Pyacademy

Conference Guide

Date:

4 – April – 2014

Welcome

Thank you for your interest in our 3-4 hours Python conference. Our conferences are designed to be more practical and interactive rather than a boring long theoretical discussion. Therefore we come up with this idea to provide mentioned information before you come to our conference. This will give use this opportunity to skip these information in favor of having more practical and challenging sections in coming conference.

First section of this guide comes with our conference outlines, and it won't be just a simple outline. This section provides you all of the Python codes, samples, and resources we will use and teach you on our coming conference.

As mentioned above, in our conference you will learn to write Python code exactly on our conference. So please be sure to bring this guide (specially second section) to conference, because you need examples and code for our Pythonic day!

Second section of this guide explains what the Python is, and why it's worthy to learn? Then next section illustrates a salary comparison between Python and other top programming languages. After that you will be familiar with the creator of Python and some other effective people. Finally this section will be finished by discussion about the concept of dynamic programming languages as well as some other relevant information with Python. Most of information on this section is selected from valuable online resources without any changes.

Please do NOT hesitate to contact us for any question. And happy to see you soon ☺

Table of Contents

Section One :: Conference Outline

Start Time		Task
0:00	2 Min	Hello World
0:02	1 Min	Poppy Presentation
0:03	7 Min	Greeting
0:13	2 Min	Introduce Speaker
0:17	40 Min	Need for Speed? Learn Python
1:00	15 Min	BREAK
1:20	10 Min	Python for Bioinformatics
1:30	10 Min	Database
1:40	10 Min	Object Oriented Programming
1:50	10 Min	Make Graphical User Interface
2:00	15 Min	BREAK
2:20	50 Min	Create an awesome personal BLOG!
3:10	30 Min	Question & Answering

Part I: Python for Everyone

Install Python

We will show how to first install Python on your machine and then we will discuss regarding some useful utilities that must be added to your python for a better experience. If you are Linux, or Mac users you have to be happy because already Python is installed in your machine! Just open up terminal, and write Python! And if you are windows users, that won't be a big job! We will show you how to install and get Python ready on your machines.

After Installation you have to add some more sugar to your Python to make it tasty! Anyway sometimes it would be difficult to add these sugars ☺ But don't worry we will show you all things!

Need for Speed? Learn Python

Ok, let's ready to have a Pythonic time to learn this simple language and then start to write your first baby applications. Please check this slides and we will go through of them with together in our conference.

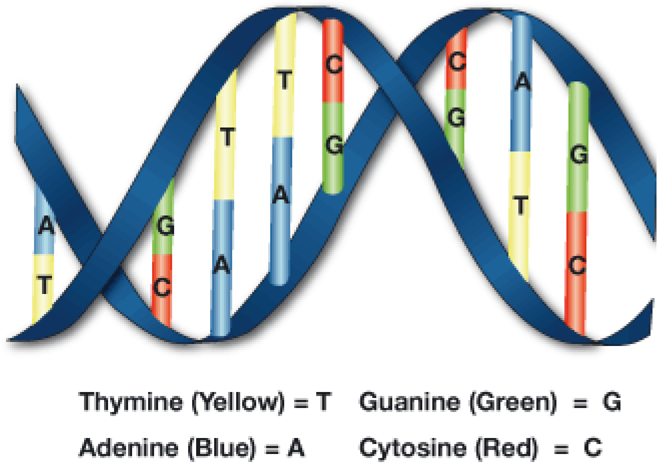
Part II: Python for Everywhere

Bioinformatics

One of the goals of molecular biology is to understand the processes that take place within the cells of living organisms. One such process is the creation of proteins, some of the most basic raw materials of all living things. Almost every process within a living creature makes use of, or is influenced by, these large, complex molecular structures. There are thousands of different proteins and we have barely begun to understand them in any detail. One thing we do know is that the creation of proteins is determined by the information encoded within the genetic material in each cell, called DNA.

More about DNA

DNA is a linear structure made up of a sequence of molecules called nucleotides or bases. Four nucleotides appear in DNA: adenine, cytosine, guanine, and thymine. These nucleotides are usually represented by their initials, A, C, G and T. DNA is actually composed of two strands of these nucleotides wound around each other in the famous double helix shape.



