Introduction to Python for Kids!

What do you need to get started?

1. Anaconda 3.6+ free - https://www.continuum.io/downloads

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

Wi-Fi: -

Introduction to Python for Kids!

What do you need to get started?

1. Anaconda 3.6+ free - https://www.continuum.io/downloads

2. Code Download: https://github.com/anooptrivedi/workshops-python-level1

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

Wi-Fi: -

Introduction to Python for Kids!

What do you need to get started?

1. Anaconda 3.6+ free - https://www.continuum.io/downloads

2. Code Download: https://github.com/anooptrivedi/workshops-python-level1/blob/master/PythonLesson2.ipynb

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

Wi-Fi: HD –

Math - Quiz

John thinks about a four digit number which is less than 2000. The ones and hundreds are even numbers. The sum of four numbers is 20. The number at ones place is twice the thousands place.

Can you guess the number?

Math - Quiz

- Number is more than 1000 and less than 2000
- Addition of four numbers, say, $a + b + c + d = 20$

  - Thousand_Place = odd (o)
  - Hundred_Place = even (e)
  - Tens_Place = odd (o)
  - Ones_Place = even (e)

- O E O E

Math – Quiz – Solution

```
mylist = [0,1,2,3,4,5,6,7,8,9]
MAX = 10
total = 20

for index in mylist:
    if(mylist[index] > 0):
        count = 2
        while count < MAX:
            if(mylist[count] == 2*mylist[index]):
                if((mylist[index] % 2 != 0) and (mylist[count] % 2 == 0)):
                    if(mylist[index] < 2):
                        thousand_place = mylist[index]
                        ones_place = mylist[count]
                        remaining = total - thousand_place - ones_place

            if(mylist[count] + mylist[index] == remaining):
                if(mylist[count] % 2 == 0):
                    hundred_place = mylist[count]
                    tens_place = mylist[index]
            count += 1

print(thousand_place,hundred_place,tens_place,ones_place)
```

DEVOXX 4KIDS

Who is helping you today?

Anoop Trivedi: Instructor

Pradeep Bhatter: Co-Instructor

Tori: Logistics

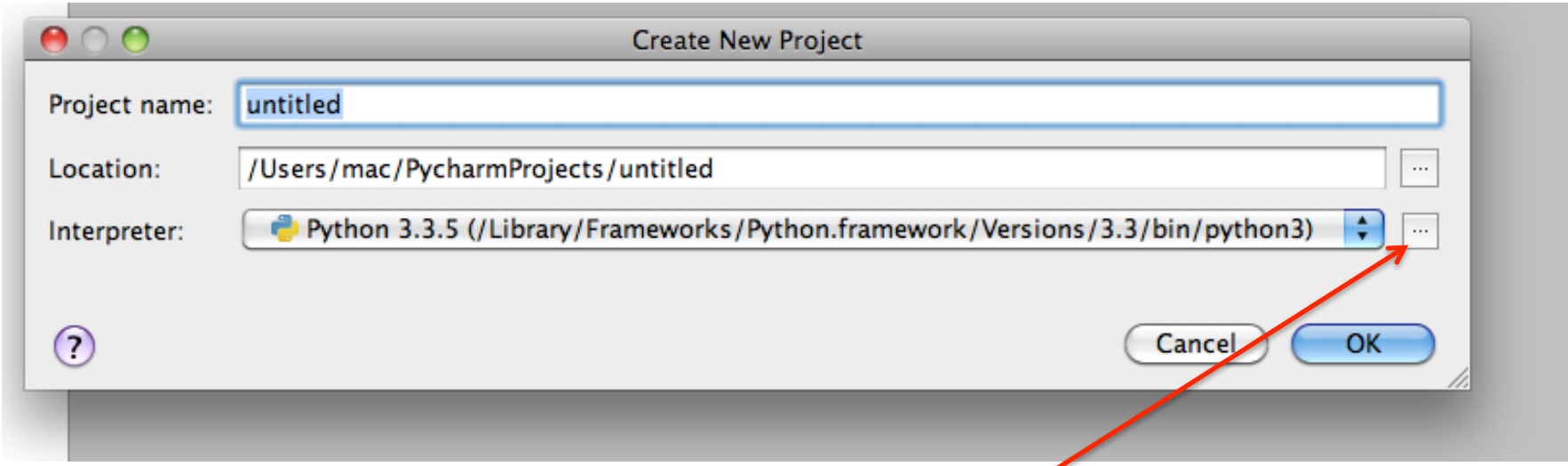
Vivian: Facilities

What are we going to do today ?

