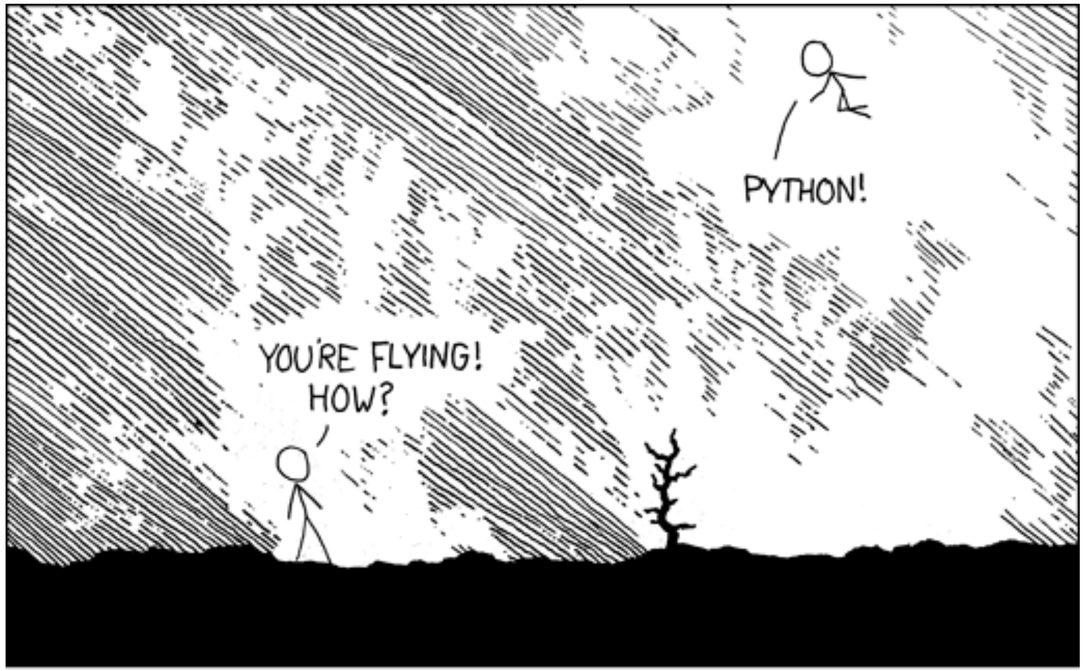
Python Scripting - Part 1

Fall 2018
PCfB Class 4
September 21, 2018



Outline

- Why Python?
- Data types
- Variables
- Methods

Why Python?

