Welcome to Introduction to Python – Level 1

DEVOXX™ 4KIDS BAY AREA

What do you need to get started?

1. Python (3.3.5): https://www.python.org/downloads/
2. PyCharm: http://www.jetbrains.com/pycharm/download/

(Make sure you download this before the class and you have right version for your platform, Windows or Mac)

Class time: 2 hrs. and 30 minutes (with two 15 minutes breaks!)

Wi-Fi: Free-WiFi

Who is helping you today?

Arun Gupta : Instructor

Anoop Trivedi: Instructor

Tori: Logistics

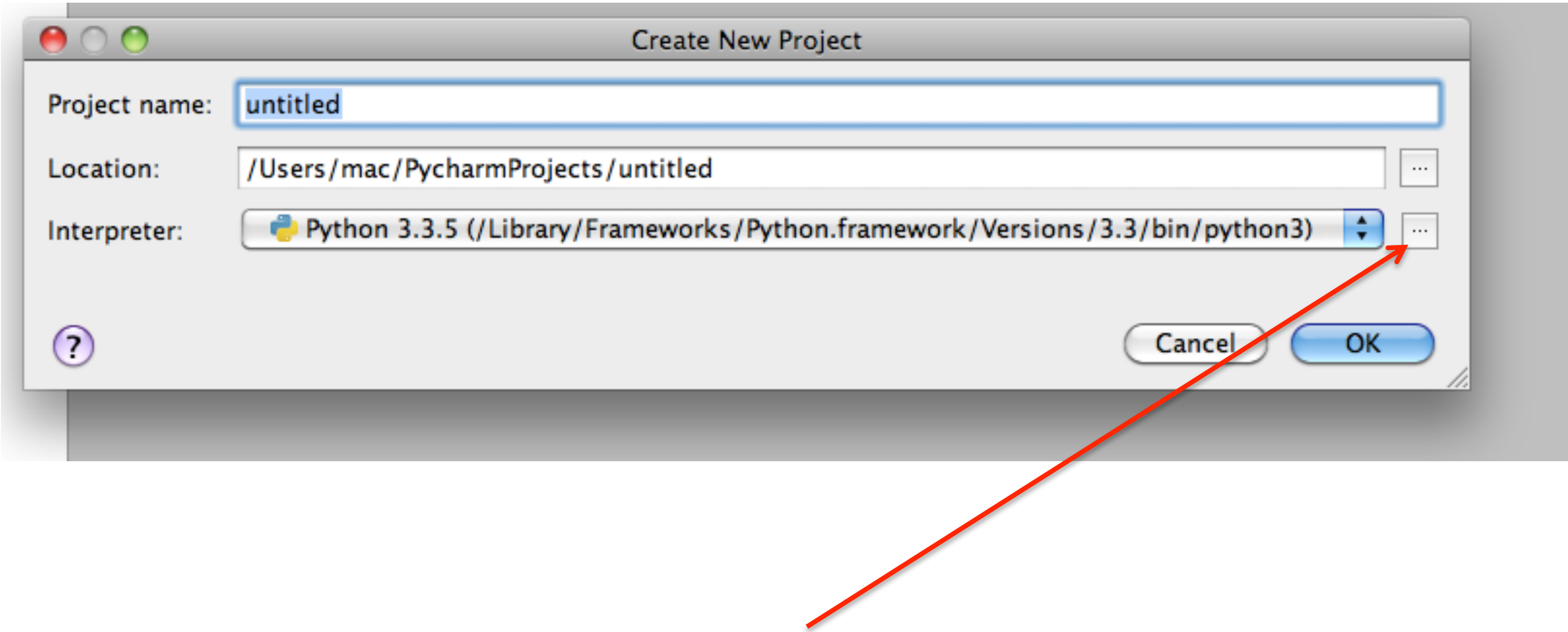
Vivian: Facilities

What are we going to do today ?

1. *Installation and Howto Run Python*
2. *Basics of Python Programming Lang.*
3. *Input and output statements*
4. *All About data 'types'*
5. *Conditional Execution*
6. *Practice Examples!*
7. *Q&A / Showcase*

Python Setup:

Create New Project in PyCharm



Create New Project

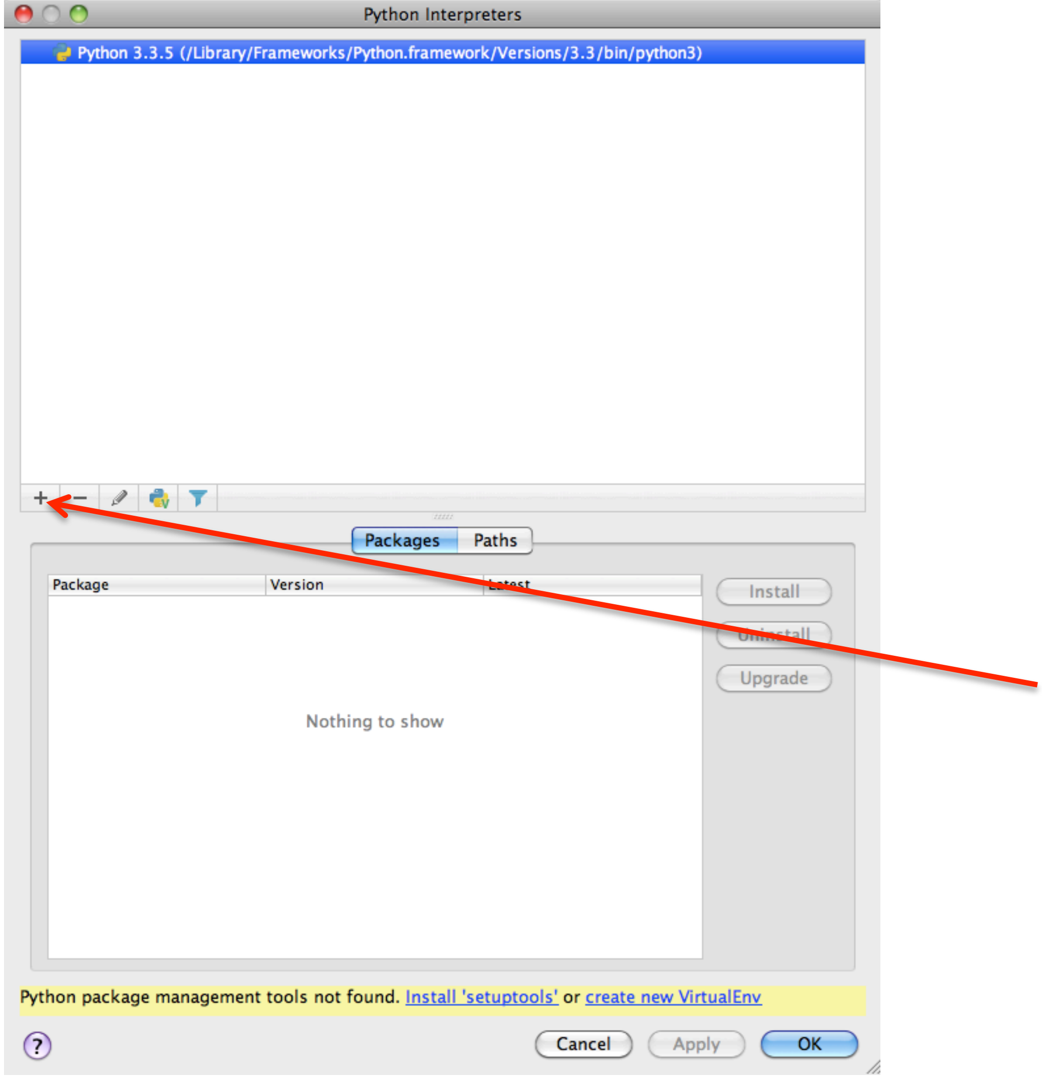
Project name: untitled

Location: /Users/mac/PycharmProjects/untitled

Interpreter: Python 3.3.5 (/Library/Frameworks/Python.framework/Versions/3.3/bin/python3)

Click here to choose your interpreter

Python Setup:



