



python<sup>TM</sup>

two snakes enter  
one snake leaves?

Anthony Baxter

two snakes enter  
one snake leaves?

Anthony Baxter

[anthony@python.org](mailto:anthony@python.org)

snake one:



python 2.6



snake two:



# python 3.0

IMAGE  
PLACEHOLDER

ADD SOMETHING HERE!!!!



# python 3.0 aka Python 3000

IMAGE  
PLACEHOLDER

ADD SOMETHING HERE!!!!





python 3.0  
aka Python 3000  
aka Py3k

IMAGE  
PLACEHOLDER

ADD SOMETHING HERE!!!!



python 3.0  
aka Python 3000  
aka Py3k  
aka p3yk

IMAGE  
PLACEHOLDER

ADD SOMETHING HERE!!!!



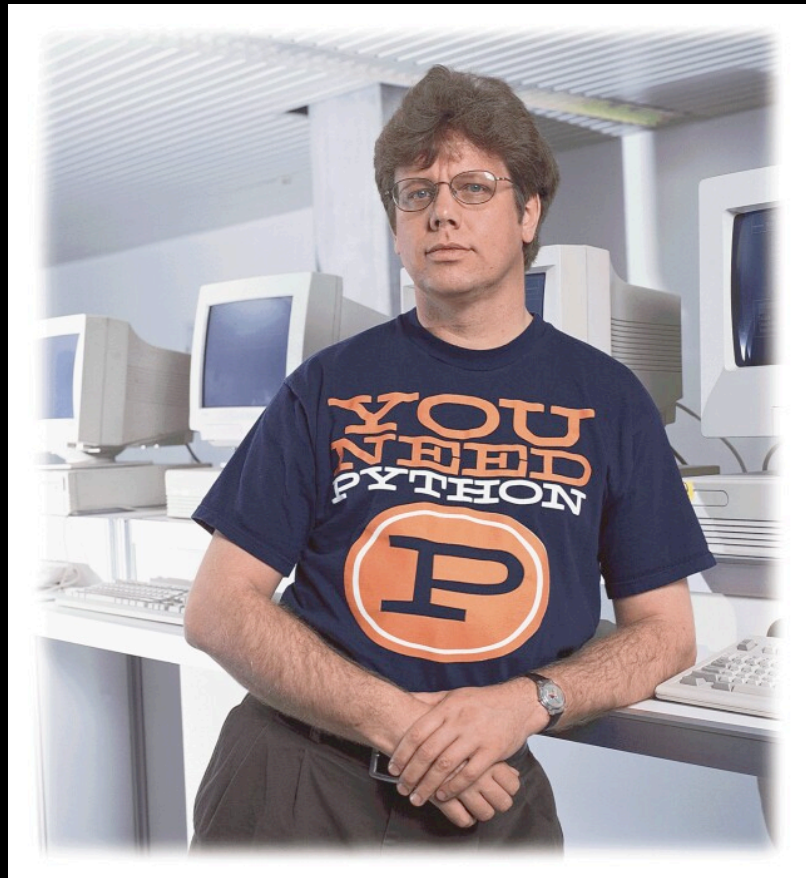
python 3.0  
aka Python 3000  
aka Py3k  
aka p3yk  
... but not Python 90210