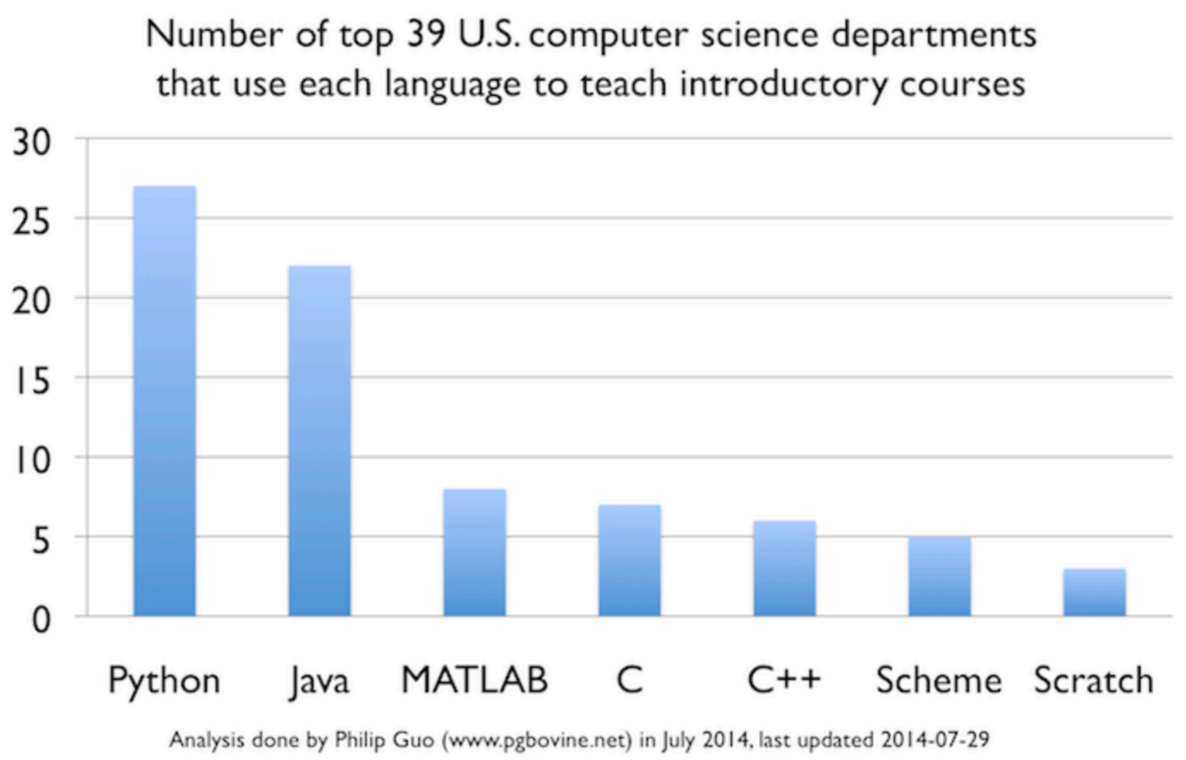
Enhanced readability

PYTHON

```
print('hello world')
```

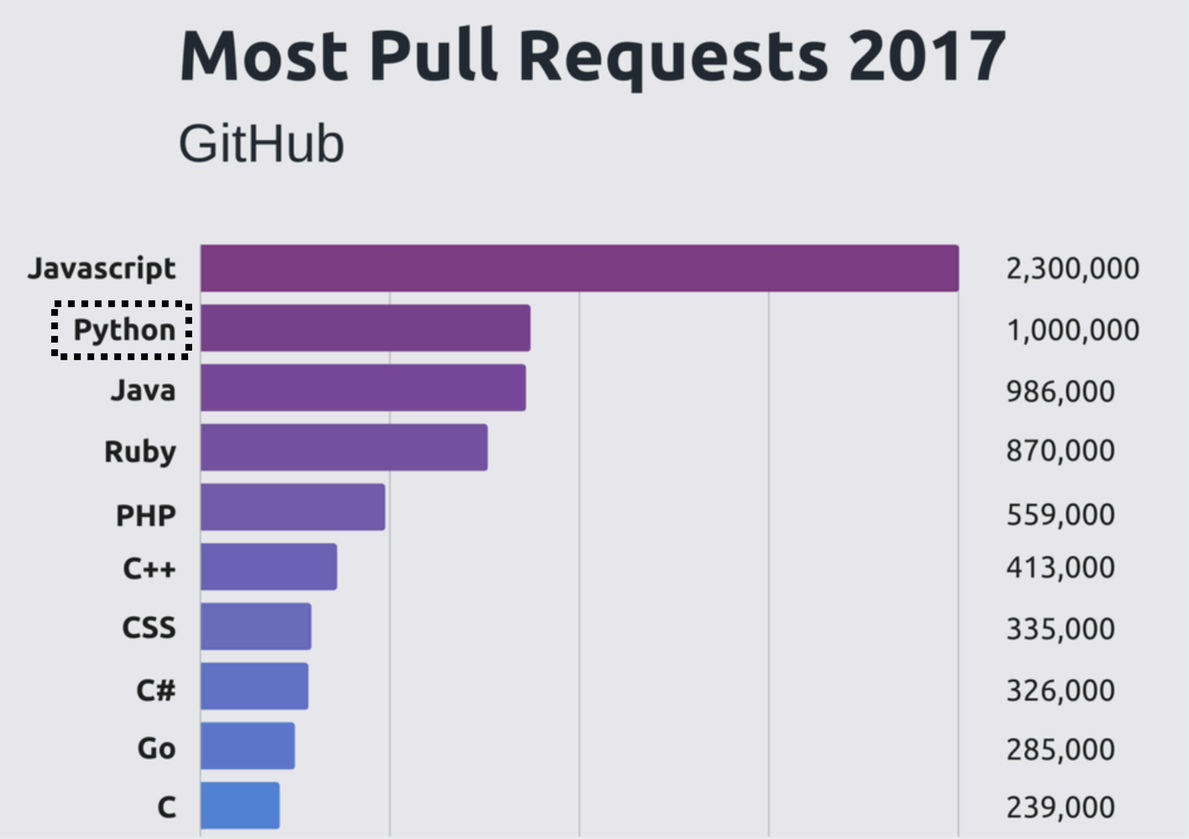
JAVA

