



# Python : An Introduction

Kannappan S (mapquest, Aol.)

PyCon India 2010

# History

- Guido Van Rossum
- 1991
- Python Software Foundation
- [python.org](https://python.org)

# Think !

*“The real purpose of education is not the learning of facts but training the mind to think”*

*- Albert Einstein*

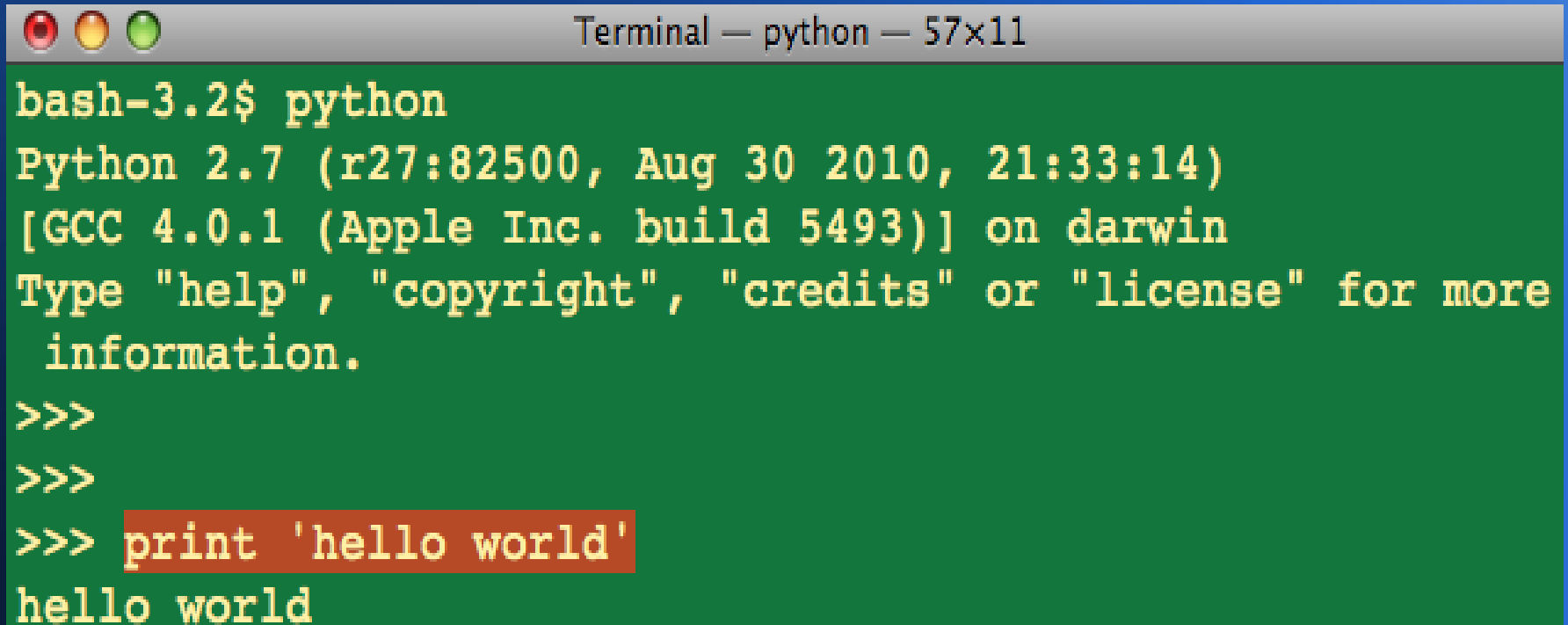
WHAT DO YOU

NEED

FROM A

PROGRAMMING LANGUAGE ?

# Hello World

A screenshot of a macOS Terminal window. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left, and the text "Terminal — python — 57x11" on the right. The main area of the terminal has a green background and displays the following text in a yellow monospaced font: "bash-3.2\$ python", "Python 2.7 (r27:82500, Aug 30 2010, 21:33:14)", "[GCC 4.0.1 (Apple Inc. build 5493)] on darwin", "Type \"help\", \"copyright\", \"credits\" or \"license\" for more", "information.", ">>>", ">>>", ">>> print 'hello world'", and "hello world". The line ">>> print 'hello world'" is highlighted with a semi-transparent orange background.

```
bash-3.2$ python
Python 2.7 (r27:82500, Aug 30 2010, 21:33:14)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "help", "copyright", "credits" or "license" for more
information.
>>>
>>>
>>> print 'hello world'
hello world
```