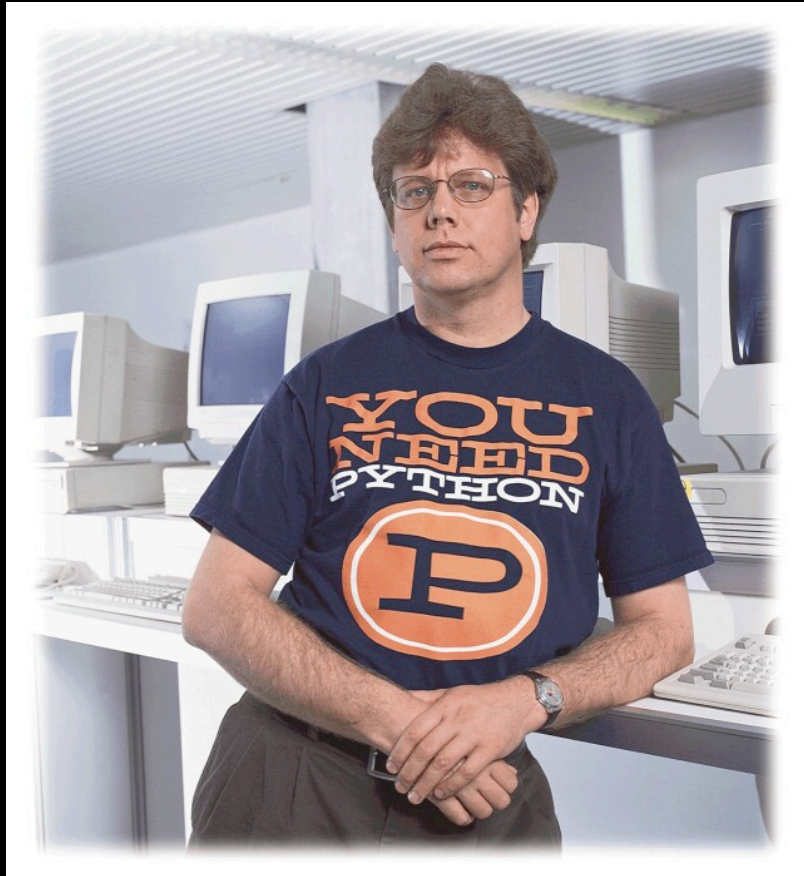
IMAGE  
PLACEHOLDER

ADD SOMETHING HERE!!!!



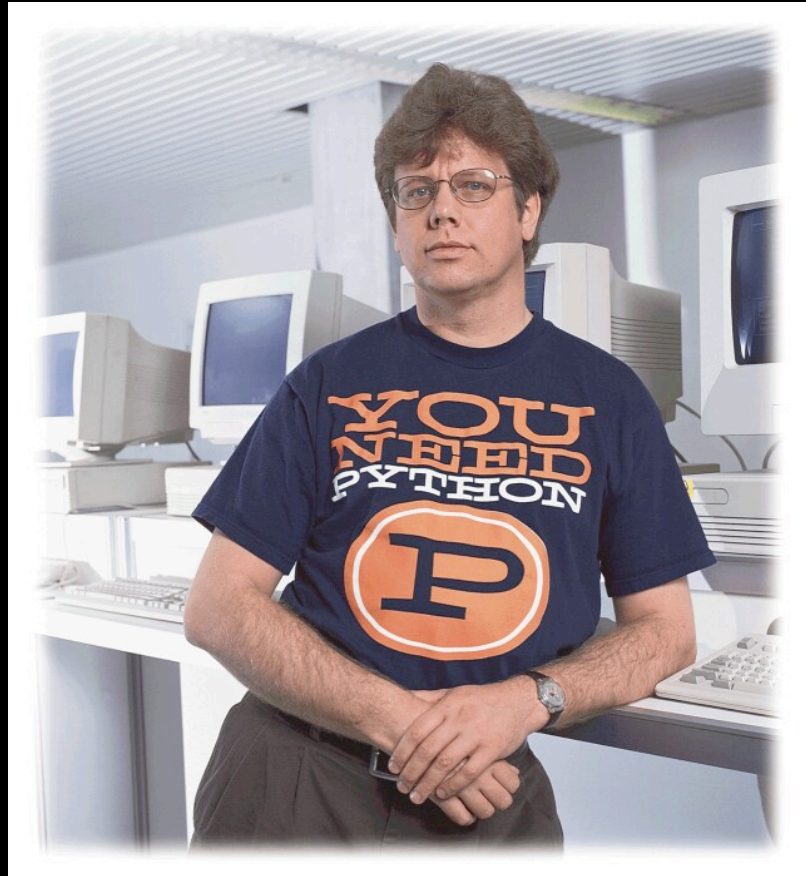






!=





!=



3.0





wherein we break all your  
code



in a good cause



promise!





would this tiny kitten lie to  
you?

wait



did he say



did he say  
“break all our code?”