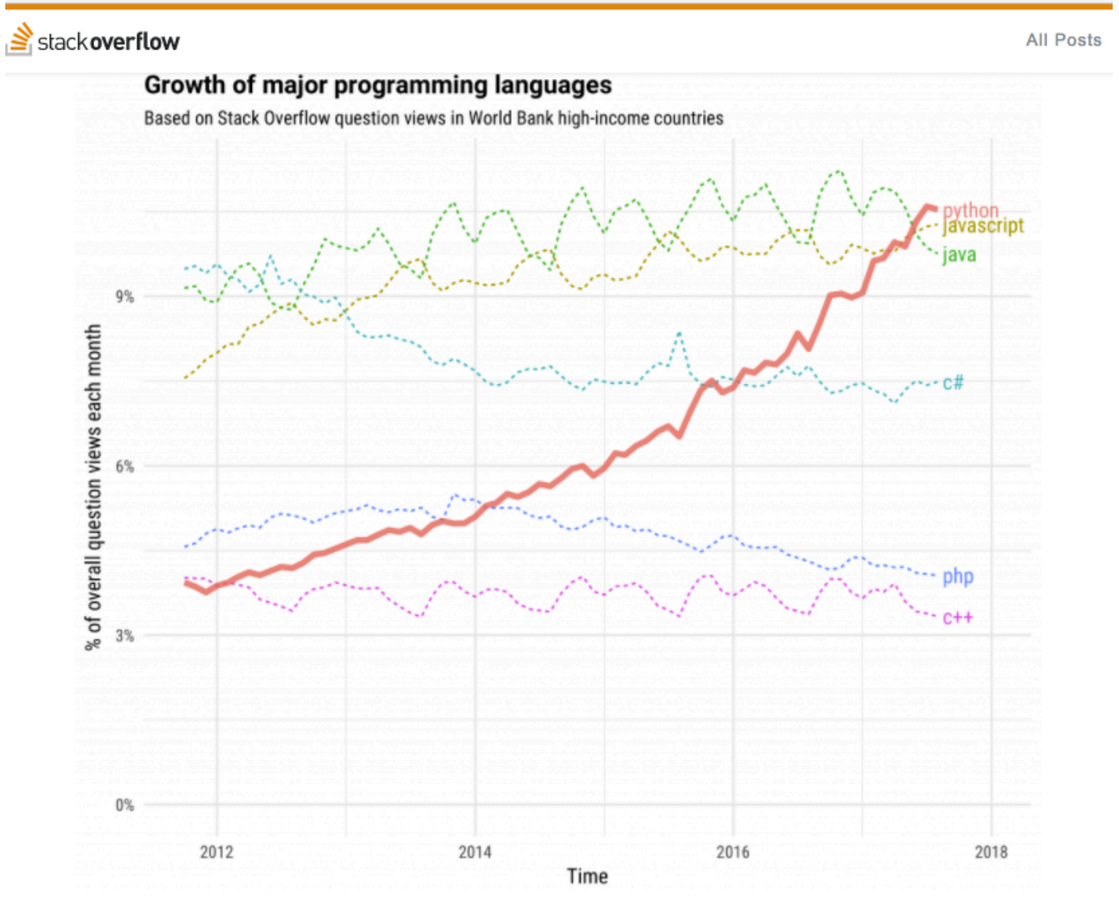
```
public class Main {
  public static void main(String[] args) {
     System.out.println("hello world");
   }
}
```