The sequence of a single strand of DNA can be represented as a sequence of alphabetical characters identifying each base in the sequence, such as ACCTTGGCACCT. Due to their chemical attractions, the nucleotides always appear in pairs, also called base pairs, such that adenine (A) always pairs up with thymine (T), and cytosine (C) always pairs up with guanine (G). Because of this base-pairing characteristic, we can easily determine the complementary, or opposite, strand of any single-stranded DNA sequence.

A simplified view of how DNA determines the creation of a protein goes something like this. A section of DNA called a gene contains the encoded information about the protein to create. Through the process of transcription, the two DNA strands along a gene separate, and the gene is copied. This single-stranded copy is called RNA, or, more precisely, messenger RNA. It is identical to the original gene sequence, except that the nucleotide uracil (U) appears in place of thymine (T).

GC-Content

In molecular biology and genetics, GC-content is the percentage of nitrogenous bases on a DNA molecule that are either guanine or cytosine. This can be translated to the ratio of total number of G and C in a DNA sequence over total number of G, C, T, and A.

$$\frac{G + C}{G + C + T + A}$$

Calculate GC-Content

First of all you need some textual DNA data in the form of A, G, C, and T or called FASTA. Then you would be able to dig in to that file and count relevant characters.

1. Google NCBI
2. Click on the first link (or www.**ncbi**.nlm.nih.gov)
3. Type BRCA1 in the search bar and click on "**Search**" button.
4. Click on "**Nucleotide**", and then select one of the interested DNA, and click on FASTA
5. Then from top-left download that file
6. Rename the file to "BRCA1.txt"

Now let's calculate GC content by using Python. You can find our DNA text file sample from **here**.

Codes:

1. Link to GC-Content
2. Generate Complement
3. Extract Codons

Database

Python comes with built-in library for SQLite. It means with out any extra installation you can just go directly and create your database. Also the same database interface in Python will be used to communicate with other databases such as MySql, MSSQL, …. As you might expect Python makes database programming a relatively painless affair. The Python database API (DBAPI) provides a 'database neutral' programming interface for databases such as MySQL, PostgreSQL, SQLite, MS-SQL and Oracle, among others, based on the structured query language SQL. In this section we provide two examples for working with database through the SQLite and also MySql.

1. Link to profile code

Object Oriented Programming

Remember our example from Bioinformatics section! Now in this section we are going to have a DNA class with some basic and important functions to show you how is Object oriented in python. Then we move on another example to create a class for “Profile” with some abilities such as save, load, and sending email. You will also get familiar with a technique called “Pickling”.

1. Link to profile code

Part III: Python for Everything

In less than an hour we will build a truly real application!

Section 2: What is Python?

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Python is a very simple language, and has a very straightforward syntax. It encourages programmers to program without boilerplate (prepared) code. The simplest directive in Python is the "print" directive - it simply prints out a line (and also includes a newline, unlike in C).

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, http://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

Source: www.python.org

Why Learn Python

- **It's free:** I've personally never heard a better reason to do anything. Python is a totally free language to download, use and play with, that's because a bunch of crazy volunteers who devote their time to improving the language (much like Wikipedia).
- **It's really easy to learn:** Not only have I been told it's a simple language to learn I have experienced it firsthand. Despite yet being an expert I have seen how fast my progress has been.
- **Google use it:** In fact Python is one of Google's preferred languages, they are always looking to hire experts in it and they have created many of their popular products with it.
- **It's versatile:** Ok, I promised no techie stuff so I'll keep it simple. Python can be used for small, large, online and offline projects. It's versatile, get it?
- **It's quick:** Some languages take an age to program not Python, remember it was created with programmer in mind and that means it is simple and quick to write code in Python.
- **Up to date:** Because of Python's volunteers and the fact that it's an open source language there are always people trying to improve it.
- **Fast (not just easy) to learn:** A Google Employee who turned me onto Python said I could become "reasonably proficient in it in less than two months", you wouldn't say that about learning French. If you have a brain suited to programming i.e. you like computers, aren't afraid of simple maths equations and are a problem solver then you should be able to learn your new skill quickly, which is a real bonus.
- **Great community:** Ever have a problem you can't figure out, or a link your can't find, just ask one of the thousands of Python community members who are more than willing to help out. You will find them on forums, Twitter, Facebook, Q&A sites, pretty much everywhere.
- **Python is very readable**. You won't waste a lot of time memorizing the arcane syntax that other programming languages will present you. Instead, you will be able to focus on learning programming concepts and paradigms
- **Library for everything**: Explain that library is just a file and then search for something like send EMAIL and show how to use