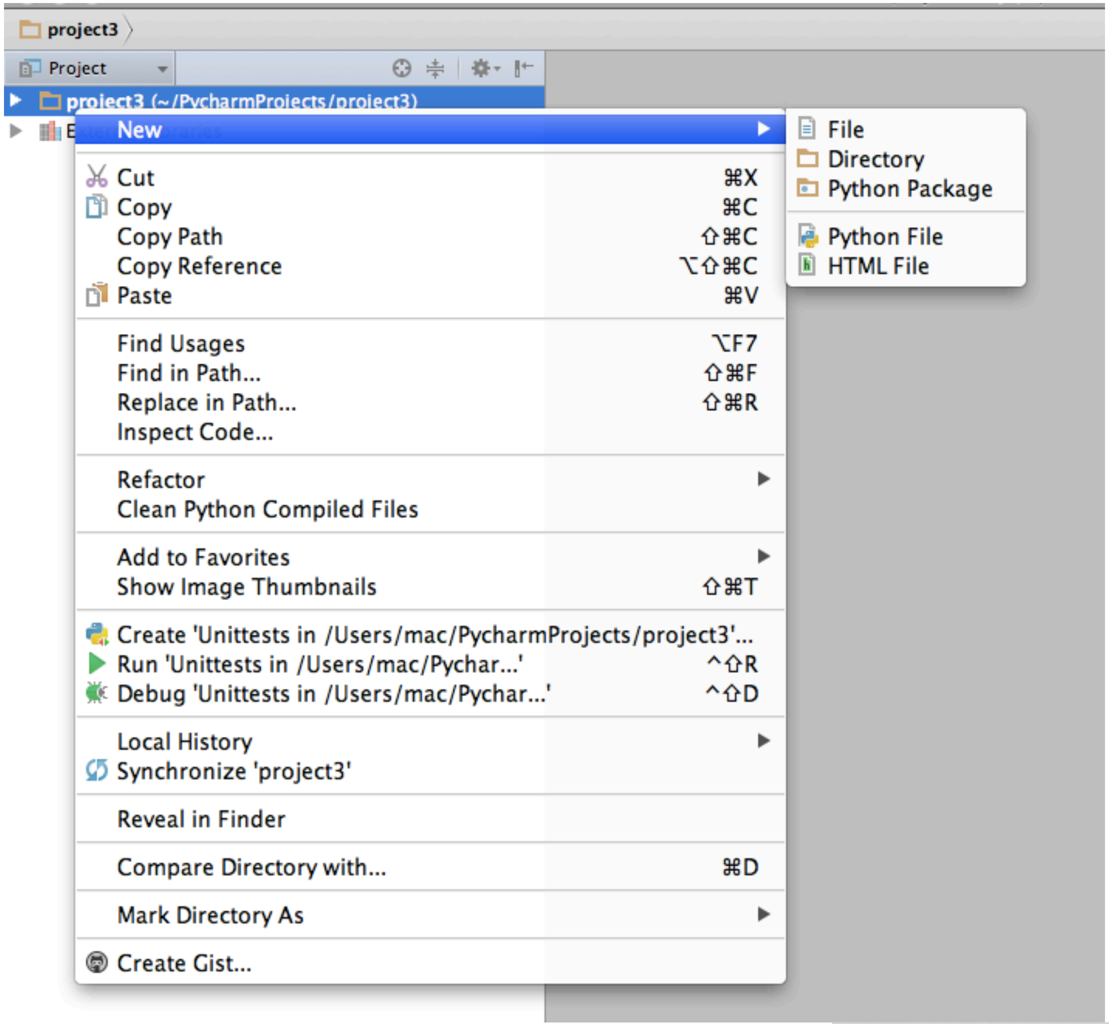
Python Interpreters

Python 3.3.5 (/Library/Frameworks/Python.framework/Versions/3.3/bin/python3)

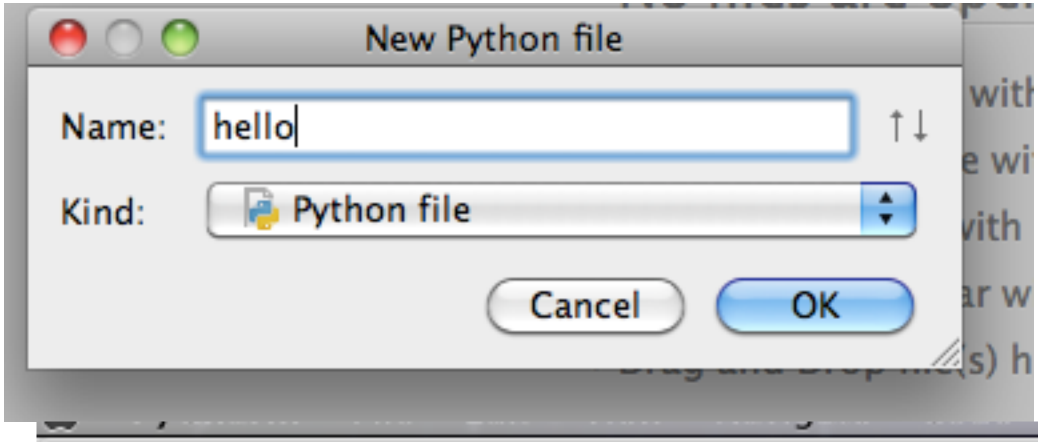
Python package management tools not found. Install 'setuptools' or create new VirtualEnv

Click on + sign to choose interpreter

Hello World!


project3
Project
project3 (~/PycharmProjects/project3)
External Libraries
New
Cut ⌘X
Copy ⌘C
Copy Path ⇧⌘C
Copy Reference ⌥⇧⌘C
Paste ⌘V
Find Usages ⌥F7
Find in Path... ⇧⌘F
Replace in Path... ⇧⌘R
Inspect Code...
Refactor
Clean Python Compiled Files
Add to Favorites
Show Image Thumbnails ⇧⌘T
Create 'Unittests in /Users/mac/PycharmProjects/project3'...
Run 'Unittests in /Users/mac/Pychar...' ^⇧R
Debug 'Unittests in /Users/mac/Pychar...' ^⇧D
Local History
Synchronize 'project3'
Reveal in Finder
Compare Directory with... ⌘D
Mark Directory As
Create Gist...
File
Directory
Python Package
Python File
HTML File

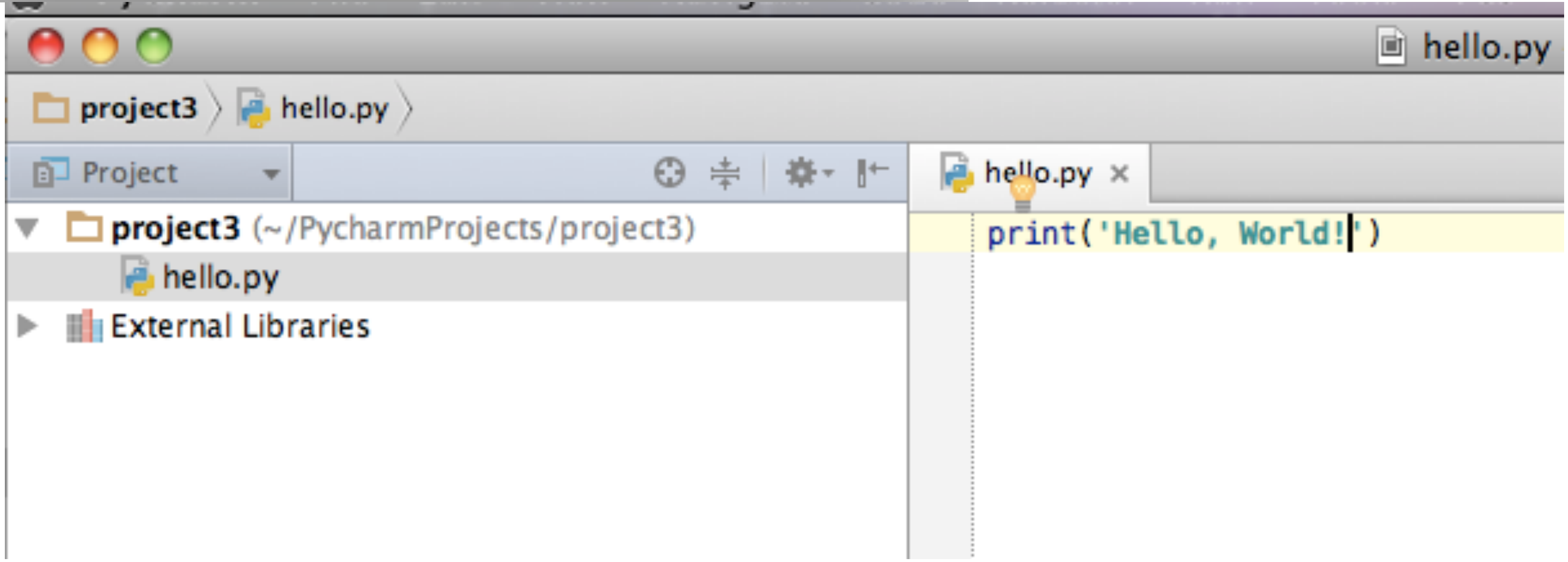
Hello World!