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

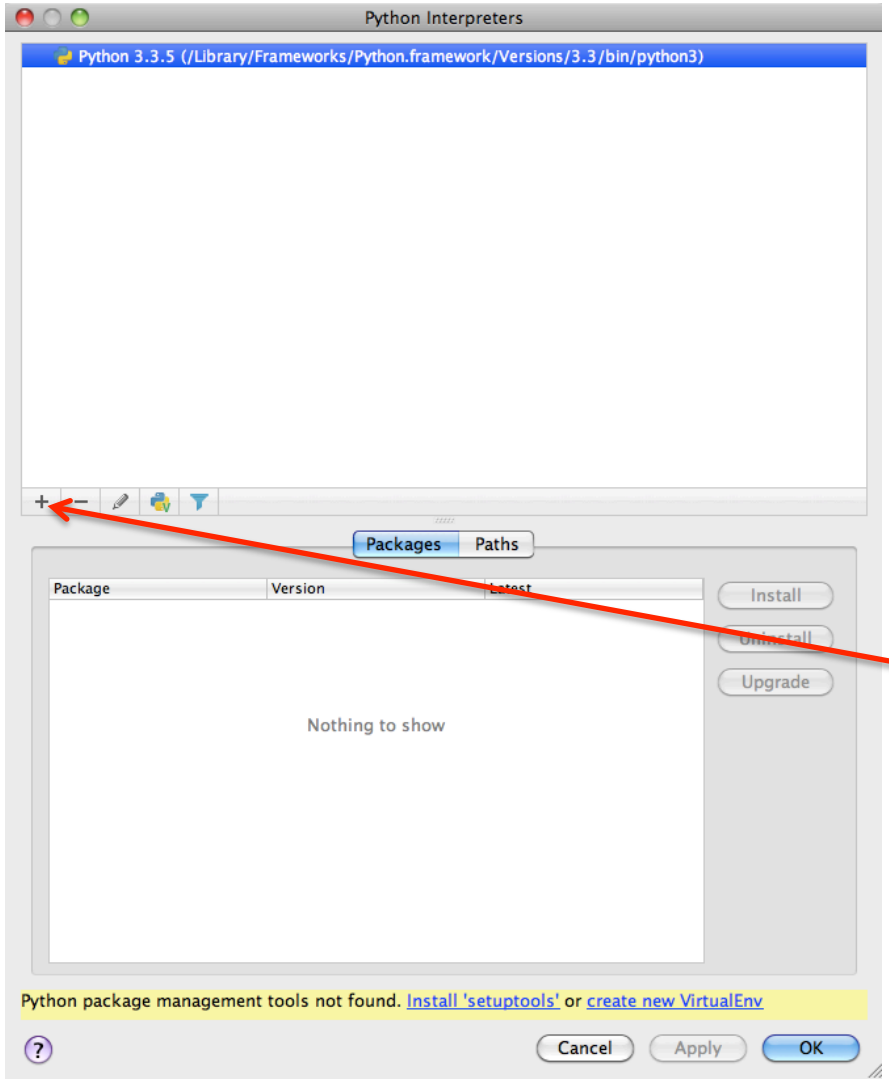
Python Setup:

Create New Project in PyCharm



Click here to choose your interpreter

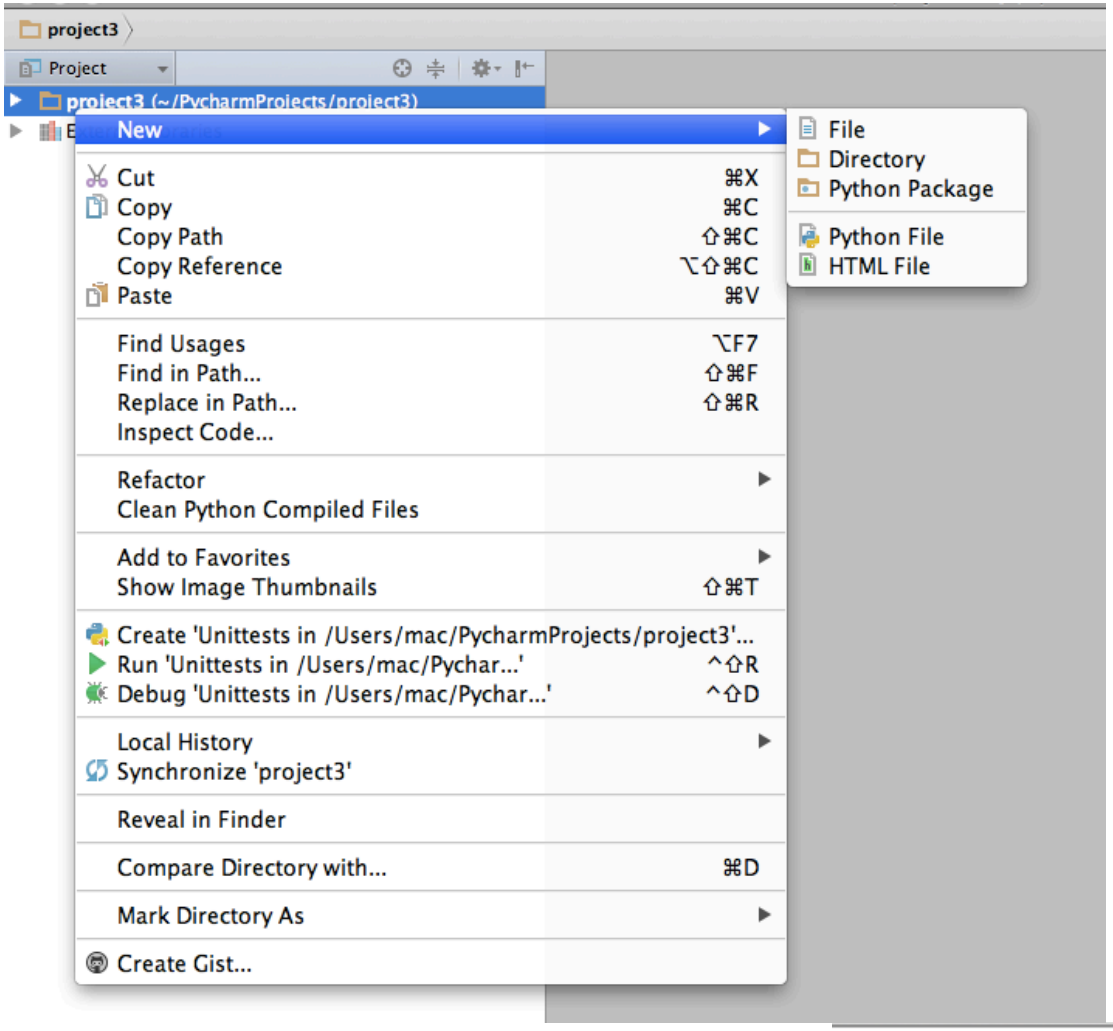
Python Setup:



Click on + sign to choose interpreter

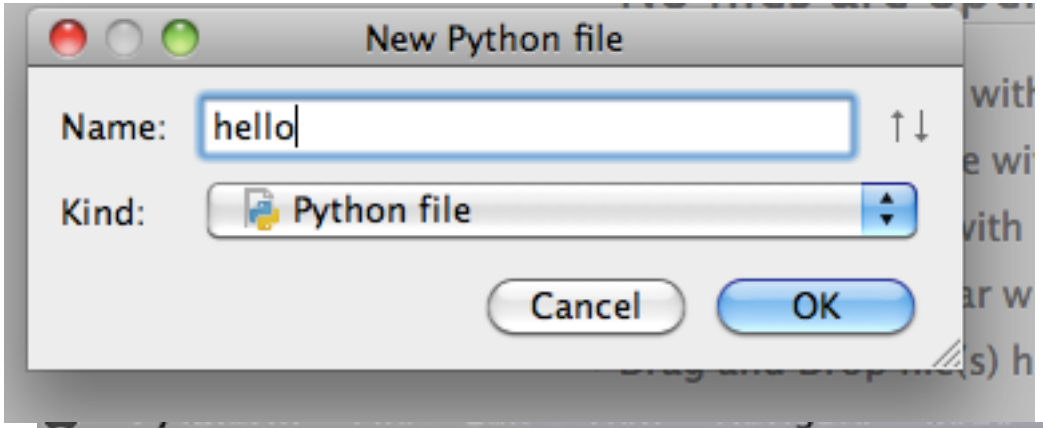
DEVOXX 4KIDS

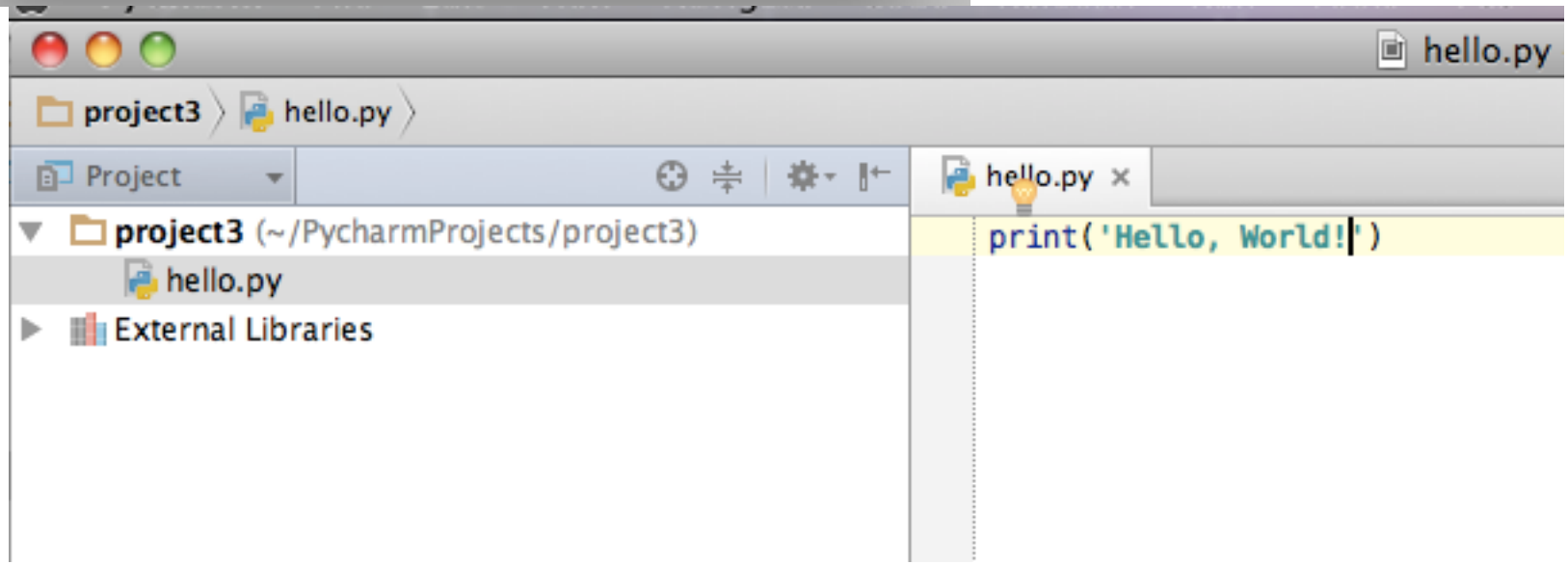
Hello World!