Source: http://www.theopenalgorithm.com/tag/python-for-beginners/

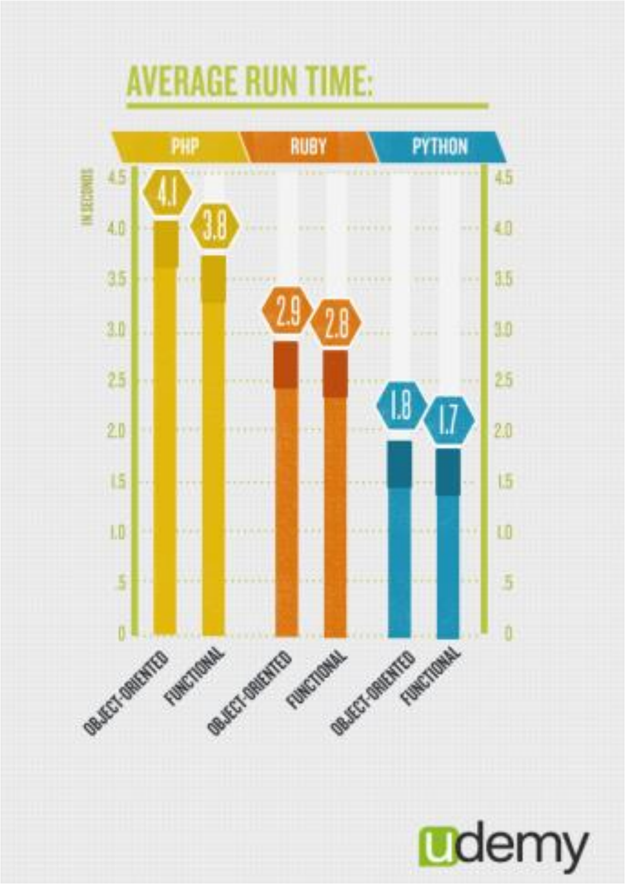
Why a Company May Choose Python?

Six Feet Up is a great software company started from 1999 and it 13 years they have choosed Python as the main developing language. The main reasons of this decision are explained by their CTO Calvin Hendryx-Parker.

1) Python is robust

There is a good reason why Bank of America has chosen Python to power many of their critical systems. (source) It's solid and powerful. Python has a relative small quantity of lines of code, which makes it less prone to issues, easier to debug, and more maintainable. The Securities Exchange Commission has sought to mandate Python as the language for a new "waterfall" program that would make Wall Street more transparent. Python can scale to solve complex problems, as illustrated by the fact that it powers most of YouTube and DropBox, not to mention Reddit, Quora, Disqus and FriendFeed. Even the mighty Google has made Python one of its official programming languages. It's also very fast.

Python is fast



2) Python is flexible

In 2007 YouTube migrated from PHP to Python for scalability purposes, citing that "Python enables flexibility". Python is used in a wide array of industries and for a long list of different usages, from websites and web applications to systems administration, voice over IP, and desktop apps. Python is also a staple of the Scientific community.

Because it wasn't originally created to answer a specific need, Python isn't driven by templates or specific APIs, and is therefore well-suited to rapid development of all kinds of applications. As a company focused on advanced web development, we really like this flexibility.

3) Python is easy to learn and use

"Python in particular emerges as a near ideal candidate for a first programming language", says John M. Zelle, in the Department of Mathematics, Computer Science, and Physics at Wartburg College in Iowa (source). We certainly agree with this as we find Python intuitive and fun. We don't have to look up references frequently, nor are we overwhelmed by the formalities of the language, like we would in Java or C++.

Python's simple and straight-forward syntax also encourages good programming habits, especially through its focus on white space indentation, which contributes to the development of neat looking code.

Finally, while PHP is notorious for the inconsistency in its naming methods, Python's naming convention is prevalent from module to module, so developers are less likely to make syntax errors. This means fewer bugs and faster development.

4) Python reduces time to market

Gartner estimates that 90% of enterprises are using open source software—including Perl, Python and Tcl—to build business-critical applications. That's because dynamic languages are an excellent solution for fast time-to-market for enterprise applications. Python makes it possible to get applications to market faster in part due to the fact that it has a huge standard library and is often referred to as coming with "batteries included". In addition, Python stays out of my way. Therefore I can be more productive than if I was using Java/XML: the same task will require less code using Python.