New Python file

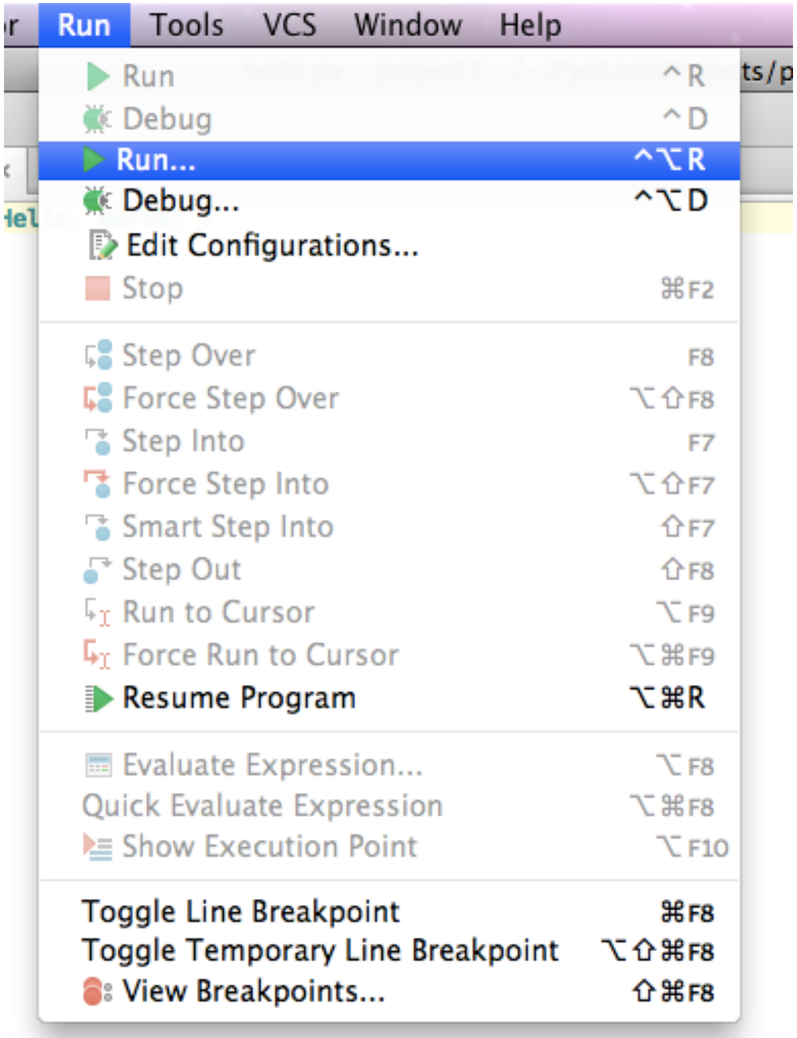
Name: hello

Kind: Python file

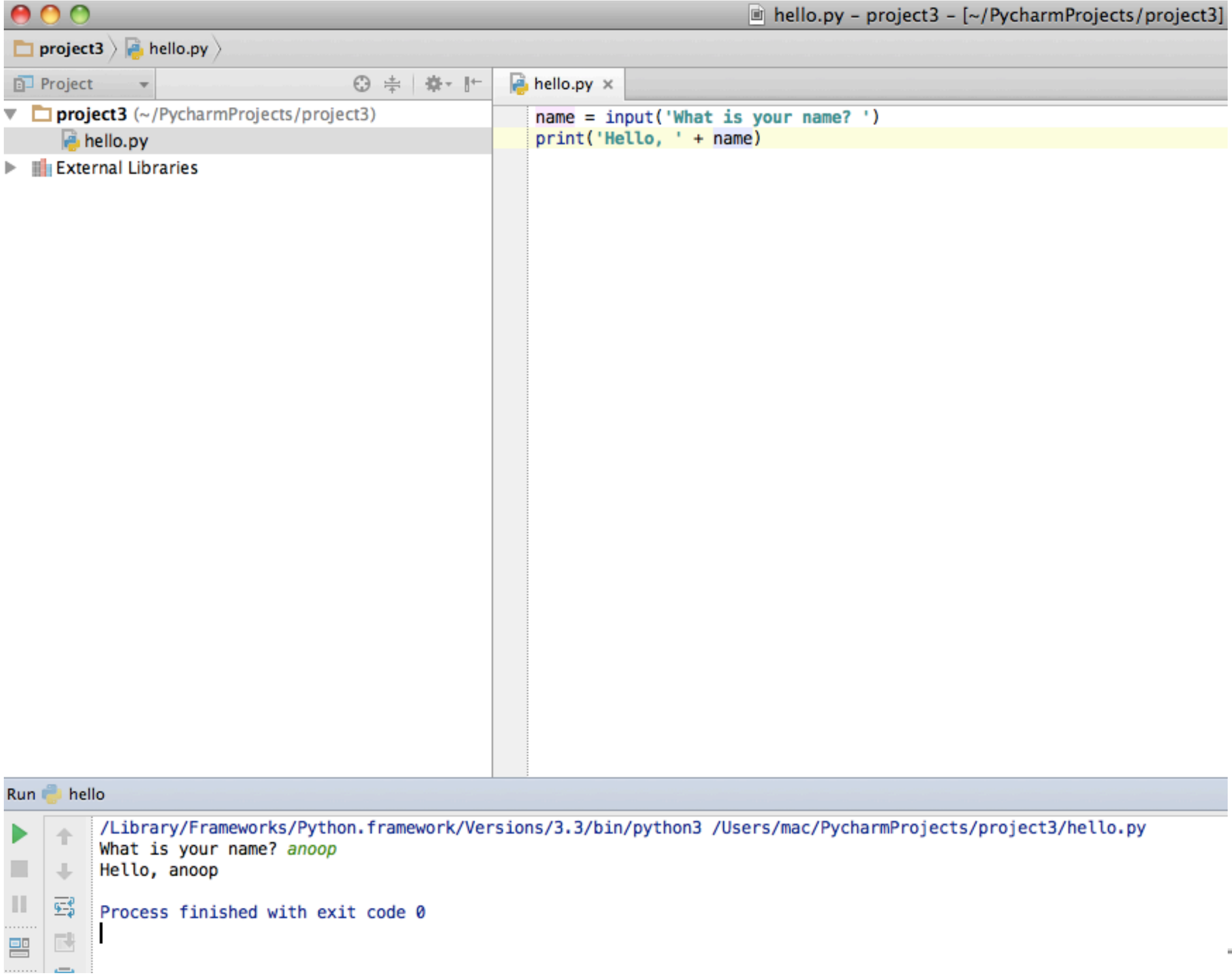


```
print('Hello, World!')
```

Hello World!


r Run Tools VCS Window Help
Run ^R
Debug ^D
Run... ^⌥R
Debug... ^⌥D
Edit Configurations...
Stop ⌘F2
Step Over F8
Force Step Over ⌥⇧F8
Step Into F7
Force Step Into ⌥⇧F7
Smart Step Into ⇧F7
Step Out ⇧F8
Run to Cursor ⌥F9
Force Run to Cursor ⌥⌘F9
Resume Program ⌥⌘R
Evaluate Expression... ⌥F8
Quick Evaluate Expression ⌥⌘F8
Show Execution Point ⌥F10
Toggle Line Breakpoint ⌘F8
Toggle Temporary Line Breakpoint ⌥⇧⌘F8
View Breakpoints... ⇧⌘F8
ts/p
Hel

Viola: Hello World!

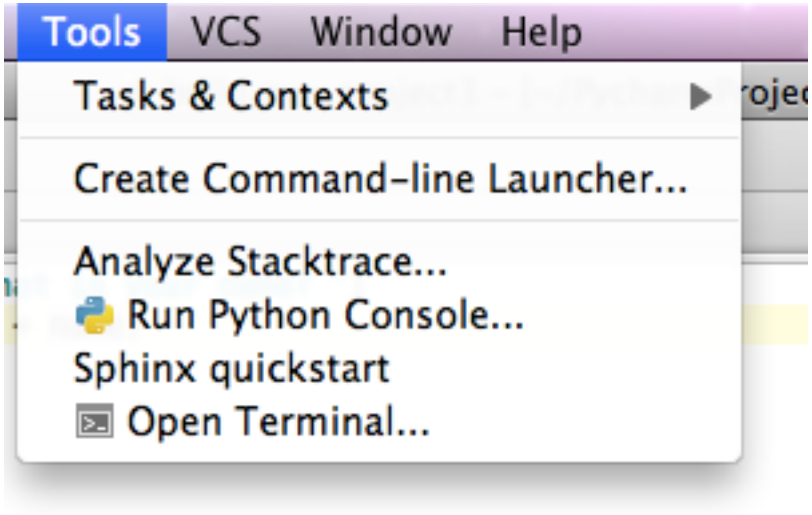

hello.py – project3 – [~/PycharmProjects/project3]
project3
hello.py
Project
project3 (~/PycharmProjects/project3)
hello.py
External Libraries
hello.py
Run
hello