DEVOXX 4KIDS

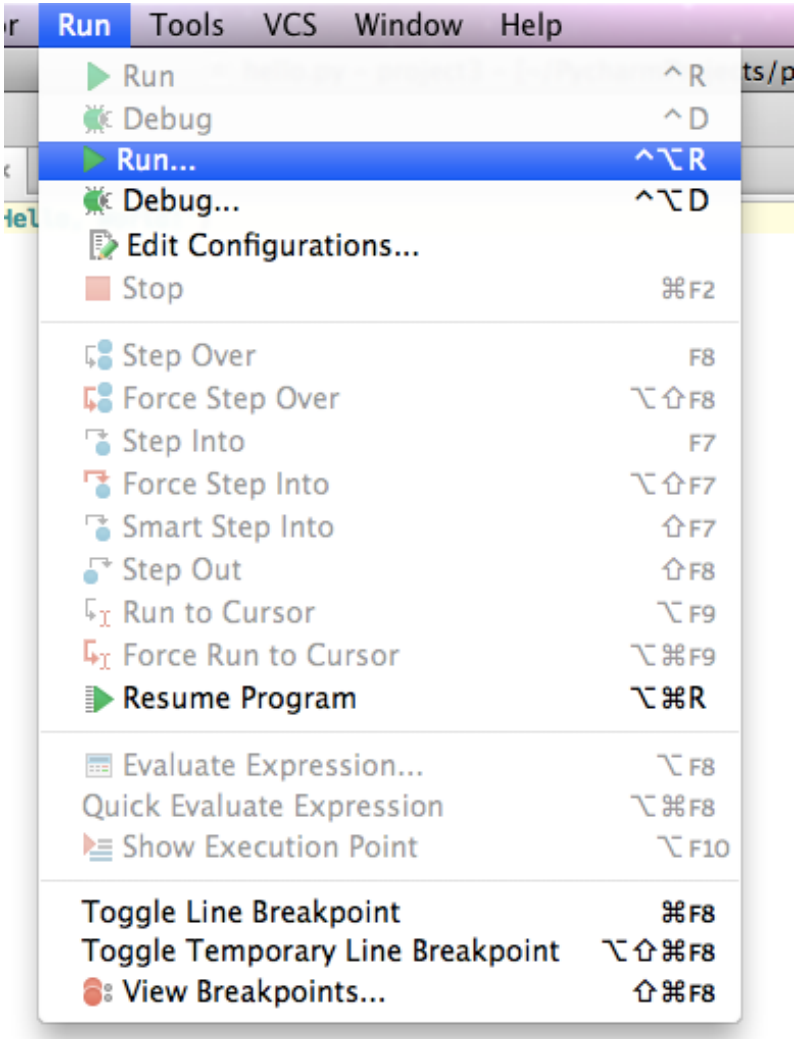
Hello World!





