Where we hope to teach the fundamentals of programming and create connections between those elements and previous experience with Mathematica
BASIC PROGRAMMING
PROGRAMMING LANGUAGES

- Needed as human language is ambiguous
- Computer code composed of:
  - Syntax – structure of code
  - Semantics - meaning of code

<table>
<thead>
<tr>
<th>High level</th>
<th>Low level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python</td>
<td>Machine</td>
</tr>
<tr>
<td>C++</td>
<td>Assembly</td>
</tr>
<tr>
<td>Perl</td>
<td>Forth</td>
</tr>
<tr>
<td>Java</td>
<td>C</td>
</tr>
<tr>
<td>Ruby</td>
<td>G-code</td>
</tr>
</tbody>
</table>

Machine Instructions:
- load the number from memory location 2001 into the CPU
- load the number from memory location 2002 into the CPU
- add the two numbers in the CPU
- store the results in location 2003

Python Input:
c=a+b
Translation methods:

1. **Compiler**
   - Takes high-level code and translated into machine language
   - Occurs only once & may be run many times after
   - Tend to run faster

2. **Interpreter**
   - Simulates a computer that understands high-level language
   - Used every time program runs
   - Lend to more flexible programming

**Main Differences**
“Are you suggesting coconuts migrate?”
“Not at all. They could be carried”
“What? A swallow carrying a coconut?”
“It could grip it by the husk!”
“It's not a question of where he grips it! It's a simple question of weight ratios! A five ounce bird could not carry a one pound coconut.”
Creator: Guido Van Rossum
Where: Netherlands
When: late 1980s

Download Options:
- Python 3.3.2
- Python 2.7.5
- IronPython—running on .Net
- Jython—running on Java virtual machine
- PyPy—fast Python implementation with a JIT compiler
- Stackless Python—branch of Cpython
- Python(x,y)—scientific-oriented Python
- ...and so on
Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.

Imported NumPy 1.6.2, SciPy 0.11.0, Matplotlib 1.1.1 + guidata 1.5.1, guiqwt 2.2.1
Type "scientific" for more details.

>>> #>>> is a Python prompt indicating that it is ready for a command.
>>> #A complete command is called a sentence.
>>> print "Hello, World"
Hello, World
>>> def hello():
    print “Hello”
>>> hello():
    Hello
Definite
- Executed definite no. of times
- For Loop:
  ```python
  for <var> in <sequence>:
      <body>
  ```

Indefinite/Conditional
- Iterates until certain conditions are met
- While loop:
  ```python
  while <condition>:
      <body>
  ```

Danger of infinite loops
- Break out by pressing Ctrl+c
- Ctrl+Alt+Delete if non-responsive
- If all else fails, restart computer manually
DEcision structures

- **If statement:**
  ```python
  if <condition>:
      <body>
  ```

- **If/else statement:**
  ```python
  if <condition>:
      <statement>
  else :
      <statement>
  ```

- **If/elif/else statement**
  ```python
  if <condition>:
      <statement>
  elif <condition>:
      <statement>
  else :
      <statement>
  ```

<table>
<thead>
<tr>
<th>Python</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>==</td>
<td>Equal to</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>!=</td>
<td>Not equal to</td>
</tr>
</tbody>
</table>
try:
    <body>
Except <ErrorType>:
    <handler>

def quadSolver(a,b,c):
    try:
        discrim=math.sqrt(b*b-4*a*c)
        root1=(-b+discRoot)/(2*a)
        root2=(-b-discRoot)/(2*a)
        print "\nThe solutions are:" root1,"&", root2
    except ValueError:
        print "\nNo real roots."

Python Resources:

- Code Academy: Python
- Pythonanywhere
- *Python Programming: An Introduction to Computer Science*
  John M. Zelle

Other Coding Resources:

- Code Academy