5) Python is free.

Since Python is an open source programming language, we immediately reduce up-front project costs by leveraging Python in our development projects.

Now, I'll agree that, more than a choice of language, what matters is the experience of the development team, their process, and how well they follows standards and best practices. We only work with experts in their fields so we can be proud of the code we deliver. Should you need assistance with your Python project, we'll be happy to help through a variety of professional services.

Source: http://www.sixfeetup.com/blog/why-we-choose-python

Python as First Language

Currently, there is little consensus about which programming language is most appropriate for introductory computer science classes. Most schools use a traditional system programming language such as C, C++, Java, or Ada in CS1 and CS2. However, scripting languages such as Tcl, Perl and Python are becoming increasingly popular software development tools.

There is a study has been done by John M. Zelle, at Department of Mathematics, Computer Science, and Physics to prove that Python must be the first selected language at all schools.

Strongly we recommend you to check this article and have a better prespective of why this language must be selected at the first. You can check this article from http://mcsp.wartburg.edu/zelle/python/python-first.html.

A Bit History of Python

Guido van Rossum says:

"During the 1989 Christmas holidays, I had a lot of time on my hand, so I decided to give it a try. During the next year, while still mostly working on it in my own time, Python was used in the Amoeba project with increasing success, and the feedback from colleagues made me add many early improvements.

In February 1991, after just over a year of development, I decided to post to USENET. The rest is in the Misc/HISTORY file."

Why is it called Python?

At the same time he began implementing Python, Guido van Rossum was also reading the published scripts fromMonty Python's Flying Circus (a BBC comedy series from the seventies, in the unlikely case you didn't know). It occurred to him that he needed a name that was short, unique, and slightly mysterious, so he decided to call the language Python.

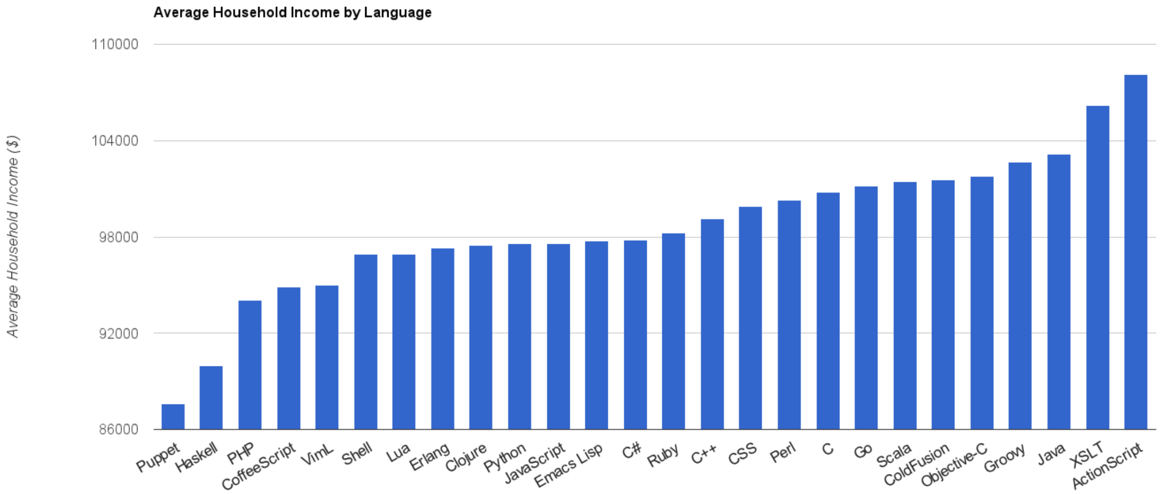
"The most important thing in the programming language is the name. A language will not succeed without a good name. I have recently invented a very good name and now I am looking for a suitable language." — **Donald Knuth**

Where is My Money

Here are the results for income, sorted from lowest average household income to highest:

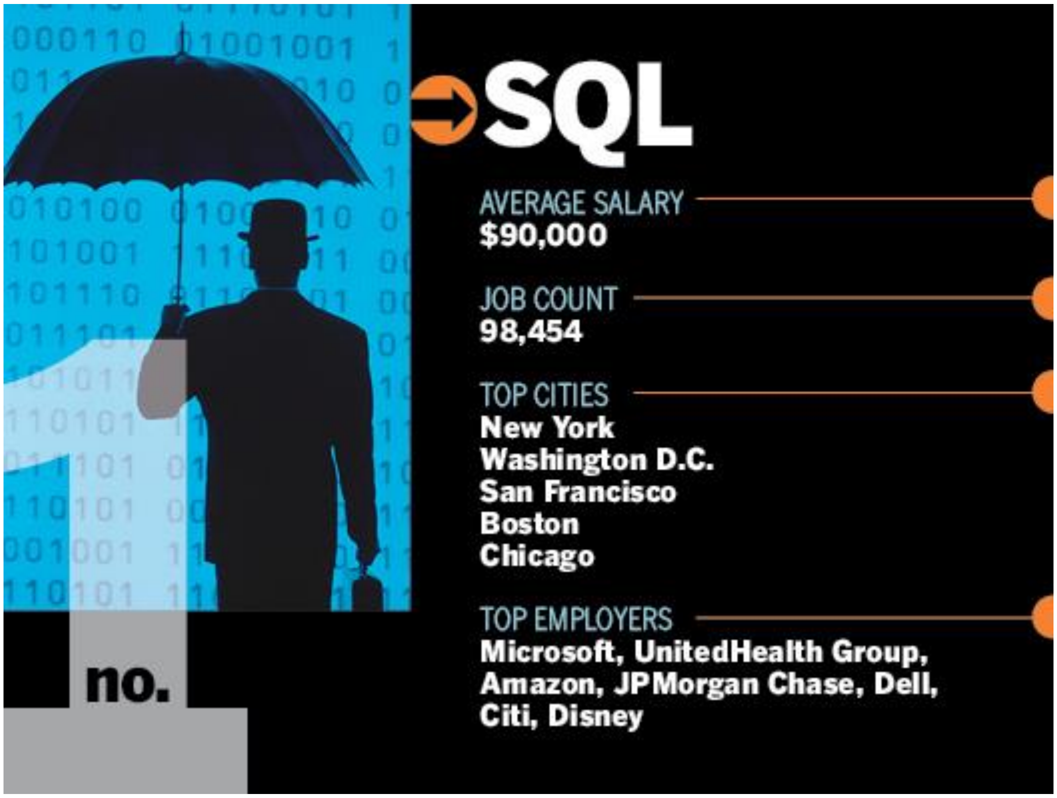
Language	Average Household Income ($)
Puppet (configure)	87,589.29
Haskell	89,973.82
PHP	94,031.19
CoffeeScript	94,890.80
Shell	96,930.54
Lua (most popular scripting language for game programming) World of Warcraft, Adobe Photosop lightroom	96,930.69
Erlang (Erricsson)	97,306.55
Python	97,578.87
JavaScript	97,598.75
Lisp	97,774.65
C#	97,823.31
Ruby	98,238.74
C++	99,147.93

CSS	99,881.40
Perl	100,295.45
C	100,766.51
Go	101,158.01
ColdFusion	101,536.70
Objective-C	101,801.60
Java	103,179.39
XSLT	106,199.19
ActionScript	108,119.47



- Haskell is a very academic language, and academia is not known for generous salaries
- PHP is a very accessible language, and it makes sense that casual / younger / lower paid programmers can easily contribute
- On the high end of the spectrum, Java and ActionScript are used heavily in enterprise software, and enterprise software is certainly known to pay well