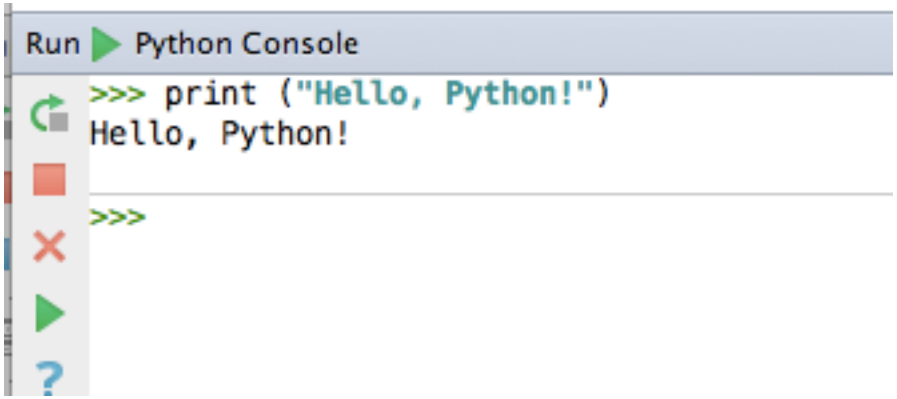
```
name = input('What is your name? ')
print('Hello, ' + name)
```

```
/Library/Frameworks/Python.framework/Versions/3.3/bin/python3 /Users/mac/PycharmProjects/project3/hello.py
What is your name? anoop
Hello, anoop

Process finished with exit code 0
```

name is a variable

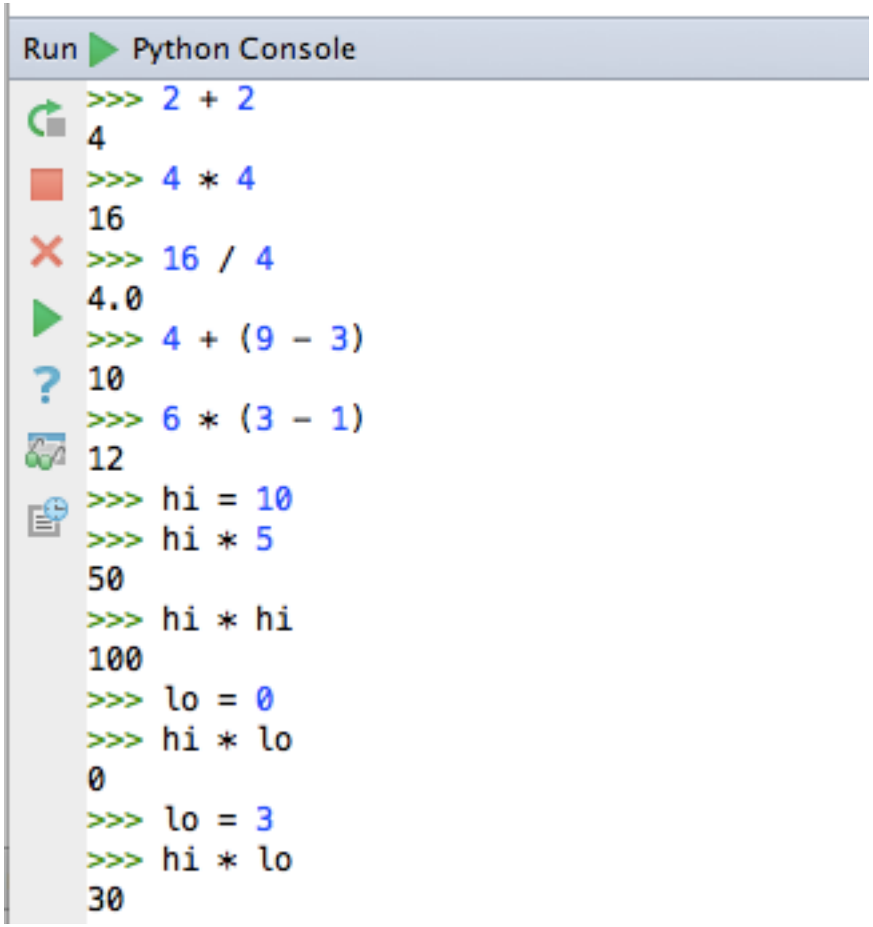
Python Console:





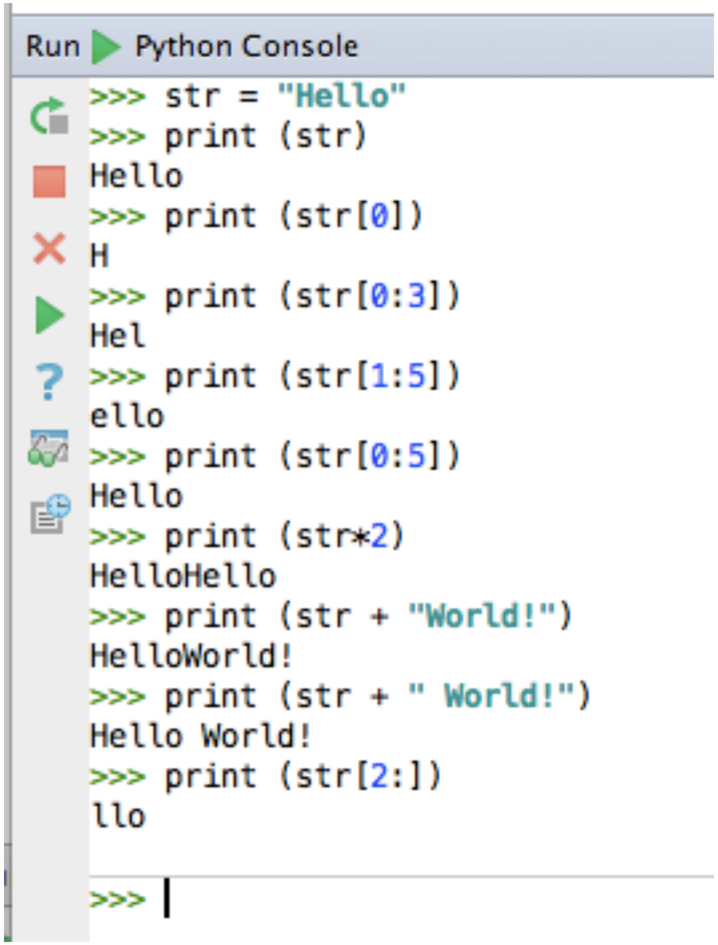
```
>>> print ("Hello, Python!")
Hello, Python!

>>>
```

More Python Console Examples:



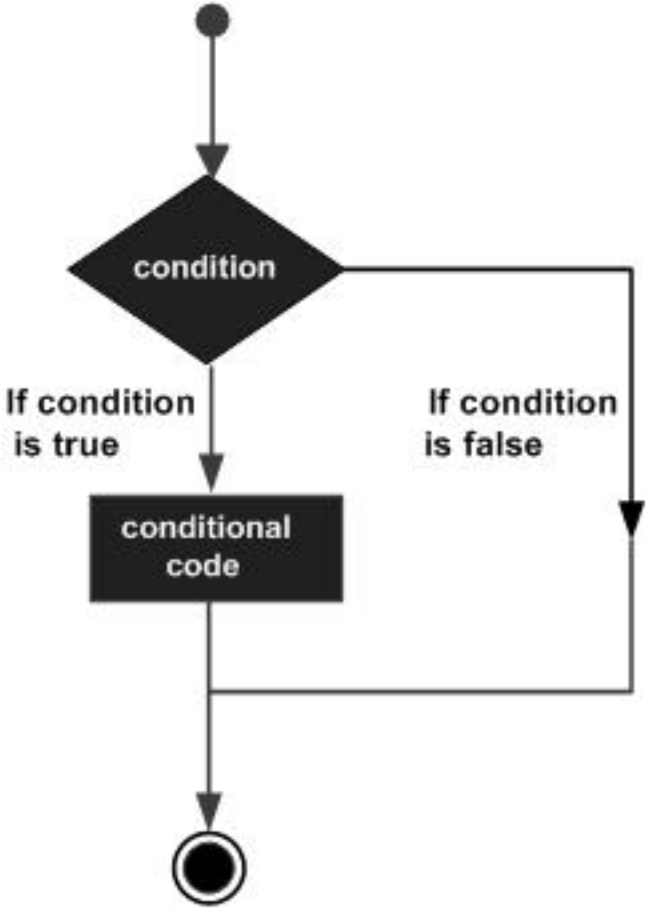
```
>>> 2 + 2
4
>>> 4 * 4
16
>>> 16 / 4
4.0
>>> 4 + (9 - 3)
10
>>> 6 * (3 - 1)
12
>>> hi = 10
>>> hi * 5
50
>>> hi * hi
100
>>> lo = 0
>>> hi * lo
0
>>> lo = 3
>>> hi * lo
30
```

