gentoo linux™

Learning to Program With Python – Part 1

Computers, Languages, and Tools: Saying "Hello"

Based on the book:

Snake Wrangling for Kids, Learning to Program with Python

by Jason R. Briggs

(Version 0.7.7-python2.7, modified by SJL)

Presented by

Steve Arnold, Principal Scientist VCT Labs

Stephanie Lockwood-Childs, President VCT Labs

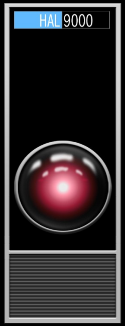
(we are also open source Gentoo Linux / Yocto developers)

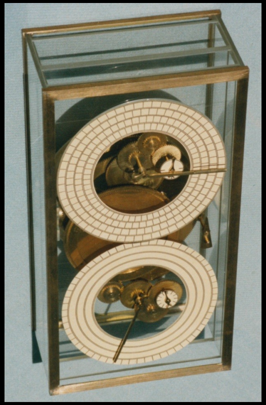
Computers and Languages

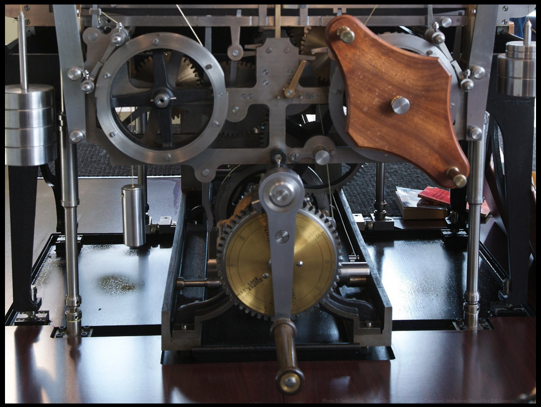
- What is a computer?
  - Abacus? Calculator? Watch? Phone? Tablet?
  - Game Console? Laptop? Desktop? HAL 9000?
- Languages
  - Humans and Computers
  - Binary language of moisture “vaporators”
  - Number systems and representations
  - Bits and bytes, text and binary
  - Machine and Programming Languages
- Computers and Intelligence
  - Can animals think?
  - Can computers think?

Moisture Vaporators

Which of these is a computer?







Computers and Languages cont.

- Files and Encodings
  - ASCII, Binary, Octal, Decimal, Hexadecimal
- Computer Language “Flavors”
  - Machine and Assembly Languages
    - Processors and Architecture
  - High Level Languages
    - Compilers and Interpreters
    - 1st “modern” language: FORTRAN (1954)
    - ALGOL, LISP (1958)
    - Then A, B, C, D, C++, C#, perl, Java, etc
  - “Human” Programming Languages
    - Ada (1st standard published 1983)
    - Python (1st implementation 1989)

Programming Language Tools

- The Language Files
  - Compiler/interpreter, libraries, other components
- Programmer's Editor
  - Text editor with special features for software development (not a word processor)
- Shell / Terminal Emulator
  - A place to type commands
  - Like a DOS prompt only better
  - Special shells, eg: python, pycrust
- Integrated Development Environment (IDE)
  - Like a fancy editor, with project interface, compilers
  - Can provide useful language-sensitive features
  - Can also require lots of system resources

Writing Your First Program in Python

- Login and launch Terminal Emulator (bash shell)
  - At the “$” prompt, type:
    - `$ python <enter>`
  - At the “>>>” prompt, type:
    - `>>> print ("Hello, world!") <enter>`
      - `Hello, world!`
- Now write it as a Python program
  - Open your editor and type:
    - `print ("Hello, world!")`
  - Save the file as “hello.py”
  - At the “$” prompt, type”
    - `$ python hello.py`
- Congratulations, you just wrote a Python program!

What Does /. Say?

What programming language will earn you the biggest salary over the long run? According to Quartz, which relied partially on data compiled by employment-analytics firm Burning Glass and a Brookings Institution economist, Ruby on Rails, Objective-C, and Python are all programming skills that will earn you more than $100,000 per year. But salary doesn't necessarily correlate with popularity. Earlier this year, for example, tech-industry analyst firm RedMonk produced its latest ranking of the most-used languages, and Java/JavaScript topped the list, followed by PHP, Python, C#, and C++/Ruby. Meanwhile, Python was the one programming language to appear on Dice's recent list of the fastest-growing tech skills, which is assembled from mentions in Dice job postings. Python is a staple language in college-level computer-science courses, and has repeatedly topped the lists of popular programming languages as compiled by TIOBE Software and others. Should someone learn a language just because it could come with a six-figure salary, or are there better reasons to learn a particular language and not others?