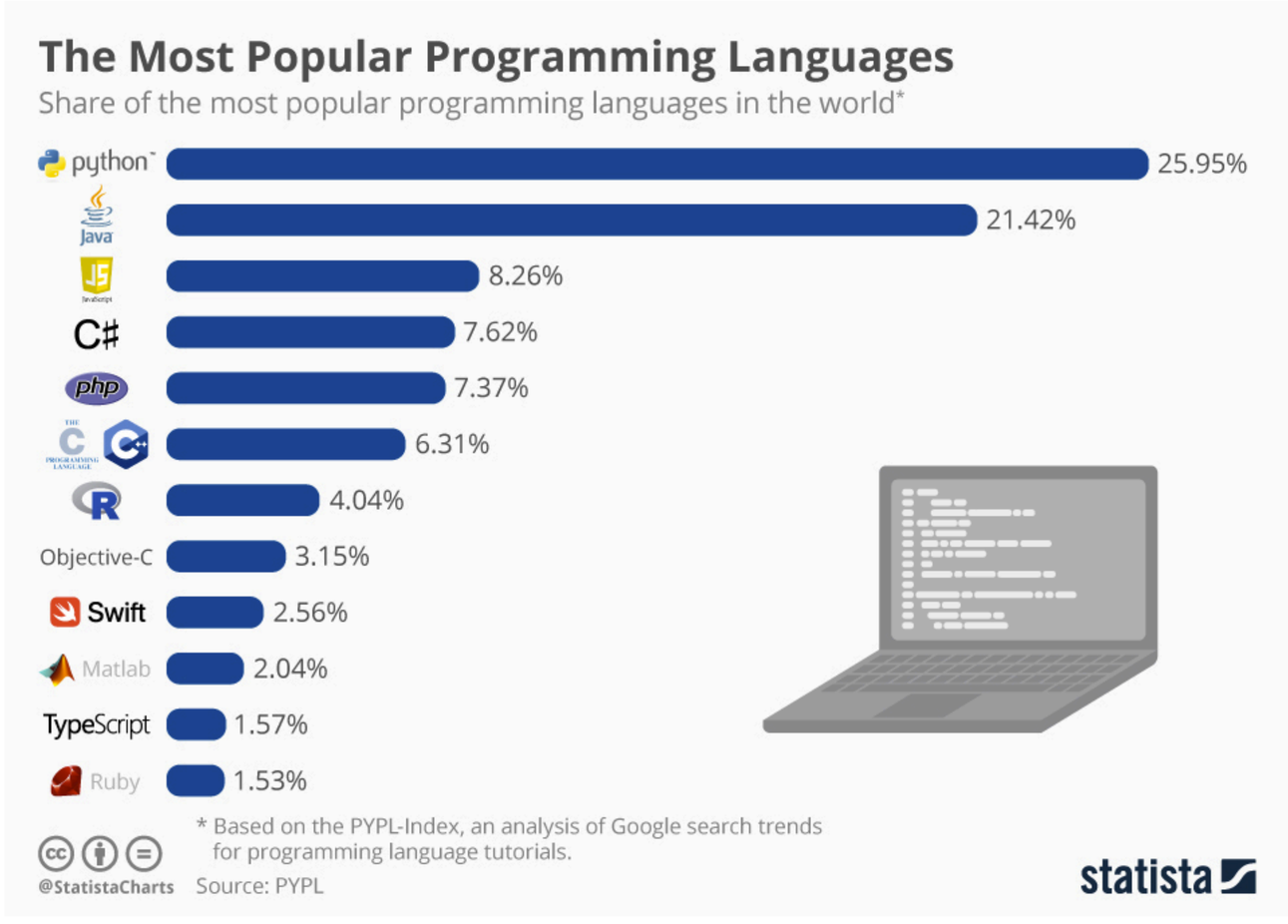


Still very powerful

Very popular




stackoverflow
All Posts
Growth of major programming languages
Based on Stack Overflow question views in World Bank high-income countries
python
javascript
java
c#
php
c++
9%
6%
3%
0%
% of overall question views each month
2012
2014
2016
2018
Time


The Most Popular Programming Languages
Share of the most popular programming languages in the world*
python 25.95%
Java 21.42%
8.26%
C# 7.62%
php 7.37%
6.31%
4.04%
Objective-C 3.15%
Swift 2.56%
Matlab 2.04%
TypeScript 1.57%
Ruby 1.53%
* Based on the PYPL-Index, an analysis of Google search trends for programming language tutorials.
@StatistaCharts
Source: PYPL
statista

vs.

PYTHON 2		PYTHON 3
Legacy	>	Future
It is still entrenched in the software at certain companies		It will take over Python 2 by 2020
Library	≠	Library
Many older libraries built for Python 2 are not forwards-compatible		Many of today's developers are creating libraries strictly for use with Python 3
0100 0001 ASCII	+	Unicode 0000 0000 0100 0001
Strings are stored as ASCII by default		Text strings are Unicode by default
5/2=2	≠	5/2=2.5
It rounds your calculation down to the nearest whole number		The expression 5 / 2 will return the expected result
print “hello”	≠	print (“hello”)
Python 2 print statement		The print statement has been replaced with a print () function

PYTHON 2.X  PYTHON 3.X

```
>>> print "Hello World!"
Hello World!
>>> print 3/2
1
>>> variable = 123456789
>>> print (type(variable))
<type 'int'>
```

```
>>> print ("Hello World!")
Hello World!
>>> print (3/2)
1.5
>>> variable = 123456789
>>> print (type(variable))
<class 'int'>
```

Ways to use Python

1. Stand-alone scripts

- Code saved in text file, executed on command line
- As described in PCfB book

2. Interactive mode via command line

- Enter commands 1-by-1 on command line
- Good for testing

3. Jupyter notebook

- Rich, web-based interface; results presented inline
- Good for teaching purposes and sharing code

Data types

Data types

String

Integer

Floating point

Boolean

Converting between types

String

Integer

Floating point

Boolean

Data containers

List

```
[1, '1', 'one', [1,2]]
```

Dictionary

```
{1:‘one’, 2:‘two’, 3:’three’}
```

Variables

Methods

Dot notation

dir()

```
['__add__', '__class__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__',
'__ge__', '__getattribute__', '__getitem__', '__gt__',
'__hash__', '__iadd__', '__imul__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__len__', '__lt__',
'__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__',
'__repr__', '__reversed__', '__rmul__', '__setattr__',
'__setitem__', '__sizeof__', '__str__', '__subclasshook__',
'append', 'clear', 'copy', 'count', 'extend', 'index',
'insert', 'pop', 'remove', 'reverse', 'sort']
```

#Comment, #comment, #comment

- Used to:
  - Guide others through your script
  - Indicate assumptions being made
  - Document changes made across versions
- You really can’t have too many comments!
- Most will probably be more useful to YOU than others