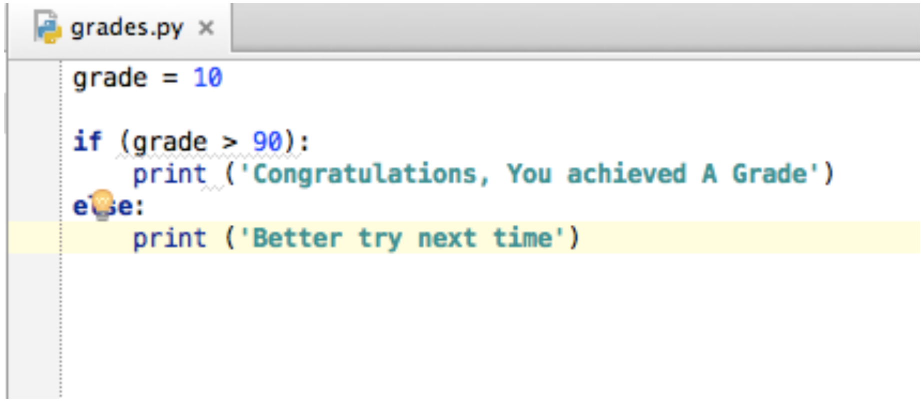
More Python Console Examples:



```
>>> str = "Hello"
>>> print (str)
Hello
>>> print (str[0])
H
>>> print (str[0:3])
Hel
>>> print (str[1:5])
ello
>>> print (str[0:5])
Hello
>>> print (str*2)
HelloHello
>>> print (str + "World!")
HelloWorld!
>>> print (str + " World!")
Hello World!
>>> print (str[2:])
llo

>>> 
```

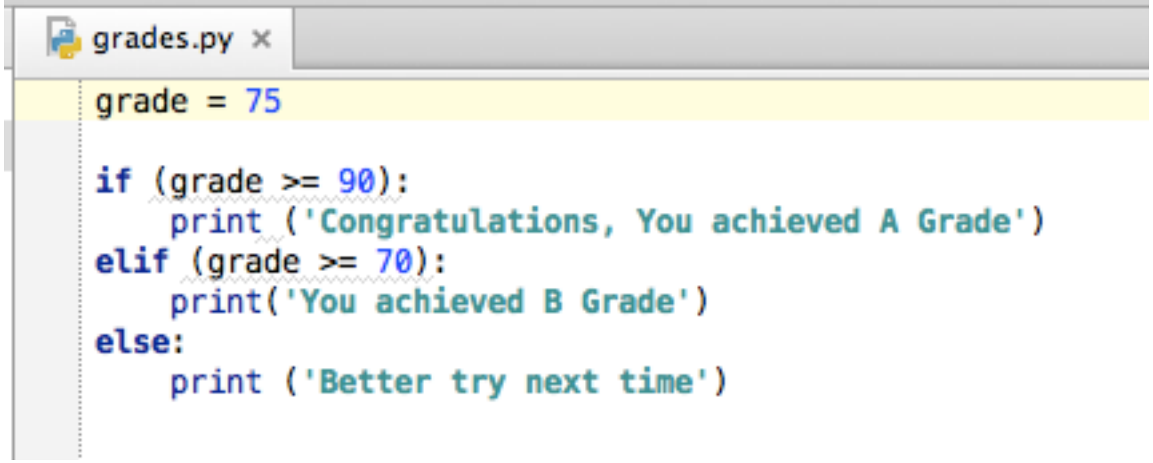
Conditional Statements





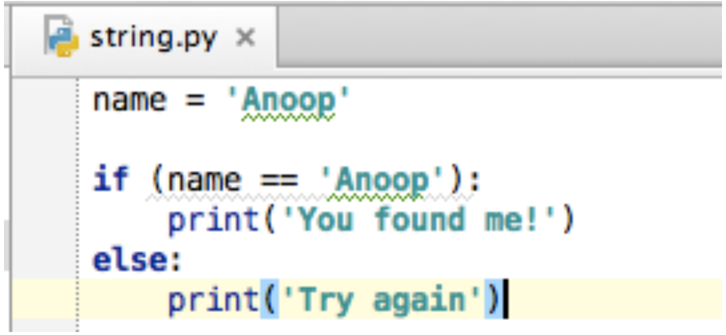
```
grade = 10

if (grade > 90):
    print ('Congratulations, You achieved A Grade')
else:
    print ('Better try next time')
```

Conditional Statements

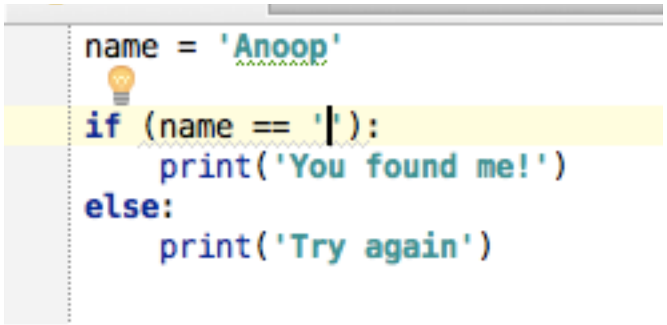


```
grade = 75

if (grade >= 90):
    print ('Congratulations, You achieved A Grade')
elif (grade >= 70):
    print('You achieved B Grade')
else:
    print ('Better try next time')
```

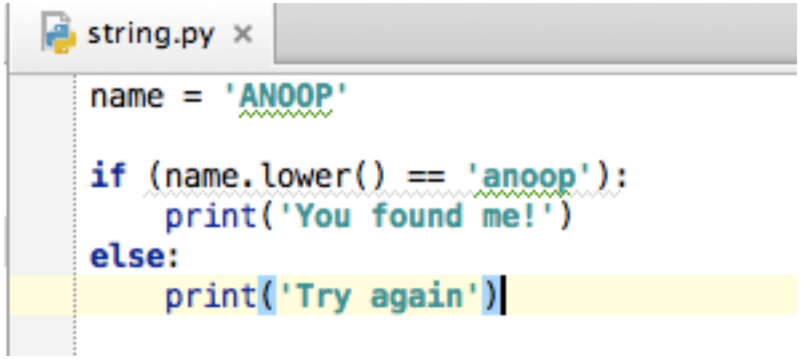
Conditional Statements



```python
name = 'Anoop'

if (name == 'Anoop'):
    print('You found me!')
else:
    print('Try again')
```

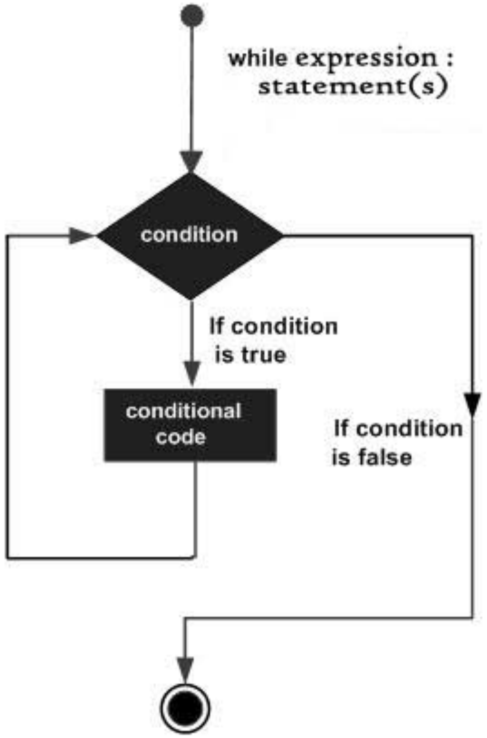


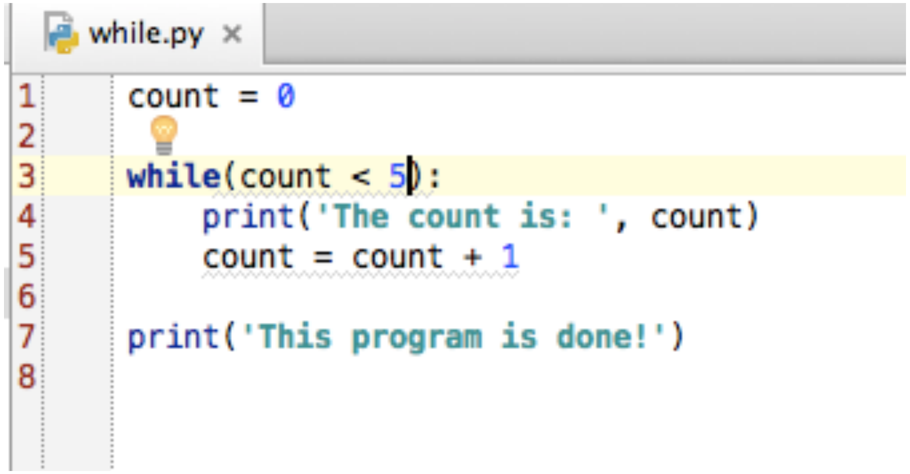
```python
name = 'Anoop'

if (name == ''):
    print('You found me!')
else:
    print('Try again')
```