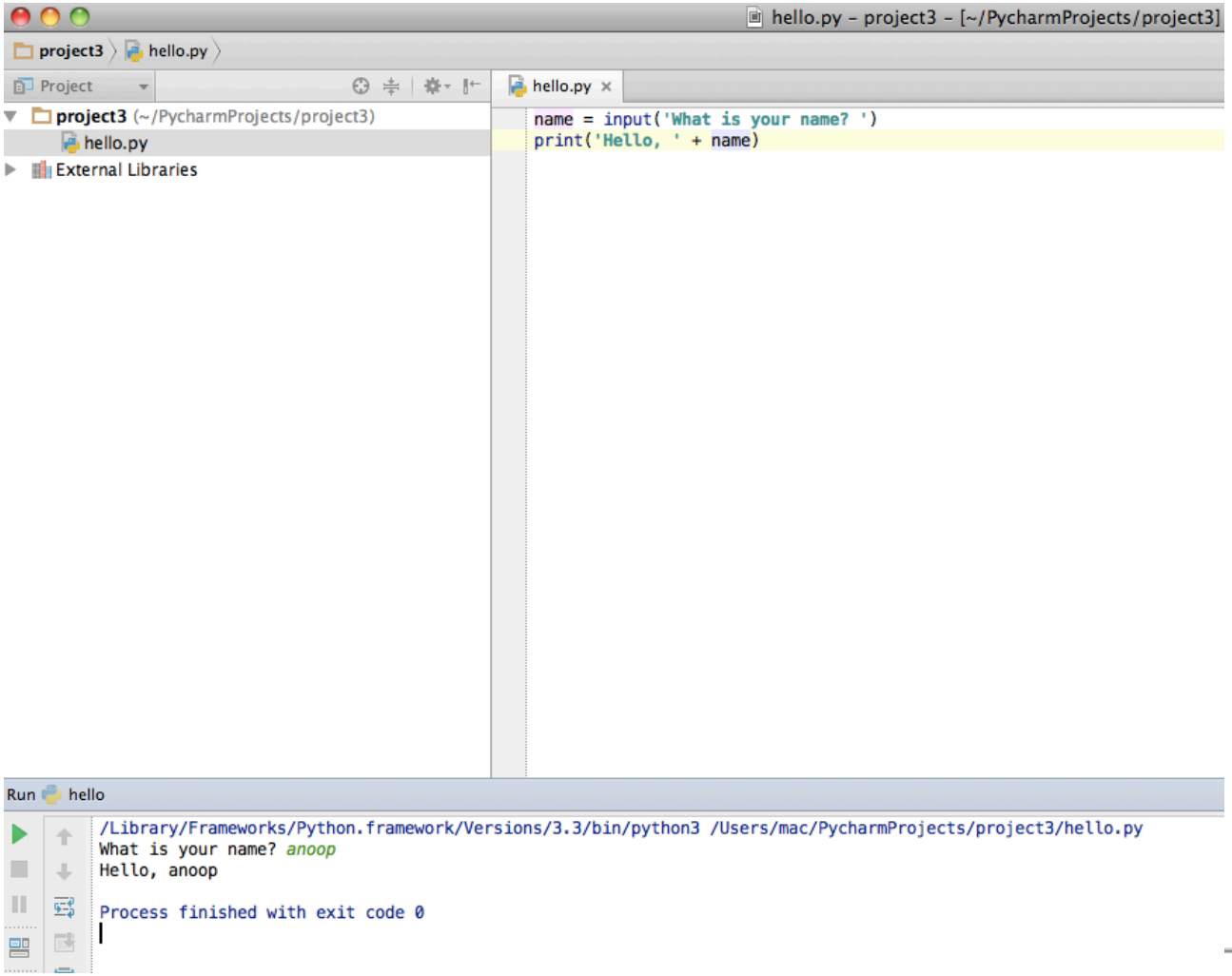
DEVOXX 4KIDS

Hello World!



DEVOXX 4KIDS

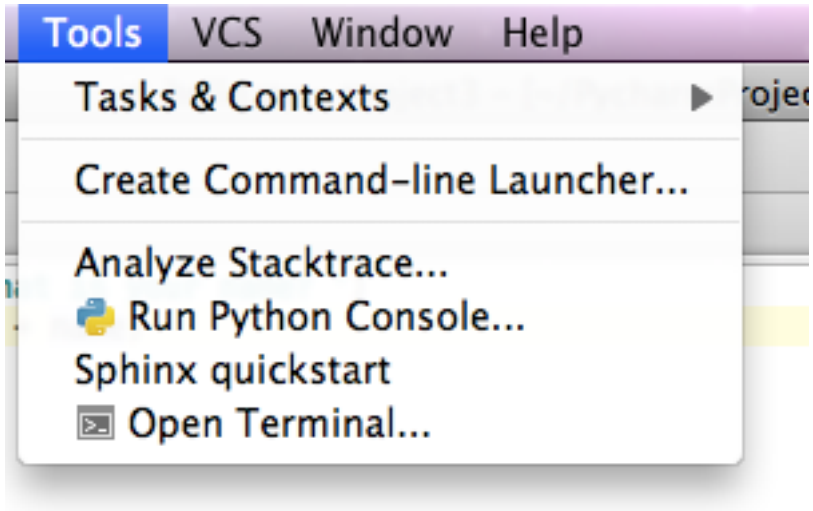
Voila: Hello World!

