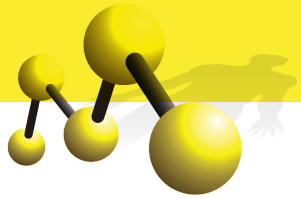
Introduction to Python Programming

Luis Pedro Coelho

Programming for Scientists

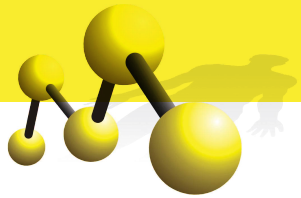
October 22, 2012

Python



Let's digress for a moment discussing the language...

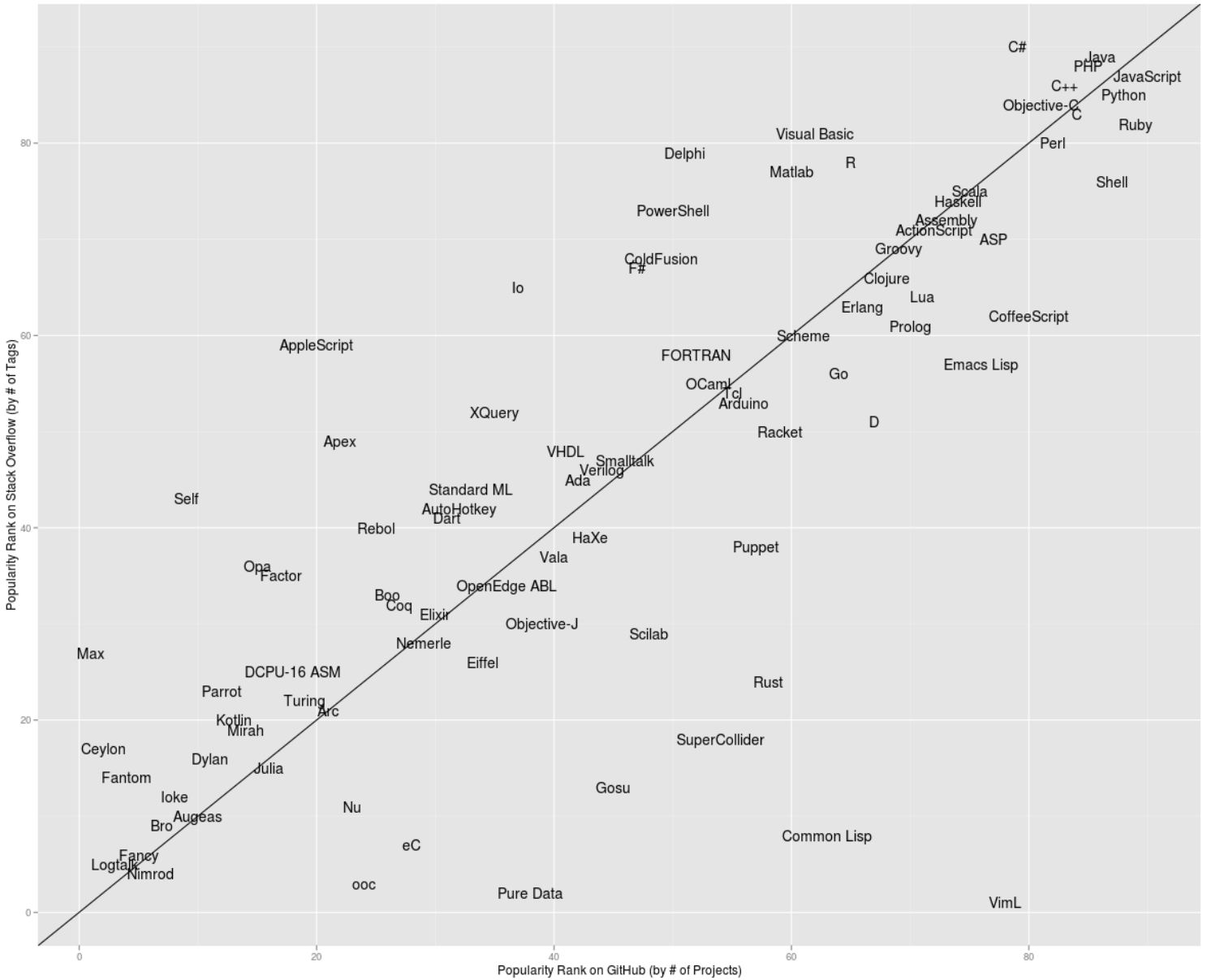
Python Language History



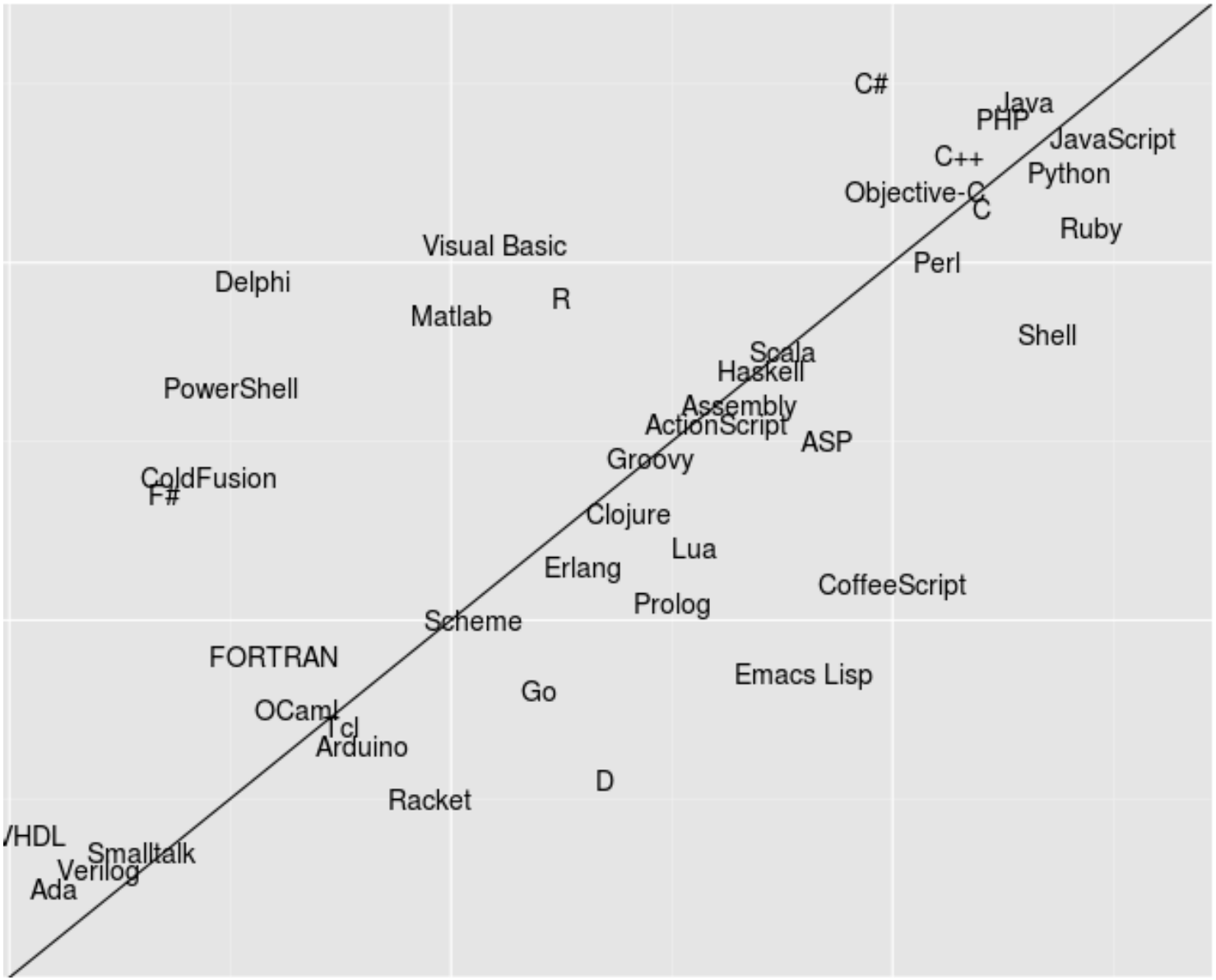
History

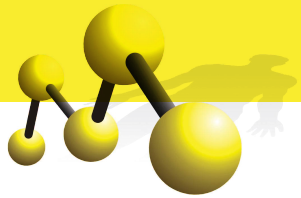
- Python was started in the late 80's.
- It was intended to be both easy to teach and industrial strength.
- It is (has always been) open-source.
- In the last 10 years, it has become one of the most widely used languages (top 10).

Popularity


C#
Java
PHP
JavaScript
C++
Python
Objective-C
C
Ruby
Visual Basic
Perl
Delphi
R
Matlab
Shell
Scala
Haskell
PowerShell
Assembly
ActionScript
ASP
Groovy
ColdFusion
F#
Clojure
Io
Lua
Erlang
CoffeeScript
Prolog
Scheme
AppleScript
FORTRAN
Emacs Lisp
Go
OCaml
Tcl
Arduino
XQuery
D
Racket
Apex
VHDL
Smalltalk
Verilog
Ada
Standard ML
Self
AutoHotkey
Dart
Rebol
HaXe
Puppet
Vala
 Opa
Factor
OpenEdge ABL
Boo
Coq
Elixir
Objective-J
Scilab
Nemerle
Max
Eiffel
DCPU-16 ASM
Rust
Parrot
Turing
Arc
Kotlin
Mirah
SuperCollider
Ceylon
Dylan
Julia
Fantom
Gosu
Ioke
Nu
Augeas
Bro
Common Lisp
eC
Fancy
Logtalk
Nimrod
ooc
Pure Data
VimL
0
20
40
60
80
Popularity Rank on Stack Overflow (by # of Tags)
Popularity Rank on GitHub (by # of Projects)