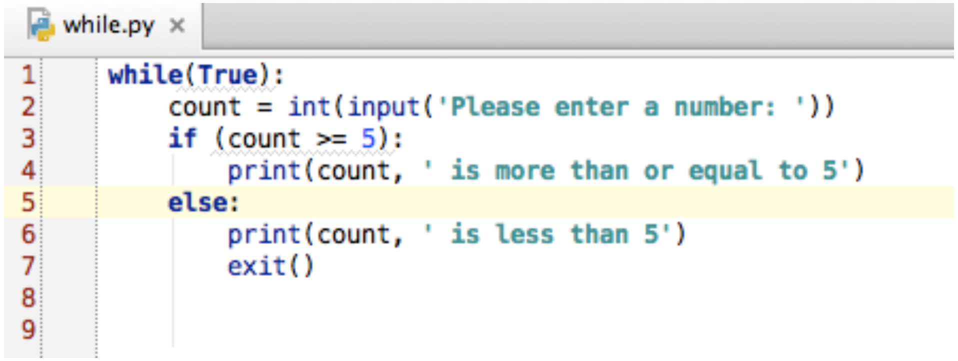
```python
name = 'ANOOP'

if (name.lower() == 'anoop'):
    print('You found me!')
else:
    print('Try again')
```

Conditional Statements





```
count = 0

while(count < 5):
    print('The count is: ', count)
    count = count + 1

print('This program is done!')
```



```
while(True):
    count = int(input('Please enter a number: '))
    if (count >= 5):
        print(count, ' is more than or equal to 5')
    else:
        print(count, ' is less than 5')
        exit()
```

Python Types

1. Numbers : (int, long, float, complex)

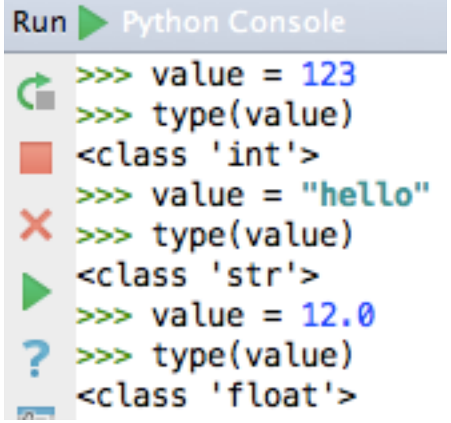
2. Strings

3. Lists

4. Tuple

5. Dictionary

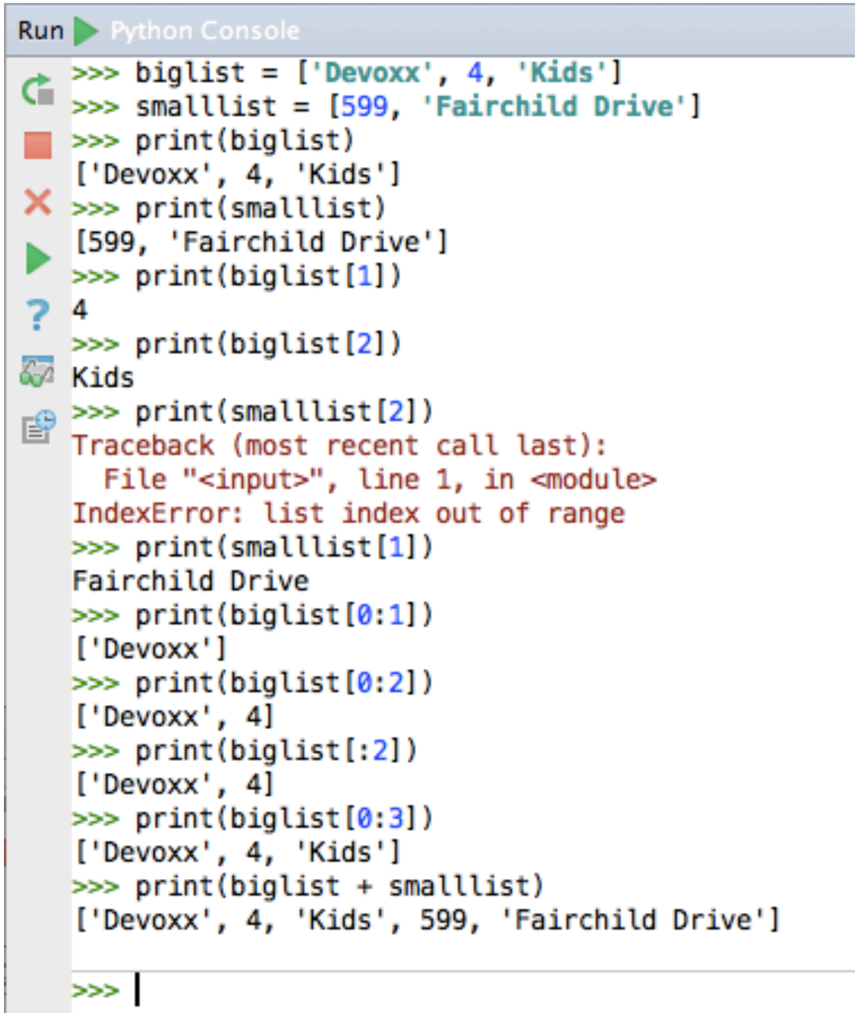
int, float, and string



```
>>> value = 123
>>> type(value)
<class 'int'>
>>> value = "hello"
>>> type(value)
<class 'str'>
>>> value = 12.0
>>> type(value)
<class 'float'>
```

List

"A list contains items separated by commas and enclosed within square brackets ([])"

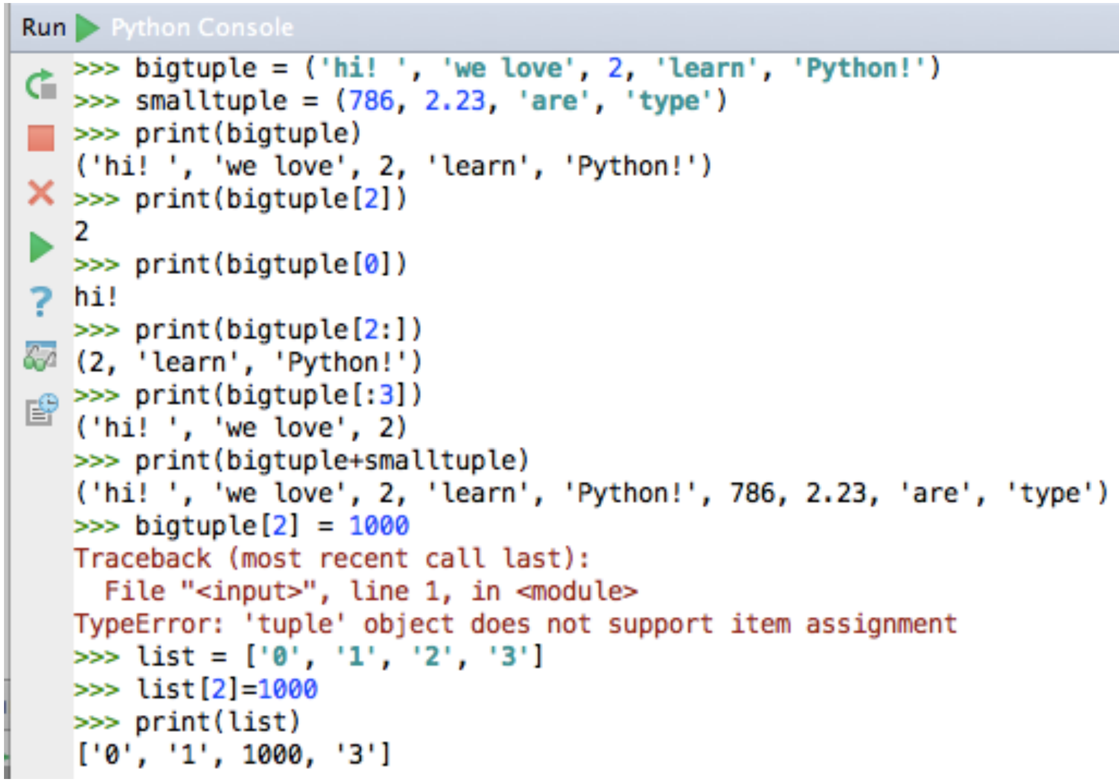


```
>>> biglist = ['Devoxx', 4, 'Kids']
>>> smalllist = [599, 'Fairchild Drive']
>>> print(biglist)
['Devoxx', 4, 'Kids']
>>> print(smalllist)
[599, 'Fairchild Drive']
>>> print(biglist[1])
4
>>> print(biglist[2])
Kids
>>> print(smalllist[2])
Traceback (most recent call last):
  File "<input>", line 1, in <module>
IndexError: list index out of range
>>> print(smalllist[1])
Fairchild Drive
>>> print(biglist[0:1])
['Devoxx']
>>> print(biglist[0:2])
['Devoxx', 4]
>>> print(biglist[:2])
['Devoxx', 4]
>>> print(biglist[0:3])
['Devoxx', 4, 'Kids']
>>> print(biglist + smalllist)
['Devoxx', 4, 'Kids', 599, 'Fairchild Drive']

>>> 
```