What Computer World Magazine Says…



no. 1

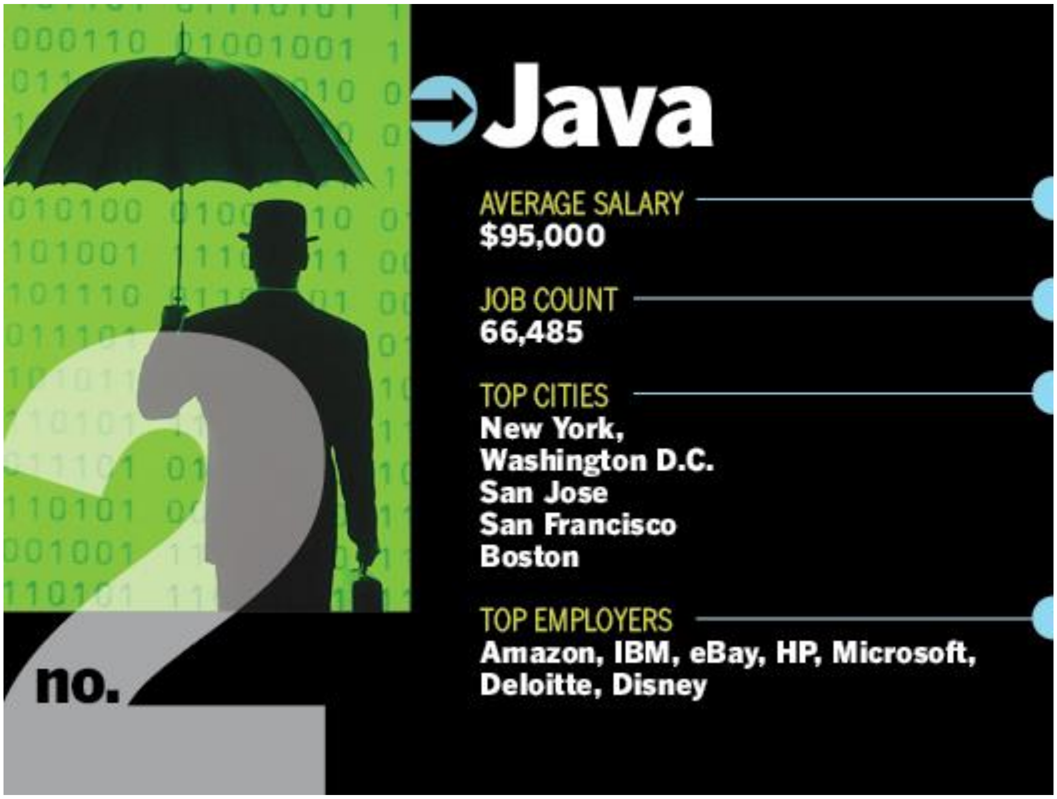
SQL

AVERAGE SALARY
$90,000

JOB COUNT
98,454

TOP CITIES
New York
Washington D.C.
San Francisco
Boston
Chicago

TOP EMPLOYERS
Microsoft, UnitedHealth Group, Amazon, JPMorgan Chase, Dell, Citi, Disney



no. 2

Java

AVERAGE SALARY
$95,000

JOB COUNT
66,485

TOP CITIES
New York,
Washington D.C.
San Jose
San Francisco
Boston

TOP EMPLOYERS
Amazon, IBM, eBay, HP, Microsoft, Deloitte, Disney

HTML

AVERAGE SALARY
$81,000

JOB COUNT
45,456

TOP CITIES
New York
Washington D.C.
San Francisco
Boston
Los Angeles

TOP EMPLOYERS
Amazon, Microsoft, eBay, IBM, AT&T, Disney, Fidelity

JavaScript

AVERAGE SALARY
$88,000

JOB COUNT
43,189

TOP CITIES
New York
San Francisco
Washington D.C.
Mountain View
Boston

TOP EMPLOYERS
Amazon, Microsoft, IBM, Dell, Yahoo!, Disney, Salesforce

C++

AVERAGE SALARY
$94,000

JOB COUNT
32,018

TOP CITIES
Santa Clara
Seattle
New York
San Francisco

TOP EMPLOYERS
Amazon, Microsoft, Intel, Qualcomm, HP, The MathWorks, Disney

C#

AVERAGE SALARY
$91,000

JOB COUNT
31,936

TOP CITIES
Seattle
New York
Boston
Chicago
Atlanta

TOP EMPLOYERS
Microsoft, Amazon, HP, Digi-Key Corporation, Dell, Intel, Allscripts

XML

AVERAGE SALARY
$92,000

JOB COUNT
31,431

TOP CITIES
Washington D.C.
New York
San Francisco
Boston
Atlanta

TOP EMPLOYERS
Amazon, Dell, Microsoft, Lorven Technologies, SAIC, JPMorgan Chase, AT&T

