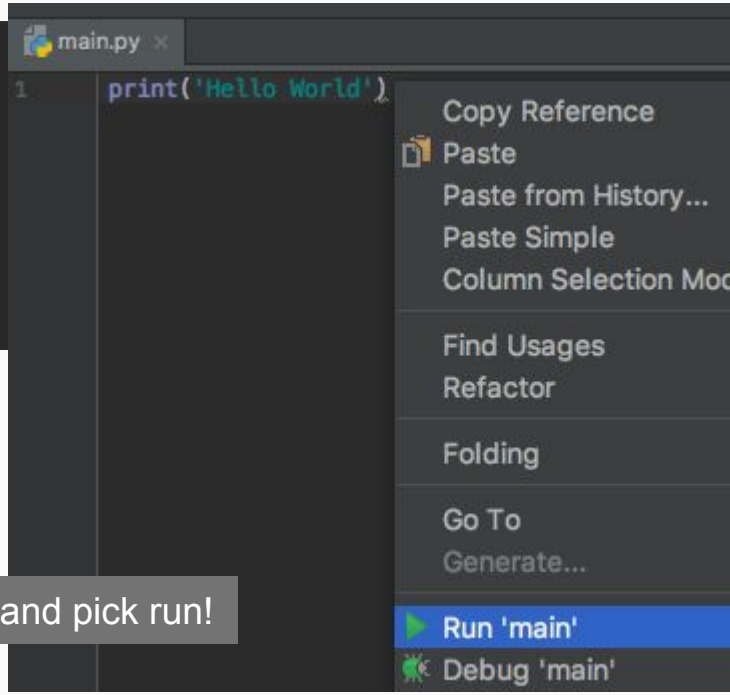
```
>>> a = 5
>>> b = 10
>>> if a < 10 and b > a:
...     print('fizz')
... elif a == 3 or b <= 3:
...     print('buzz')
... else:
...     print('pop')
...
fizz
```

```
>>> shop = ["apple", "banana"]
>>> b = 10
>>> if "apple" in shop and b != 10:
...     print('fizz')
... elif "pear" not in shop:
...     print('buzz')
... else:
...     print('pop')
...
buzz
```


PyCharm Community Edition
Version 2016.3.2
1
Create New Project
Open
Check out from Version Control
Configure


2
Make sure the interpreter is python3.x
Location:
/Users/nick/PycharmProjects/untitled2
Interpreter:
3.5.2 at /Library/Frameworks/Python.framework/Versions/3.5/bin/python3.5


Project
untitled2
External
New
Cut ⌘X
Copy ⌘C
Copy Path ⇧⌘C
Copy as Plain Text
File
Directory
Python Package
Python File
Jupyter Notebook
3
Right click the project new > python file


main.py
print('Hello World')
Copy Reference
Paste
Paste from History...
Paste Simple
Column Selection Mod
Find Usages
Refactor
Folding
Go To
Generate...
Run 'main'
Debug 'main'
4
Right click the file and pick run!

References

- http://pwp.stevecassidy.net/python/basic-python.html
- https://thenounproject.com/