Tuples

“A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as read-only lists”

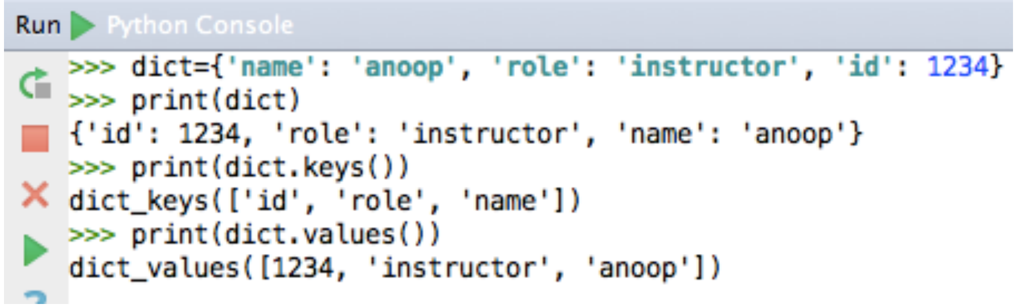


```
>>> bigtuple = ('hi! ', 'we love', 2, 'learn', 'Python!')
>>> smalltuple = (786, 2.23, 'are', 'type')
>>> print(bigtuple)
('hi! ', 'we love', 2, 'learn', 'Python!')
>>> print(bigtuple[2])
2
>>> print(bigtuple[0])
hi!
>>> print(bigtuple[2:])
(2, 'learn', 'Python!')
>>> print(bigtuple[:3])
('hi! ', 'we love', 2)
>>> print(bigtuple+smalltuple)
('hi! ', 'we love', 2, 'learn', 'Python!', 786, 2.23, 'are', 'type')
>>> bigtuple[2] = 1000
Traceback (most recent call last):
  File "<input>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>> list = ['0', '1', '2', '3']
>>> list[2]=1000
>>> print(list)
['0', '1', 1000, '3']
```

Dictionary

“Python's dictionaries are kind of hash table type. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries have no concept of order among elements. It is incorrect to say that the elements are "out of order"; they are simply unordered.”



```
>>> dict={'name': 'anoop', 'role': 'instructor', 'id': 1234}
>>> print(dict)
{'id': 1234, 'role': 'instructor', 'name': 'anoop'}
>>> print(dict.keys())
dict_keys(['id', 'role', 'name'])
>>> print(dict.values())
dict_values([1234, 'instructor', 'anoop'])
```

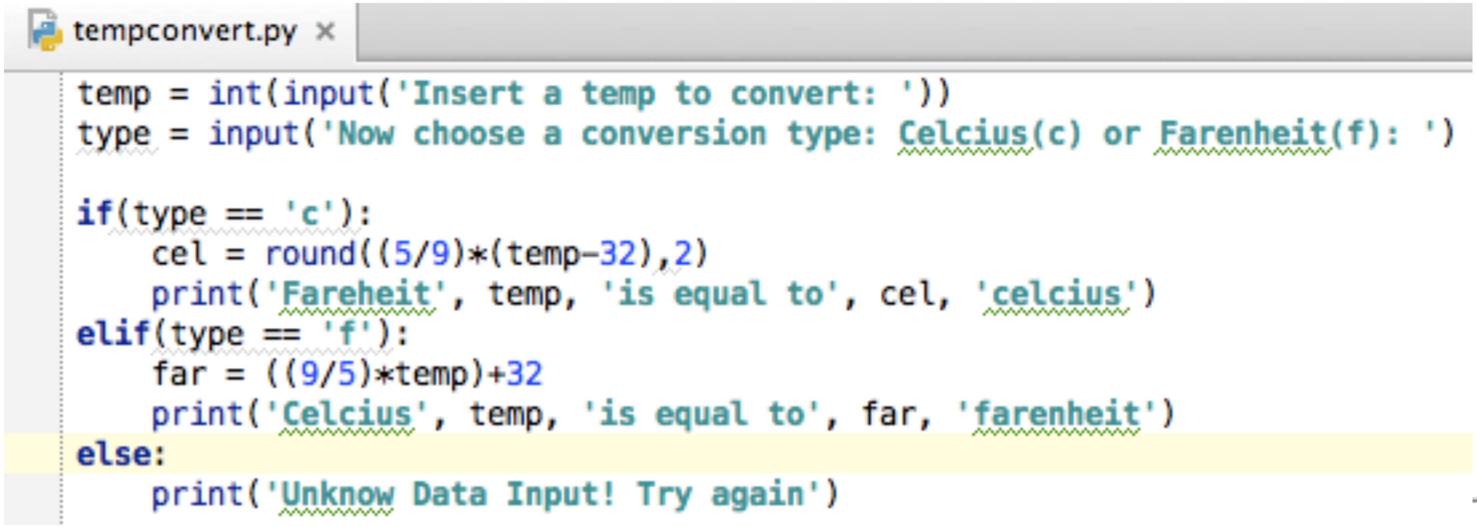
Temp Converter

Fahrenheit to Celsius:

(°F - 32) x 5/9 = °C or in plain English, First subtract 32, then multiply by 5, then divide by 9.

Celsius to Fahrenheit:

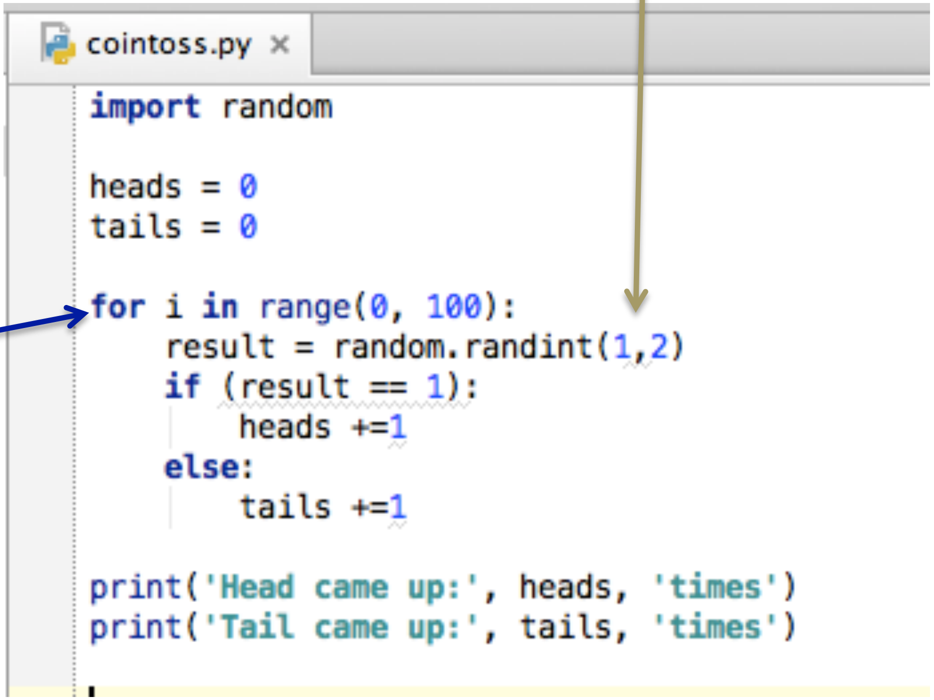
(°C × 9/5) + 32 = °F or in plain English, Multiple by 9, then divide by 5, then add 32.



```
temp = int(input('Insert a temp to convert: '))
type = input('Now choose a conversion type: Celcius(c) or Farenheit(f): ')

if(type == 'c'):
    cel = round((5/9)*(temp-32),2)
    print('Fareheit', temp, 'is equal to', cel, 'celcius')
elif(type == 'f'):
    far = ((9/5)*temp)+32
    print('Celcius', temp, 'is equal to', far, 'farenheit')
else:
    print('Unknow Data Input! Try again')
```

Coin Toss Game:

What Is the probability of heads or tails, if you flip a coin 100 times?