Popularity

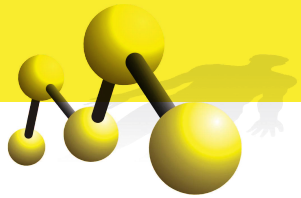

C#
Java
PHP
JavaScript
C++
Python
Objective-C
C
Ruby
Visual Basic
Perl
Delphi
R
Matlab
Shell
Scala
Haskell
PowerShell
Assembly
ActionScript
ASP
Groovy
ColdFusion
F#
Clojure
Lua
Erlang
CoffeeScript
Prolog
Scheme
FORTRAN
Emacs Lisp
Go
OCaml
Tcl
Arduino
D
Racket
VHDL
Smalltalk
Verilog
Ada



Python Versions

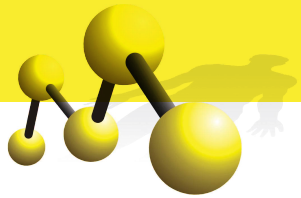
- The current versions of Python are 2.7 and 3.3
- This class assumes you have 2.6–2.7
- There are some small differences when compared to version 3.x

What is a Computer?



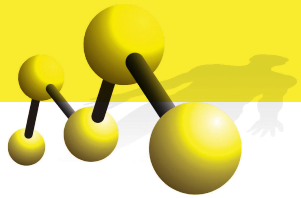
1. Memory
2. Processor
3. Magic

Python Model



1. Objects
2. Operations on objects
3. Magic

Python Example

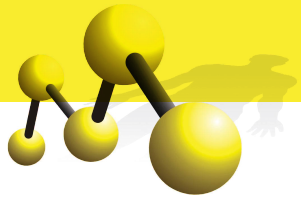


```
print "Hello World"
```

Running Python

1. From a file
2. Interactively

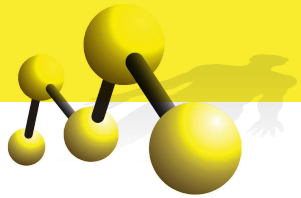
Computer Program



helloword.py

```
print 'Hello World'
```

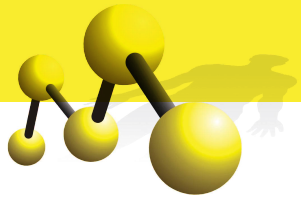
Running a Program



1. Shell
2. IDE

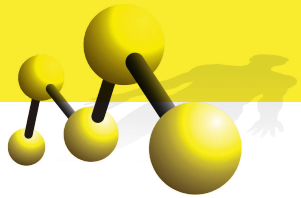
Let me show you a demonstration...

More Complex Example



What is 25 times 5?

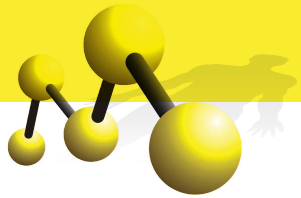
More Complex Example



What is 25 times 5?

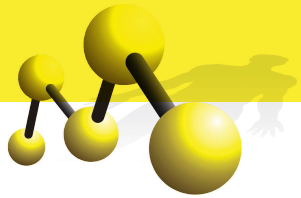
```
print 25 * 5
```

More Complex Example

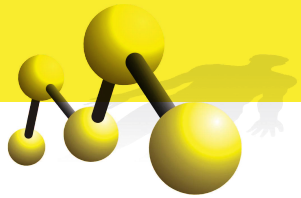


```
name = 2
other = 3
yetanother = name + other
name = 5
print yetanother + name
```

Blackboard demonstration

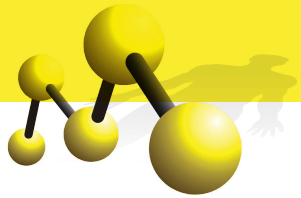


Conditionals



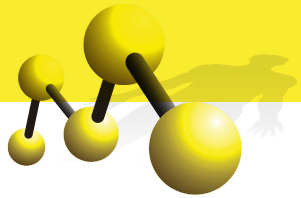
```
if <condition>:
    <statement 1>
    <statement 2>
else:
    <statement 3>
```

Lists



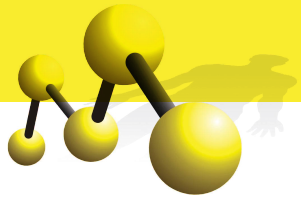
```
students = ['Luis','Mark','Rita']

print students[0]
print students[1]
print students[2]
```