yes





yes  
sorry



backwards incompatible



rather a lot



imprecise



imprecise  
more precisely:



almost every program  
will need changes



~~almost~~ every program  
will need changes



some big changes



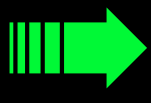


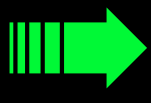


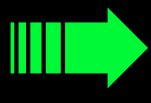


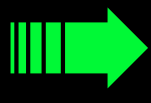
print → print()

print  print()  
unicode strings default

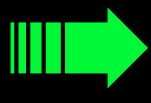
print  print()  
unicode strings default  
byte strings

print  print()  
unicode strings default  
byte strings  
set literals

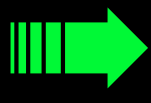
print  print()  
unicode strings default  
byte strings  
set literals  
set, dict comprehensions

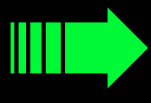
print  print()  
unicode strings default  
byte strings  
set literals  
set, dict comprehensions  
function annotations



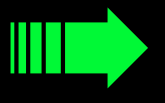
print  print()  
unicode strings default  
byte strings  
set literals  
set, dict comprehensions  
function annotations  
new style division



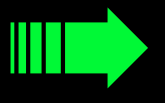
print  print()  
unicode strings default  
byte strings  
set literals  
set, dict comprehensions  
function annotations  
new style division  
iterators everywhere

print  print()  
unicode strings default  
byte strings  
set literals  
set, dict comprehensions  
function annotations  
new style division  
iterators everywhere  
dictionary views



print  print()  
unicode strings default  
byte strings  
set literals  
set, dict comprehensions  
function annotations  
new style division  
iterators everywhere  
dictionary views  
new I/O layer



print  print()  
unicode strings default  
byte strings  
set literals  
set, dict comprehensions  
function annotations  
new style division  
iterators everywhere  
dictionary views  
new I/O layer



std library reorg





# PEP 0008

# PEP 0008

## BaseHTTPServer





PEP 0008  
BaseHTTPServer  
cPickle



# PEP 0008

BaseHTTPServer

cPickle

cStringIO

# PEP 0008

BaseHTTPServer

cPickle

cStringIO

BAD

BAD

# PEP 0008

BaseHTTPServer

cPickle

BAD

cStringIO

BAD

BAD

stuff going away, too



stuff going away, too  
old style classes



stuff going away, too  
old style classes  
string exceptions



stuff going away, too  
old style classes  
string exceptions





stuff going away, too  
old style classes  
string exceptions

<>

dict.has\_key()

stuff going away, too  
old style classes  
string exceptions

<>

dict.has\_key()  
apply()

stuff going away, too  
old style classes  
string exceptions

<>

dict.has\_key()  
apply()  
reduce()

stuff going away, too  
old style classes  
string exceptions

<>

dict.has\_key()

apply()

reduce()

coerce()

stuff going away, too  
old style classes  
string exceptions

<>

dict.has\_key()

apply()

reduce()

coerce()

str.find()

BUT WAIT



it's not that bad



it's not that bad





it's not that bad



2.x is not going away



2.x is not going away



2.6



2.6

2.7



plus



plus  
migration tools







2to3  
-Wpy3k  
from future import ...



more on those later...



back to basics:



back to basics:  
why?



python is 16 years old



python is 16 years old  
first released 1991



like all 16 year olds



it has some annoying features





(all languages have them)



(all languages have them)  
even the simplest



<blink>, anyone?



</blink>

all languages have warts



an example



an example  
(with visual)







`backticks` for repr()





``backticks` for repr()`

in addition



Python breaks  
it's own rule:



Python breaks  
it's own rule:  
TOOWTDI



there's  
only  
one  
way  
to  
do  
it



there's  
only  
one  
(right)  
way  
to  
do  
it



if that's so



why:





map() and list comprehensions?



map() and list comprehensions?  
string functions and string methods?



map() and list comprehensions?  
string functions and string methods?  
old and new style classes?



map() and list comprehensions?  
string functions and string methods?  
old and new style classes?  
&c &c



the usual



IMAGE  
PLACEHOLDER

ADD SOMETHING HERE!!!!