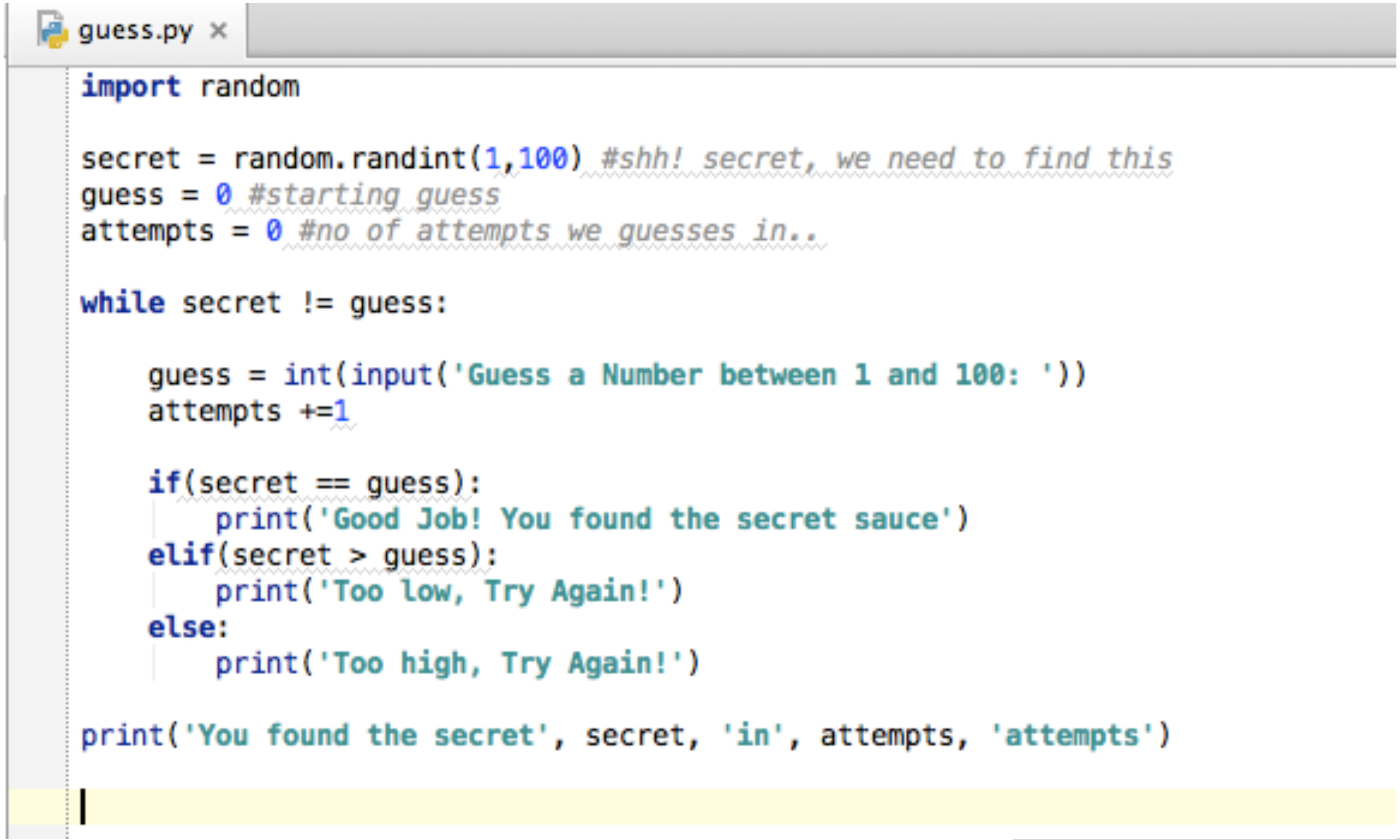


```python
import random

heads = 0
tails = 0

for i in range(0, 100):
    result = random.randint(1,2)
    if (result == 1):
        heads +=1
    else:
        tails +=1

print('Head came up:', heads, 'times')
print('Tail came up:', tails, 'times')
```

"For loop, another way of iterating, if you know size of set"

Mission: Find my secret number?

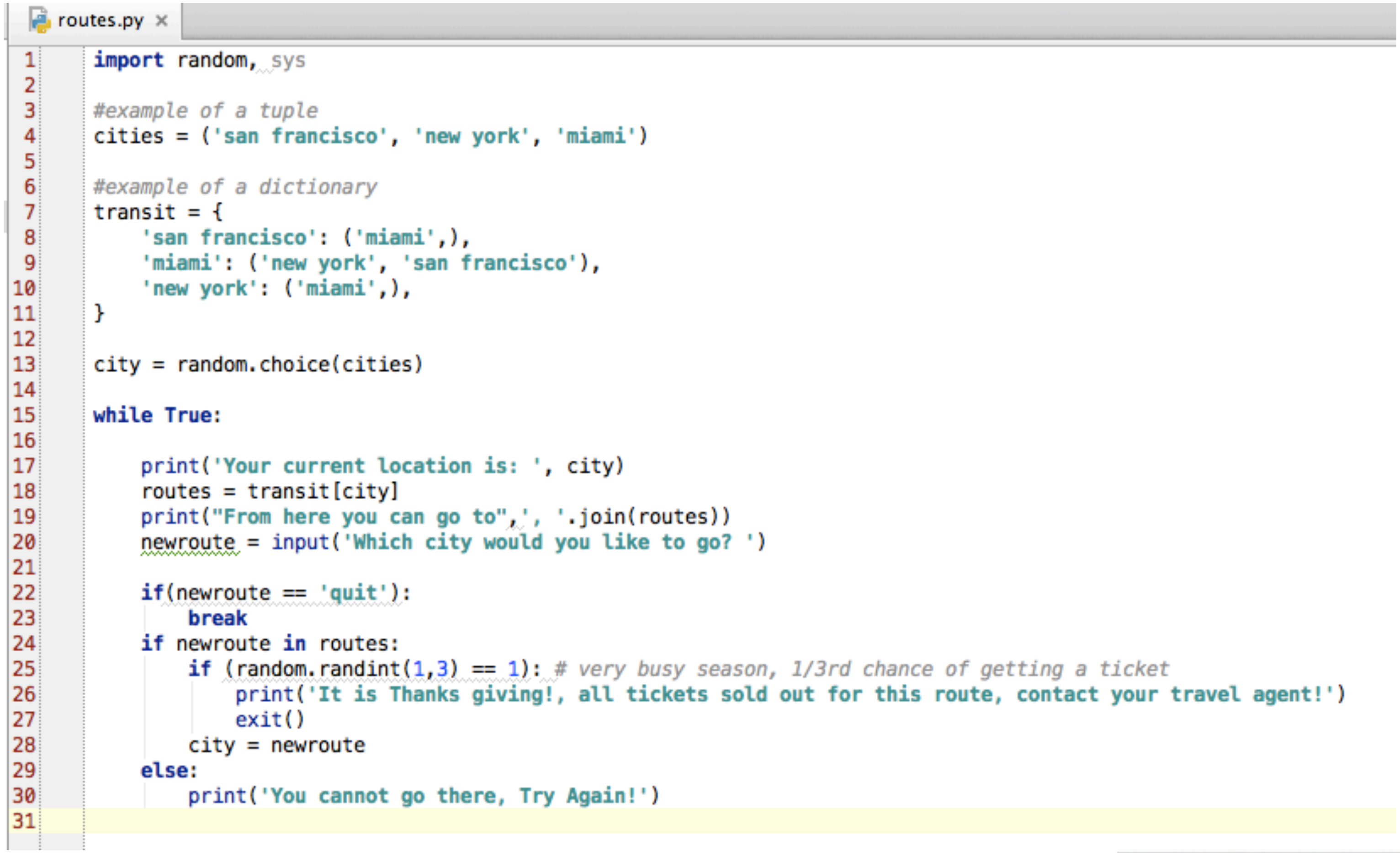
Computer has generated a secret number, can you hack it?



```python
import random

secret = random.randint(1,100) #shh! secret, we need to find this
guess = 0 #starting guess
attempts = 0 #no of attempts we guesses in..

while secret != guess:

    guess = int(input('Guess a Number between 1 and 100: '))
    attempts +=1

    if(secret == guess):
        print('Good Job! You found the secret sauce')
    elif(secret > guess):
        print('Too low, Try Again!')
    else:
        print('Too high, Try Again!')

print('You found the secret', secret, 'in', attempts, 'attempts')
```

Mission: Route Home!

It’s thanks giving weekend, you got to be home for that Turkey!!



```python
import random, sys

#example of a tuple
cities = ('san francisco', 'new york', 'miami')

#example of a dictionary
transit = {
    'san francisco': ('miami',),
    'miami': ('new york', 'san francisco'),
    'new york': ('miami',),
}

city = random.choice(cities)

while True:

    print('Your current location is: ', city)
    routes = transit[city]
    print("From here you can go to",', '.join(routes))
    newroute = input('Which city would you like to go? ')

    if(newroute == 'quit'):
        break
    if newroute in routes:
        if (random.randint(1,3) == 1): # very busy season, 1/3rd chance of getting a ticket
            print('It is Thanks giving!, all tickets sold out for this route, contact your travel agent!')
            exit()
        city = newroute
    else:
        print('You cannot go there, Try Again!')
```