PROGRAMMING WITH PYTHON

Writing simple programs in a scripting language

Cyrus Bazeghi
A PROGRAM

Set of instructions written in a programming language that tells the computer what to do
Python

- Python is a remarkably powerful dynamic programming language that is used in a wide variety of application domains. Python is often compared to Tcl, Perl, Ruby, Scheme or Java.*

* From www.python.org website
**Python Continued**

- Very clear, readable syntax
- Strong introspection capabilities
- Intuitive object orientation
- Natural expression of procedural code
- Full modularity, supporting hierarchical packages
- Exception-based error handling
- Very high level dynamic data types
- Extensive standard libraries and third party modules for virtually every task
- Extensions and modules easily written in C, C++ (or Java for Jython, or .NET languages for IronPython)
- Embeddable within applications as a scripting interface

*From [www.python.org](http://www.python.org) website*
## Python Syntax

- **Reserved words**

<table>
<thead>
<tr>
<th>and</th>
<th>exec</th>
<th>not</th>
</tr>
</thead>
<tbody>
<tr>
<td>assert</td>
<td>finally</td>
<td>or</td>
</tr>
<tr>
<td>break</td>
<td>for</td>
<td>pass</td>
</tr>
<tr>
<td>class</td>
<td>from</td>
<td>print</td>
</tr>
<tr>
<td>continue</td>
<td>global</td>
<td>raise</td>
</tr>
<tr>
<td>def</td>
<td>if</td>
<td>return</td>
</tr>
<tr>
<td>del</td>
<td>import</td>
<td>try</td>
</tr>
<tr>
<td>elif</td>
<td>in</td>
<td>while</td>
</tr>
<tr>
<td>else</td>
<td>is</td>
<td>with</td>
</tr>
<tr>
<td>except</td>
<td>lambda</td>
<td>yield</td>
</tr>
</tbody>
</table>
Lines and indentation

One of the first caveats programmers encounter when learning Python is the fact that there are no braces to indicate blocks of code for class and function definitions or flow control. Blocks of code are denoted by line indentation, which is rigidly enforced.

The number of spaces in the indentation is variable, but all statements within the block must be indented the same amount. Both blocks in this example are fine

```python
if True:
    print "True"
else:
    print "False"
```
Python Shell
**Python IDE**

![Python IDE screenshot](image)

---

Personal firewall software may warn about the connection IDLE makes to its subprocess using this computer's internal loopback interface. This connection is not visible on any external interface and no data is sent to or received from the Internet.

---

**IDLE 2.6.4**

```python
>>> |
```
GETTING INPUT FROM THE USER

- Two main commands “raw_input” and “input” for getting user input via the keyboard

```python
>>> a = raw_input("Please enter your name: ")
Please enter your name: Cyrus Bazeghi
>>> print a
Cyrus Bazeghi
>>> print "Hello " + a
Hello Cyrus Bazeghi
```
CASTING INPUT TO A NUMBER

- When you get input with “raw_input” it is treated as text, if we want to use it for math we need to convert it to a number.

```python
>>> a = int(raw_input("Enter an int: "))
Enter an int: 12
>>> print a
12
>>> b = float(raw_input("Enter a real: "))
Enter a real: 12.3
>>> print b
12.3
```
```python
>>> x = int(raw_input("Enter an integer: "))
Please enter an integer: 42
>>> if x < 0:
...    x = 0
...    print 'Negative changed to zero'
... elif x == 0:
...    print 'Zero'
... elif x == 1:
...    print 'Single'
... else:
...    print 'More'
```

There can be zero or more “elif” parts, and the “else” part is optional.
FOR STATEMENTS

>>> # Measure some strings:
... a = ['cat', 'window', 'defenestrate']
>>> for x in a:
... print x, len(x)
...
cat 3
window 6
defenestrate 12
>>> # Fibonacci series:
... # the sum of two elements defines the next
>>> a, b = 0, 1
>>> while b < 10:
...     print b
...     a, b = b, a+b
...
1
1
2
3
5
8
THE PROGRAMMING PROCESS

- Defining the problem
- Planning the solution
- Coding the program
- Testing the program
- Documenting the program
DEFINING THE PROBLEM

- What is the input
- What output do you expect
- How do you get from the input to the output
Flow Control Elements
Accept series of numbers and display the average