C

AVERAGE SALARY
$93,000

JOB COUNT
24,801

TOP CITIES
Santa Clara
Seattle
New York
San Francisco
San Diego

TOP EMPLOYERS
Amazon, Microsoft, Qualcomm, Intel, Digi-Key Corporation, Dell, HP

no. 9

Perl

AVERAGE SALARY
$93,000

JOB COUNT
20,579

TOP CITIES
San Jose
New York
Seattle
San Francisco
Boston

TOP EMPLOYERS
Amazon, Intel, Qualcomm, Dell, Yahoo! Inc., Raytheon, eBay

no. 10

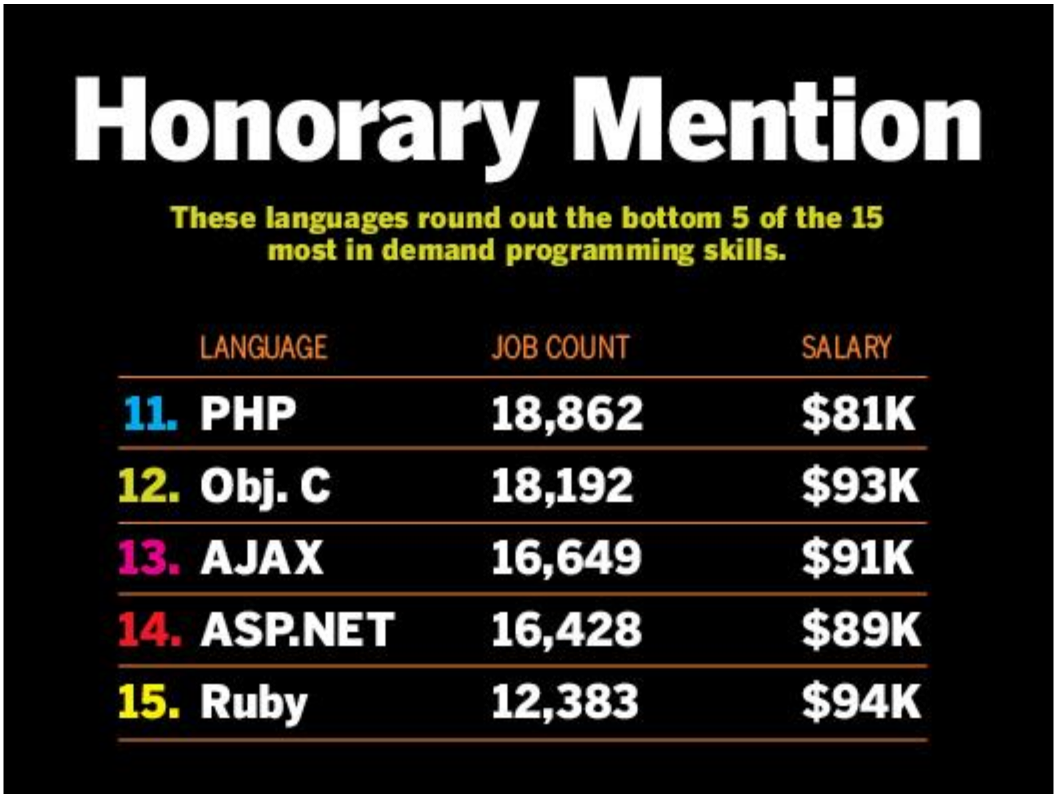
Python

AVERAGE SALARY
$83,000

JOB COUNT
19,627

TOP CITIES
Mountain View
San Francisco
New York
Boston
Seattle

TOP EMPLOYERS
Amazon, Intel, Dell, Google, eBay, Yahoo! Inc., EMC



Honorary Mention

These languages round out the bottom 5 of the 15 most in demand programming skills.

	LANGUAGE	JOB COUNT	SALARY
11.	PHP	18,862	$81K
12.	Obj. C	18,192	$93K
13.	AJAX	16,649	$91K
14.	ASP.NET	16,428	$89K
15.	Ruby	12,383	$94K

Good Books

I have in the course of learning to program bought two books, the best for beginners being Hello World! Computer Programming For Kids and Other Beginners. Don't be put off that the book is aimed at kids, that just means it is easy to use and contains simple language anyone could understand. But if your afraid to have a book with "Kids" in the title on your bookshelf then the other book I bought was Python 3 for Absolute Beginners, which I found less useful as it was more theory than the other book which contained a lot of exercises but would probably be quite handy for somebody with some sort of basic programming experience.

Source: http://www.theopenalgorithm.com/learning-python/10-non-techie-reasons-you-should-learn-python/

PyCons Video

We recommend you to follow up the main Python committee conferences that called PyCons. You can have access to their video from http://pyvideo.org/speaker.