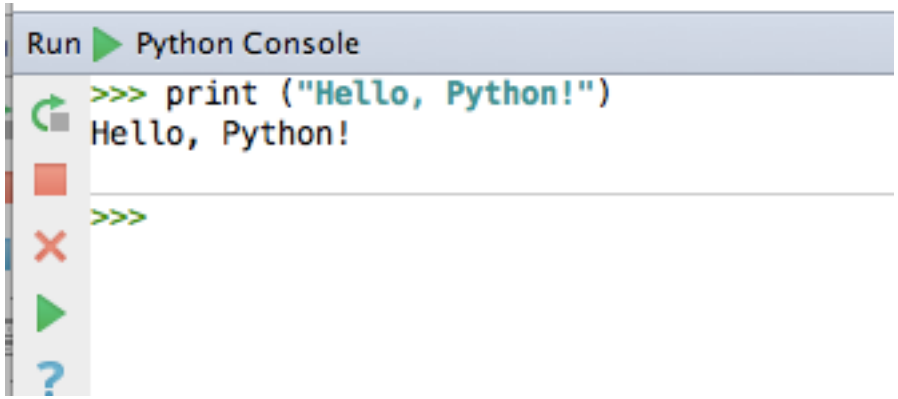


name is a variable

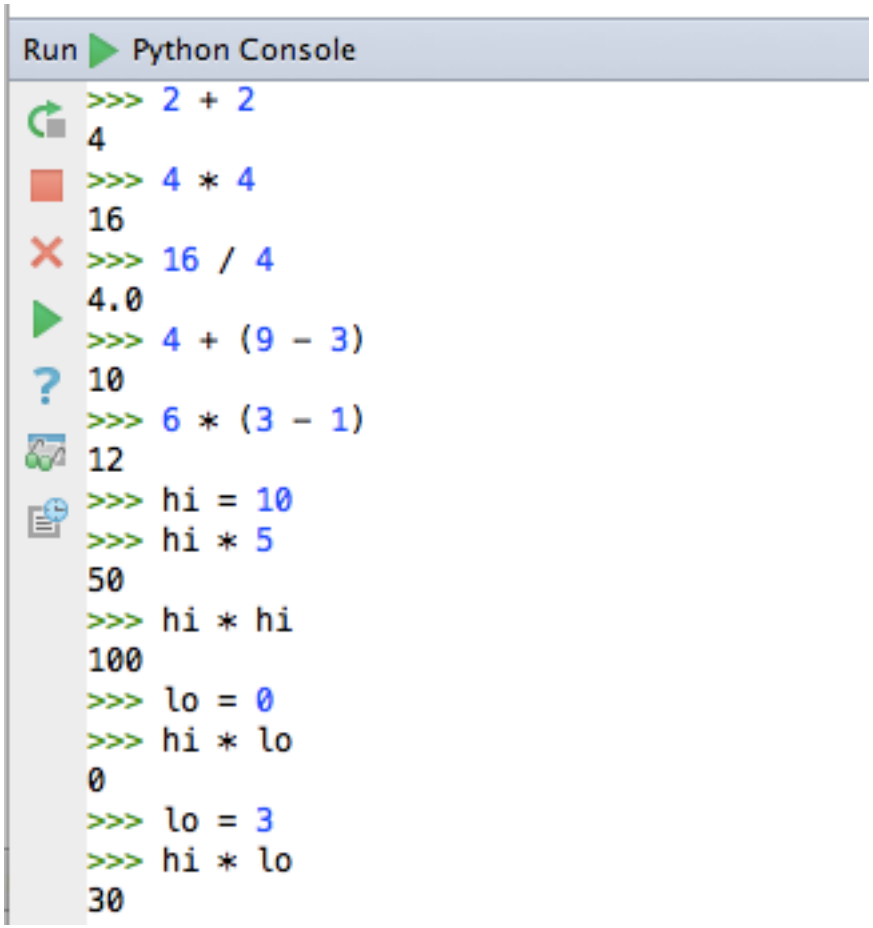
DEVOXX 4KIDS

Python Console:

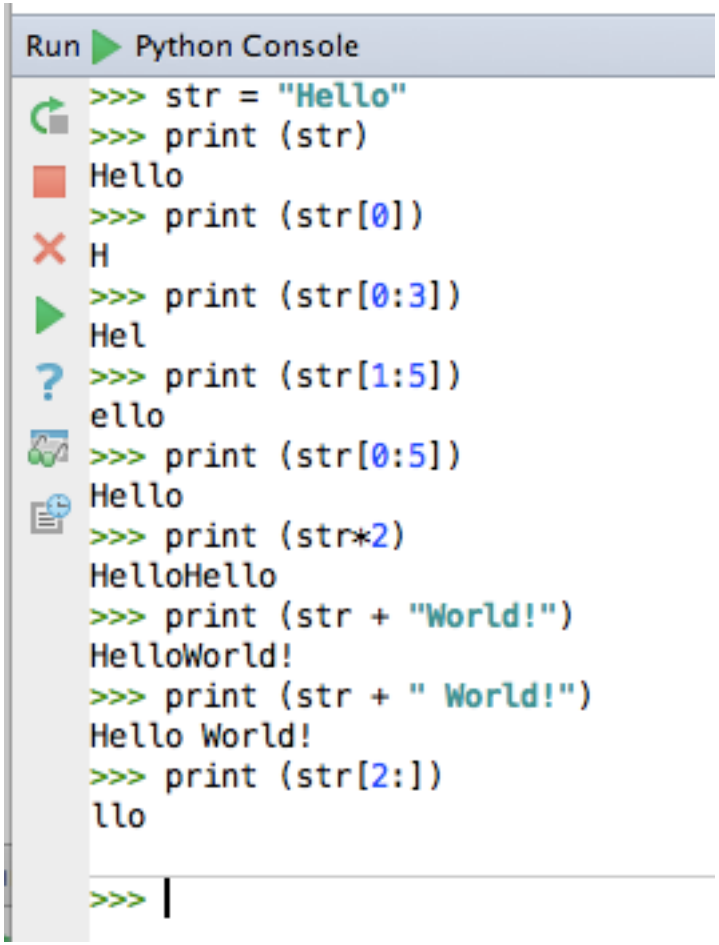




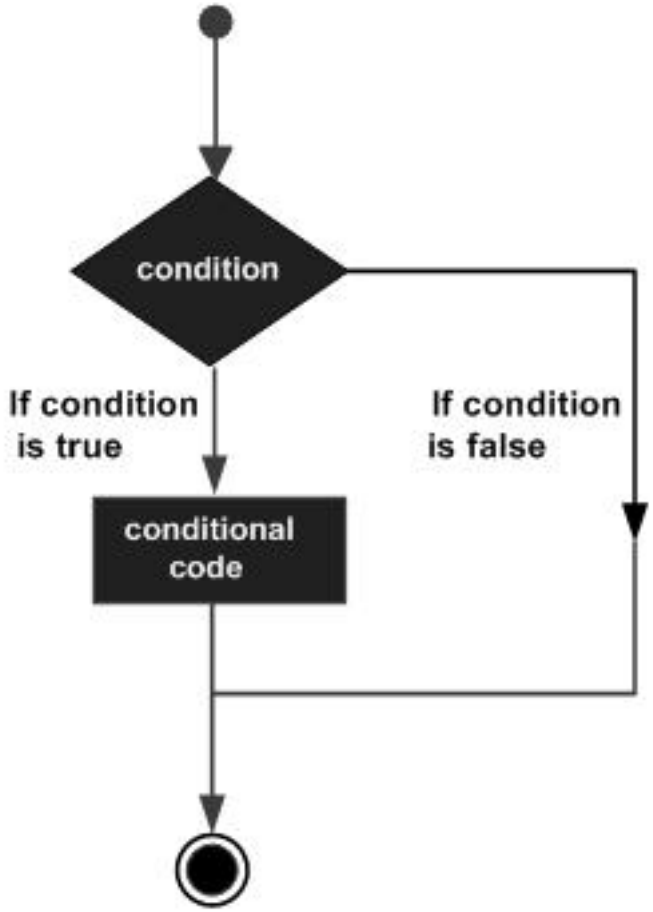
More Python Console Examples:



More Python Console Examples:



Conditional Statements



grades.py

```
grade = 10

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

Conditional Statements

grades.py

```
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

string.py

```
name = 'Anoop'

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

```
name = 'Anoop'

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

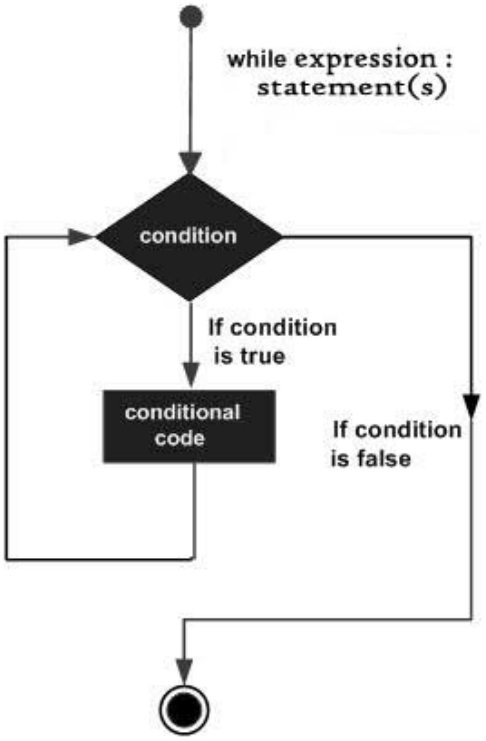
string.py

```
name = 'ANOOP'

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

DEVOXX 4KIDS

Conditional Statements



while.py

```
1  count = 0
2
3  while(count < 5):
4      print('The count is: ', count)
5      count = count + 1
6
7  print('This program is done!')
8
```

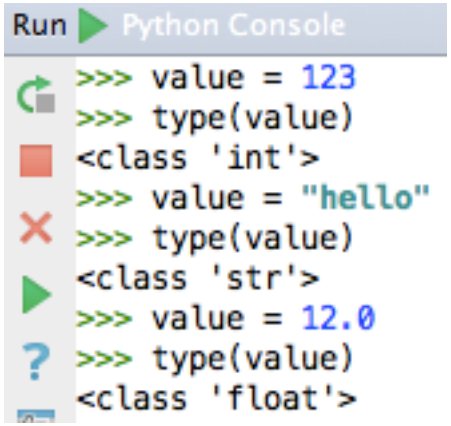
while.py