# hysterical raisins





found a better solution





plus



plus  
maintaining b/w compat



b/w compat GOOD



but



a cost



“python fits in your brain”



brains are limited



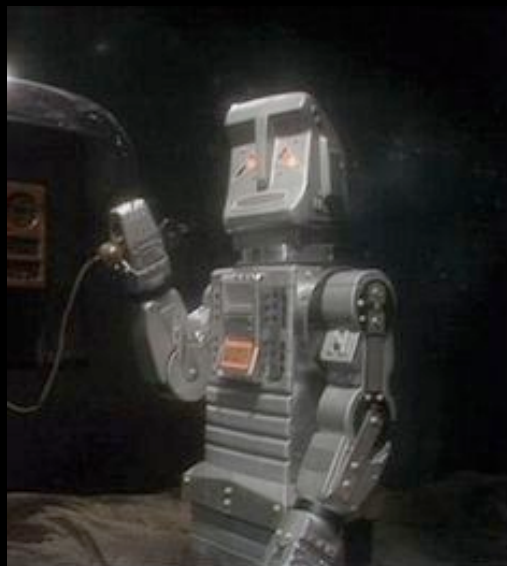
brains are limited<sup>1</sup>





brains are limited<sup>1</sup>

1:



bigger language is



harder to learn  
harder to use



harder to learn:







NCSS 2008



70 high school students



70 high school students  
limited programming





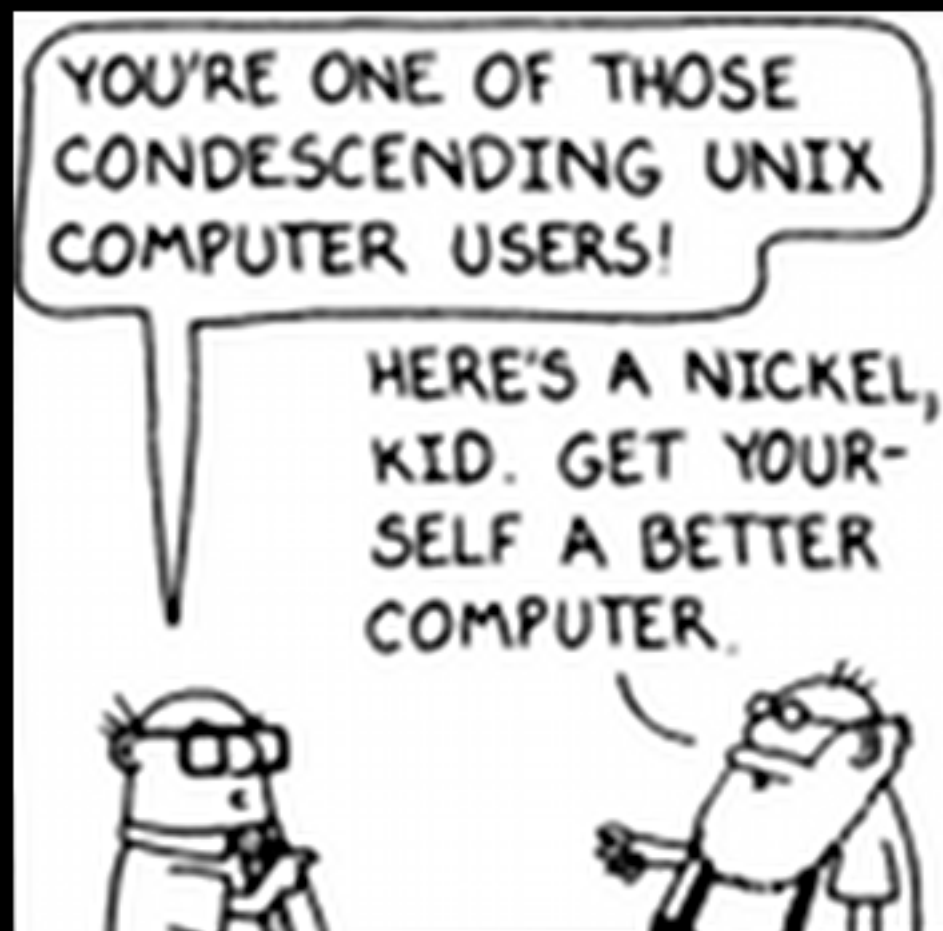


!= programming  
language



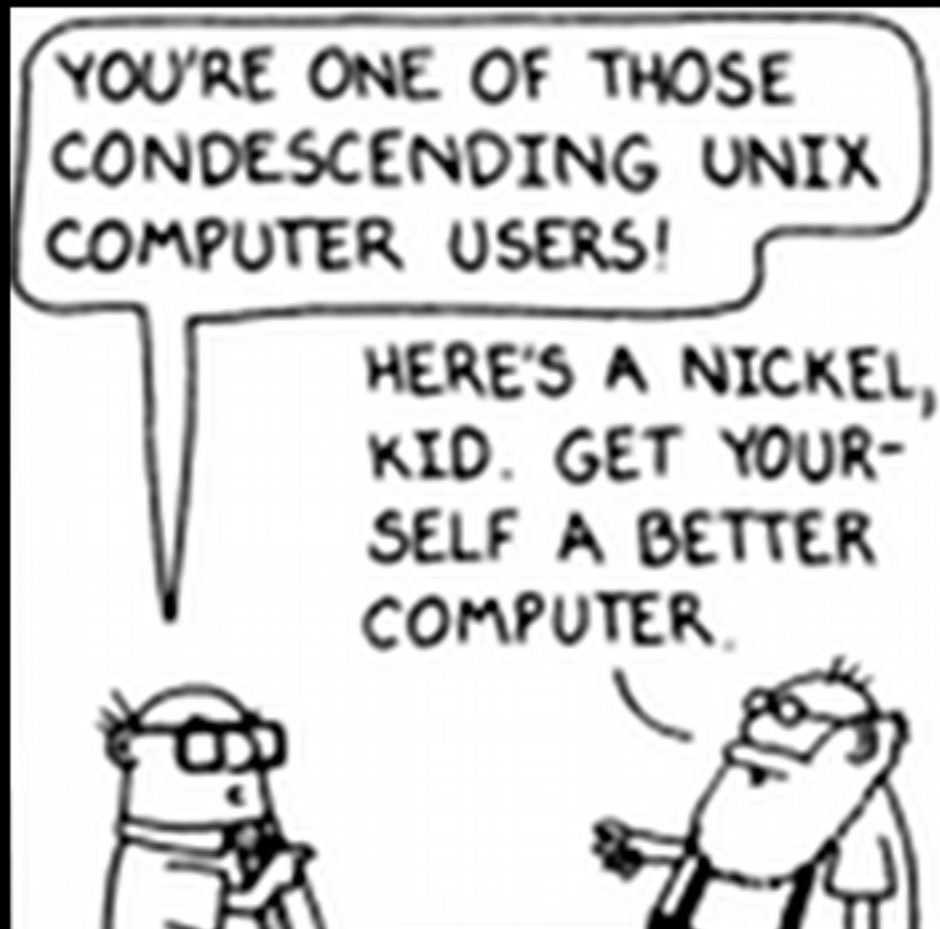


!= programming language





!= programming  
language



(yes, yes  
I am)

in 9 days:



in 9 days:  
learn python



in 9 days:  
learn python  
build a web site



in 9 days:  
learn python  
build a web site  
build a search engine



most manage it



python is great  
for teaching





1



# (aside)

**Green**

**Tea**

**Press**



## **How to Think Like a (Python) Programmer**

by Allen B. Downey

**Download [How to Think Like a \(Python\) Programmer](#) in PDF.**

UPDATE: December 7, 2007. I will be printing a short run of *How to Think Like a (Python) Programmer* for my class [Software Design](#) in the Spring. If you are a teacher using *How to Think Like a Computer Scientist*, and you are interested in helping me test the new edition, please contact downey at allendowney dot com and we will arrange to get hardcopies for your students.

This book is a substantially revised version of [How to Think Like a Computer Scientist: Learning with Python](#).

As of August 2007, this version of the book is a DRAFT! The chapters are mostly complete, but not carefully edited. Some of the debugging sections are not done, and not all chapters have exercises.

language complexity  
has a cost



builtins



# builtins by version





1.0.1 (1994): 40



1.0.1 (1994): 40

1.5.2 (1999): 54





1.0.1 (1994): 40  
1.5.2 (1999): 54  
2.1 (2001): 69

1.0.1 (1994): 40  
1.5.2 (1999): 54  
2.1 (2001): 69  
2.5 (2006): 82

worse yet



BABY SNOT  
SUCKING  
APPARATUS



NEW



apply()

apply()  
callable()

apply()  
callable()  
input() vs raw\_input()

```
>>> foo = input("Enter a number: ")
Enter a number: os.system('echo pwned')
pwned
>>> 
```