Loops



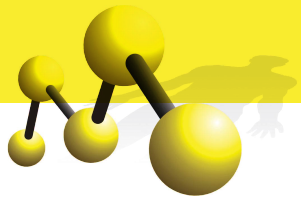
```
students = ['Luis','Mark','Rita',...]

for st in students:
    print st
```

Example



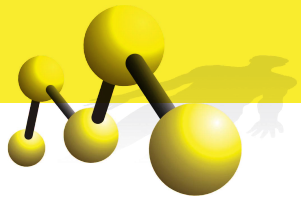
```
values = [0.11, -0.23, -0.16, 0.18, 0.23, 0.19]

sum = 0.0
sum2 = 0.0
for v in values:
    sum = sum + v
    sum2 = sum2 + v * v

mu = sum/len(values)
mu2 = sum2/len(values)
print 'Average: {0}'.format(mu)
print 'Std Dev: {0}'.format(mu2 - mu*mu)
```

Example



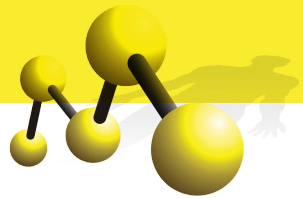
```
values = [0.11, -0.23, -0.16, 0.18, 0.23, 0.19]

sum = 0.0
sum2 = 0.0
for v in values:
    sum += v
    sum2 += v * v

mu = sum/len(values)
mu2 = sum2/len(values)
print 'Average: {0}'.format(mu)
print 'Std Dev: {0}'.format(mu2 - mu*mu)
```

Example



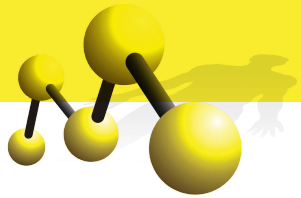
```
values = [0.11, -0.23, -0.16, 0.18, 0.23, 0.19]

mu = 0.0
mu2 = 0.0
for v in values:
    mu += v
    mu2 += v * v

mu /= len(values)
mu2 /= len(values)
print 'Average: {0}'.format(mu)
print 'Std Dev: {0}'.format(mu2 - mu*mu)
```

Example



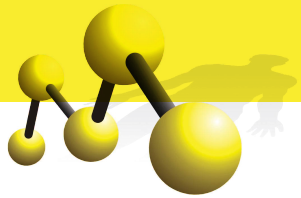
```
values = [0.11, -0.23, -0.16, 0.18, 0.23, 0.19]

mu = 0.0
mu2 = 0.0
for v in values:
    mu += v
    mu2 += v * v

mu /= len(values)
mu2 /= len(values)
print 'Average: {0}'.format(mu)
print 'Std Dev: {0}'.format(mu2 - mu*mu)
```

Exercise



Adapt the code to ignore negative numbers.

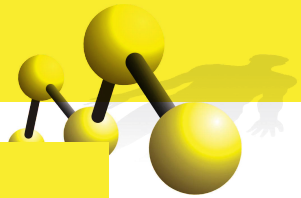
Exercise



Adapt the code to ignore negative numbers.

```
values = [0.11, -0.23, -0.16, 0.18, 0.23, 0.19]

mu = 0.0
mu2 = 0.0
n = 0.0
for v in values:
    if v >= 0.0:
        mu += v
        mu2 += v * v
        n += 1

mu /= n
mu2 /= n
print 'Average: {0}'.format(mu)
print 'Std Dev: {0}'.format(mu2 - mu*mu)
```

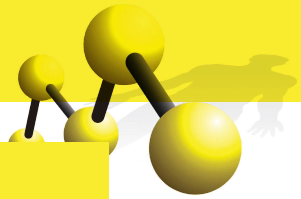
Loops (II)



Greatest Common Divisor (Euclid's Method)

$$\gcd(a, b) = \begin{cases} a & \text{if } b = a \\ \gcd(a - b, b) & \text{if } a > b \\ \gcd(a, b - a) & \text{o.w.} \end{cases}$$

Loops (II)

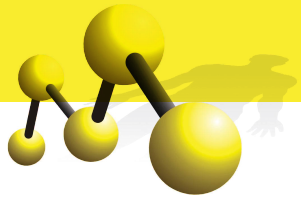


Greatest Common Divisor (Euclid's Method)

$$\gcd(a, b) = \begin{cases} a & \text{if } b = a \\ \gcd(a - b, b) & \text{if } a > b \\ \gcd(a, b - a) & \text{o.w.} \end{cases}$$

```
a = 9344
b = 6497

while a != b:
    if a > b:
        a,b = a-b,b
    else:
        a,b = a,b-a
print a
```

For Monday



- Install Python(x,y)
  (or the equivalent on your platform)