```
1  while(True):
2      count = int(input('Please enter a number: '))
3      if (count >= 5):
4          print(count, ' is more than or equal to 5')
5      else:
6          print(count, ' is less than 5')
7          exit()
8
9
```

Python Types

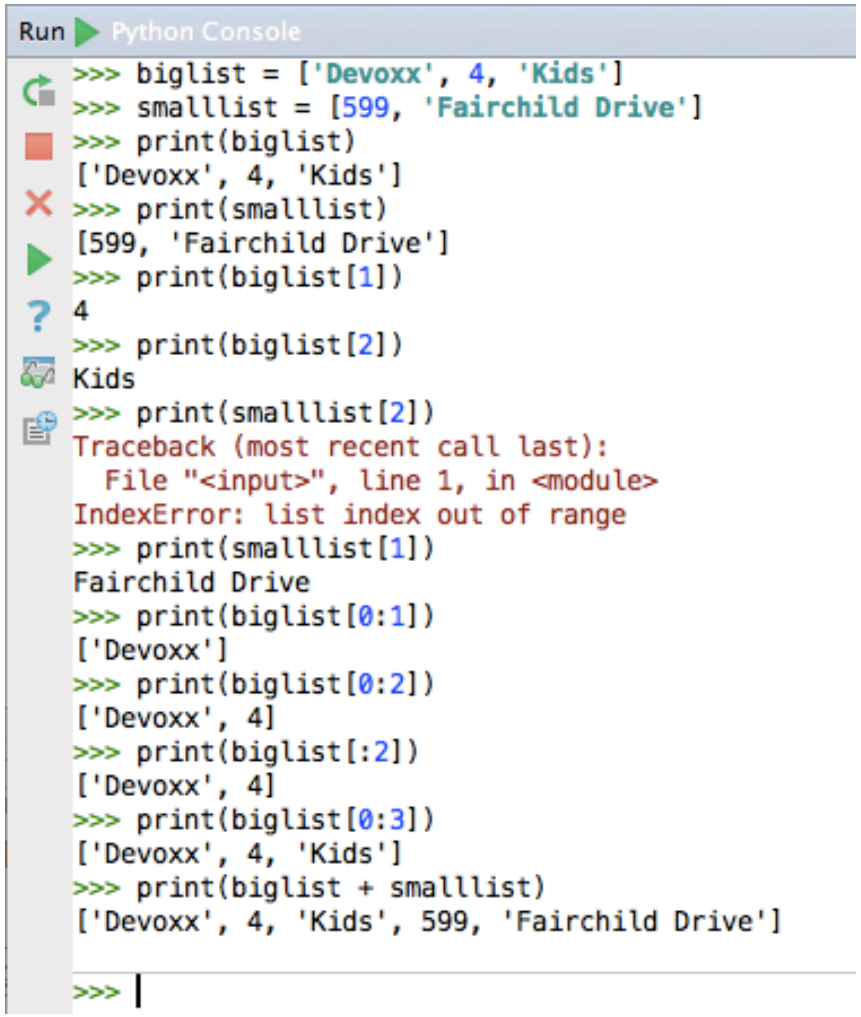
1. Numbers : (int, long, float, complex)
2. Strings
3. Lists
4. Tuple
5. Dictionary

int, float, and string



List

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

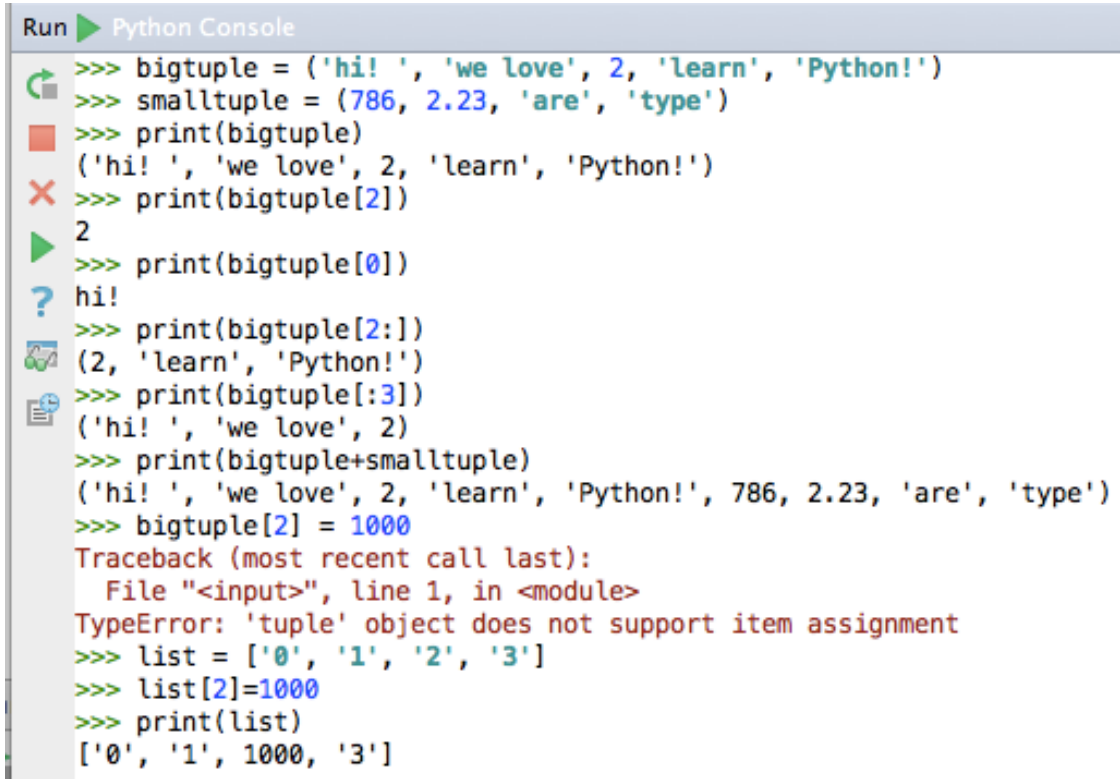


DEVOXX 4KIDS

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"

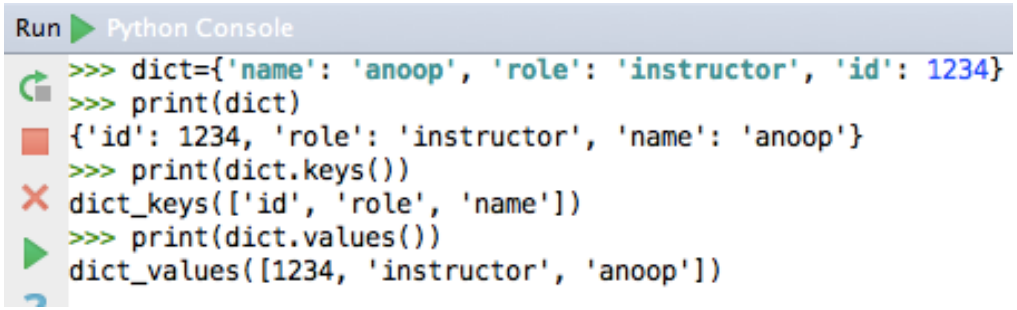


DEVOXX 4KIDS

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."



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.

tempconvert.py

```
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?

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

cointoss.py

```
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')
```

Mission: Find my secret number?

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

guess.py

```
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!!

routes.py

```
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!')

```

DEVOXX 4KIDS