# T2

## International - Check In

Airline	Flight	Destination	Check-In	Dep	Status
JAL	JL5144	TOKYO	34-49	1025	CHECK IN OPEN
SINGAPORE AIRLINES	SQ238	SINGAPORE	10-18	1050	CHECK IN OPEN
QANTAS	QF359	HANOI	67-75	1125	CHECK IN OPEN
UNITED AIRLINES	UA9558	AUCKLAND			CHECK IN OPEN
American Airlines	AA7349	LOS ANGELES			CHECK IN OPEN
UNITED AIRLINES	UA840	LOS ANGELES			CHECK IN OPEN
KLM	KL4212	KUALA LUMPUR			
CATHAY PACIFIC	CX104	HONG KONG			
Lufthansa	LH9775	SINGAPORE	10-18	1700	
BRITISH AIRWAYS	BA7309	LONDON	34-49	1710	
Lufthansa	LH9737	BANGKOK	67-75	1730	
QANTAS	QF387	LONDON	78-86	1735	

Airport 20/20 Graphical Display Unit

Airport 20/20 Graphical Display Unit has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

Please tell Microsoft about this problem. We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

To see what data this error report contains, [click here](#).

10:09

PLEASE DO NOT LEAVE YOUR

NEC





T2

# International - Check In

Airline	Flight	Destination	Check in	Departure	Status
JAL	JL144	TOKYO	34-49	1020	CHECK IN OPEN
SINGAPORE AIRLINES	SQ38	SINGAPORE	10-18	1050	CHECK IN OPEN
QANTAS	QF387	HAWAII	67-75	1120	CHECK IN OPEN
UNITED AIRLINES	UA9558	AUCKLAND			CHECK IN OPEN
American Airlines	AA7349	LOS ANGELES			CHECK IN OPEN
UNITED AIRLINES	UA840	LOS ANGELES			CHECK IN OPEN
KLM	KL4212	KUALA LUMPUR			
CATHAY PACIFIC	CX104	HONG KONG			
Lufthansa	LH9775	SINGAPORE		1700	
BRITISH AIRWAYS	BA7309	LONDON	34-49	1700	
Lufthansa	LH9737	BANGKOK	67-75	1700	
QANTAS	QF387	LONDON	78-86	1735	

Report 20/20 Graphical Display Unit

Airport 20/20 Graphical Display Unit has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

Please tell Microsoft about this problem. We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

To see what data this error report contains, [click here](#).

Send Error Report    Don't Send

# GIGANT FAIL

10:09

PLEASE DO NOT LEAVE YOUR

NEC

makes teaching harder



not just learning...