# Data Types

Python is dynamically typed

Some important data types

- int
- float
- strings
- lists
- dictionaries

# Numerical Operations

- `>>> 5 + 3`

8

- `>>> 5 - 3`

2

- `>>> 5 * 3`

15

- `>>> 5 / 3`

2

- `>>> 5.0 / 3`

1.6666666666666667

# Numerical Functions

- `>>> import math`
- `>>> math.pow(5,3)`  
125.0
- `>>> math.sqrt(25)`  
5.0
- `>>> math.log(1024, 2)`  
10.0
- `>>> math.factorial(5)`  
120

# Conditional Operations

- `>>> 5 == 5`  
True
- `>>> 5 != 5`  
False
- `>>> 5 > 3`  
True
- `>>> 5 <= 3`  
False
- `>>> 0 < 5 > 3 < 2`  
False



# Strings

- compound data type
- `>>> company1 = 'Apple'`
- `>>> company2 = 'Google'`
- `>>> company1 + company2`  
`'AppleGoogle'`
- `>>> company1[0]`  
`'A'`
- `>>> company2[0:3] + company1[-2]`  
`'Gool'`

# String Functions

- `>>> import string`
- `>>> string.lower('Apple')`  
`'apple'`
- `>>> string.upper('Google')`  
`'GOOGLE'`
- `>>> string.replace('Apple', 'App', 'Peop')`  
`'People'`
- `>>> string.strip('Apple', 'e')`  
`'Appl'`

# Lists

- workhorse of python
- ```
>>> companies = ['Apple', 'Google', 'Yahoo', 'Microsoft', 'AOL']
```
- ```
>>> len(companies)
```

  
5
- ```
>>> companies[-1]
```

  
'AOL'
- ```
>>> newcompanies = ['facebook', 'twitter']
```
- ```
>>> companies + newcompanies
```

  
['Apple', 'Google', 'Yahoo', 'Microsoft', 'AOL', 'facebook', 'twitter']

# List Functions

- ```
>>> newcompanies.append('zynga')  
['facebook', 'twitter', 'zynga']
```
- ```
>>> newcompanies.remove('facebook')  
['twitter', 'zynga']
```
- ```
>>> newcompanies.index('twitter')  
0
```
- ```
>>> newcompanies.reverse()  
['zynga', 'twitter']
```
- ```
>>> 'twitter' not in newcompanies  
False
```

# Dictionaries

- `>>> english2french = {}`
- `>>> english2french['hello'] = 'bonjour'`
- `>>> english2french['goodbye'] = 'adieu'`
- `>>> print english2french`  
`{'hello' : 'bonjour', 'goodbye' : 'adieu'}`
- `>>> english2french['thanks'] = 'merci'`
- `>>> len(english2french)`  
`3`
- `>>> del english2french('goodbye')`  
`{'hello' : 'bonjour', 'thanks' : 'merci'}`

# Dictionary Functions

- `>>> english2french.keys()`  
`['hello', 'thanks']`
- `>>> english2french.values()`  
`['bonjour', 'merci']`
- `>>> english2french.items()`  
`[ ('hello', 'bonjour'), ('thanks', 'merci') ]`
- `>>> english2french.has_key( 'love' )`  
`False`

# Loops

- indentation is a must in python
- ```
>>> for i in range(2,4):  
    >>>     print i  
  
2  
  
3
```
- ```
>>> newcompanies = ['facebook', 'twitter']
```
- ```
>>> for company in newcompanies:  
    >>>     print company  
  
facebook  
  
twitter
```

# Loops

```
>>> count = 1
```

```
>>> while count <= 5 :
```

```
>>>     print count
```

```
>>>     count += 1
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```



# Functions

- ```
>>> def addHundred(a):  
    >>>     return a + 100
```
- ```
>>> addHundred(8)  
108
```
- ```
>>> def summation(a,b):  
    >>>     return a+b
```
- ```
>>> summation(5,10)  
15
```
- ```
>>> def isOdd(a):  
    >>>     return a%2
```

# Functional Aspects

- `>>> list1 = [16,23,36]`
- `>>> map(addHundred, list1)`  
116  
123  
136
- `>>> reduce(summation, list1)`  
75
- `>>> filter(isOdd, list1)`  
23

# small program

```
#!/usr/bin/env python
# This program generates fibonacci sequence
def fib(n):
    if n == 0 or n == 1 :
        return n
    else :
        return fib(n-1) + fib(n-2)
def main():
    for num in range(1..10) : print fib(num)
main()
```

1 1 2 3 5 8 13 21 34

# Links

- How to think like a Computer Scientist

<http://www.greenteapress.com/thinkpython/thinkCSpy/>

- Google University Video

<http://code.google.com/edu/languages/google-python-class/>