

# Up and Coming programming languages

What do we need today?

# Focusing our scrutiny

- As we begin our examination of python and ruby, need to consider:
  - what have we needed in a programming language in the past?
  - What do we need today?
  - What will we need in near future?

# Programming

- Theory of computation
  - Turing Equivalence
  - general purpose programming languages all have
    - loops, conditionals, variables
    - others can be built out of those.
- but theoretical equivalence is not everything
  - want to program a turing machine anyone?
  - languages with best syntax

# Old School programming needs

- In 50-60s
  - needed to speed numeric/scientific math calculations (matrix arithmetic etc)
    - fortran
  - Automate business functions
    - cobol
  - OS is in assembly
- through 70s
  - symbolic AI
    - lisp has large minority following for 30+ years

# More recently

- 1970s-early '80s
  - rise of minicomputer (server class today)
  - networking for universities and large companies
  - lots of hardware to be managed
  - C programming language makes its mark
    - os programming and lots of newer programs.
  - early '80s need simple language for computers with no power
    - BASIC

# Late 80s – 90s PL trends.

- personal computers
  - more and more memory, more and more complexity
  - bigger, more resource intensive programs
  - big, capable, object oriented programming languages
  - C++, Java become kings

# And today

- What are the emerging trends driving software design and programming?
  - web 2.0 (computing over the network)
  - multi-core/processing machines
  - cheap, pervasive 3-d graphics.
    - more than just games
  - quick turn around development
  - Ease of use/learning
  - What else?

# the criteria established

- As we examine these languages,
  - criteria established – how do the languages measure up?



# Good enough for day 1

- That's probably good for the first day
  - read through chapter 1 of the ruby book
  - and chapters one and two of the python book for your assignment