regular users



regular users  
irregular users



un-named  
workplace



un-named  
workplace





lots of coders

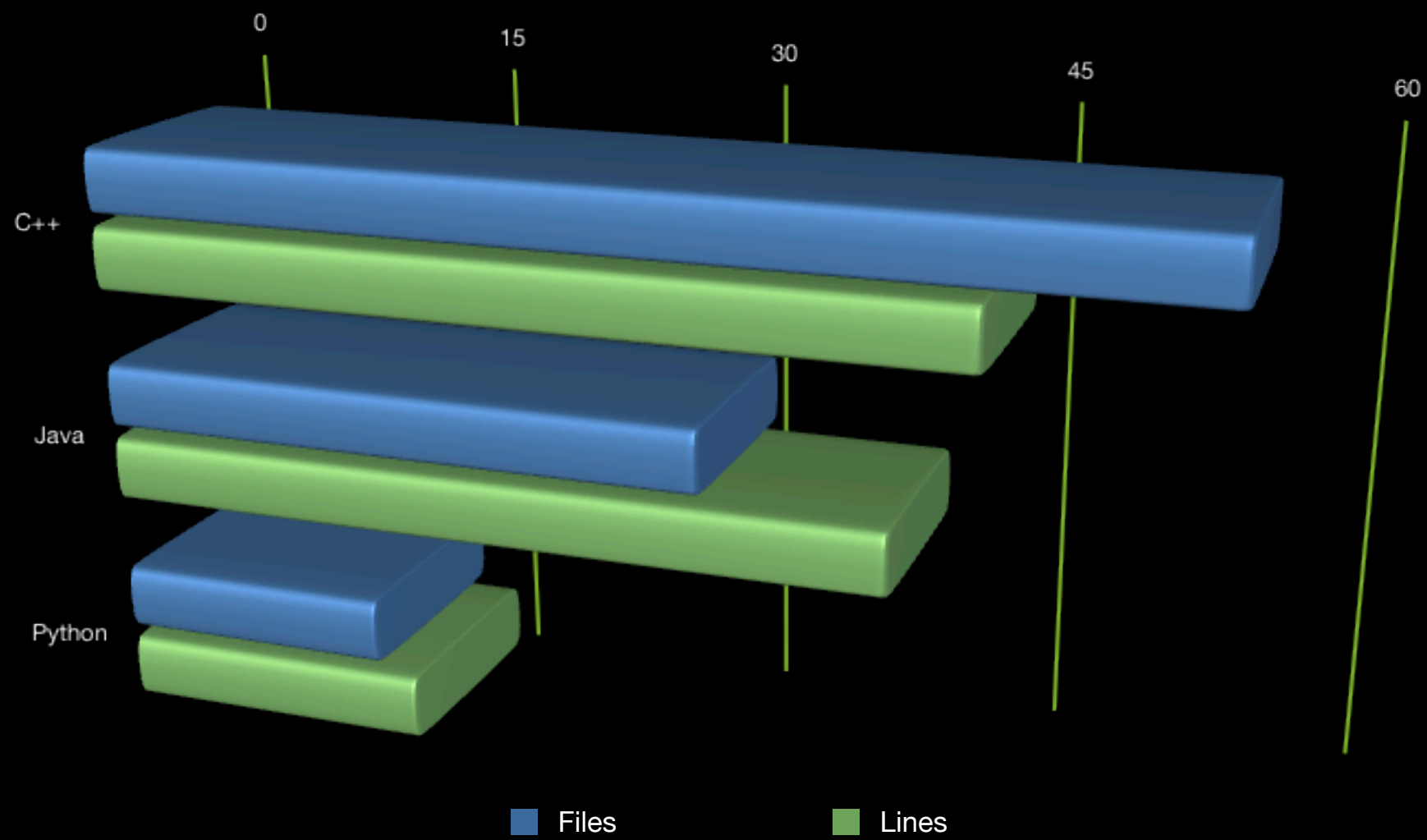


multiple languages



rare to be exclusively  
monolingual





need to context switch



lighter-weight language



lighter-weight language  
win



“executable pseudocode”





simpler is better



next...



# migration strategies



we care about this



(we have lots of code, too)



2to3.py



parses 2.x code  
converts to 3.x code



unit tests





suggested workflow



some stuff hard  
to determine



# Python 2.6 warnings



-Wpuzk



warns about 3.x problems



work in progress



```
from __future__ import  
    3k-feature
```



2.6 will have these





2.7 should have more



non-goals for 3.0



massive changes to the  
language



braces



# braces

```
Python 2.4.4 (#1, Oct 18 2006, 10:34:39)
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from __future__ import braces
      File "<stdin>", line 1
SyntaxError: not a chance
>>> □
```



multi-line lambdas



removing the GIL



rewriting the interpreter





rewrites bad



rhymes with



rhymes with  
“shmerl pix”  
or “trozilla”



# CPython



# CPython reference implementation



tries to avoid serious hacks



tries to avoid serious hacks  
hurts developers



tries to avoid serious hacks  
hurts developers  
hurts portability





other implementations



# IronPython



# Python on Mono



# Python on Mono (or .Net, if you must)



microsoft open source



microsoft open source  
no, really



microsoft open source  
no, really  
sane license even



works on Mono





works on Mono  
performance



jython



pyru



research project



EU funded



seriously gnarly



wrapping up



3.0 is our one chance





3.0 is our one chance



3.0 is our one chance



there won't be a 4.0



there won't be a 4.0



there won't be a 4.0



2.x is not going away



but 3.0 is the future



it's a 2 snake future





the goal



the goal

less of this



the goal



less of this

the goal



less of this

more of this

the goal



less of this

more of this





python<sup>TM</sup>