


Practical Computer Science with Python




James M. Allen
Hathaway Brown School
jamesallen@hb.edu

Preview

- Background
- Why Python?
- Getting Our Feet Wet
- Comparing Python to Java
- Resources (Including a free textbook!)


Background

- Hathaway Brown School
 - PreK-12 All Girls School in Shaker Heights
 - Students excel in math and science
 - CS classes have had low enrollment




Background

- Intro CS Course
 - Drawing upon CSTA's Model Curriculum, Level 2



Ohio CSTA Chapter
<https://sites.google.com/site/compscl Ohio/>

Pick a Programming Language!



<http://blog.lndigo.com.br/>

Why Python?

- Python is simple.

```

Java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
    
```

```

Python
print("Hello, world!")
    
```

Why Python?

- Python is powerful.
- Standard Python libraries include:
 - Audio manipulation
 - BSD socket library
 - HTTP/SSL/FTP/IMAP helper modules
 - Cryptography modules
 - Turtle graphics



Why Python?

- Python is widely used.



<http://wiki.python.org/moin/OrganizationsUsingPython>

Why Python?

- Python is free.



Why Python?

- It's simple.
- It's powerful.
- It's widely-used.
- It's free.
- It's fun!



DEMO: PYTHON THE CALCULATOR

The Python Shell

- The Python shell is a live "command-line" way to run Python code.
- It can be used like a calculator...
- Or you can even write quick, one-off programs in it...

Python Syntax

- A line is a line of code – there is no ending punctuation.
- Indentation matters!
 - Indents are used in lieu of curly braces to indicate blocks of code.
- Python is one of the most readable languages because good style is enforced.

Variables

- To create a variable in Python, you simply put a variable name to the left of an = sign and put something into it.
- Variables are actually strongly-typed, but their types can change at any point in a program.

Strings

- Strings in Python can look like the following:

```
'In apostrophes,'
" in quotation marks, "
"""or in triple quotes,
spread out over more than one line."""
```

Strings

- Strings can be manipulated using arithmetic operators or string methods.
- The % operator allows string-formatting similar to sprintf()

Input/Output

- Standard output: print()
- Standard input: input()

Comments

- Comments in Python are anything on a line following a # symbol.

DEMO: MAD LIBS

Madlibs.py

```
story = '''The %s man entered the %s building to visit a %s man.
"Sit down, Mr. %s. Can I interest you in any %s %s?''''
adj1 = input("Adjective: ")
adj2 = input("Adjective: ")
adj3 = input("Adjective: ")
lastname = input("Last name: ")
adj4 = input("Adjective: ")
noun1 = input("Plural noun: ")
print(story % (adj1, adj2, adj3, lastname, adj4, noun1))
```

Functions

- Functions in Python start with the `def` keyword, have parentheses containing parameters, and a colon at the end of the line followed by an indented code block.

```
def say_hello(person):
    print("Hello, %s, nice to meet you!" % (person))
```

DEMO: AREA OF A TRIANGLE

Modules

- Modules are brought into Python using the `import` keyword.
- There are many modules included with Python, and there are many more freely available.
 - Game Development: Pygame
 - GUI: PyGtk, PyQt, TkInter
 - Image Manipulation: PIL
 - "Compilation": Py2exe, Py2app
 - Web Development: Django, Pylons

DEMO: PLAYING WITH TURTLES

Python vs. Java

<p>Java</p> <ul style="list-style-type: none"> • Compiled language • Cross-platform • Statically typed • Strongly typed • High-level data structures must be imported • Data encapsulation (private) • Some variables are objects, some are primitives 	<p>Python</p> <ul style="list-style-type: none"> • Interpreted language • Cross-platform • Dynamically typed • Strongly typed • High-level data structures are built-in • No data encapsulation • Everything is an object
--	---

<http://pythonconquerstheuniverse.wordpress.com>

Python vs. Java

- Hello World

<p style="text-align: center; font-size: small;">Java</p> <pre>public class HelloWorld { public static void main(String[] args) { System.out.println("Hello, World!"); } }</pre>	<p style="text-align: center; font-size: small;">Python</p> <pre>print("Hello, world!")</pre>
--	---

Python vs. Java

- Number → string conversion and comparison

<p style="text-align: center; font-size: small;">Java</p> <pre>int n = 0; String s = String.valueOf(n); if (s.equals("0")) { System.out.println("Equal!"); }</pre>	<p style="text-align: center; font-size: small;">Python</p> <pre>n = 0 s = str(n) if (s == "0"): print("Equal!")</pre>
--	--

Python vs. Java

- Print integers from 1-100

<p style="text-align: center; font-size: small;">Java</p> <pre>for (int i = 1; i <= 100; i++) { System.out.println(i); }</pre>	<p style="text-align: center; font-size: small;">Python</p> <pre>for i in range(101): print(i)</pre>
---	--

Python vs. Java

- Read numbers from a file and compute the average.

<p style="text-align: center; font-size: small;">Java</p> <pre>try { Scanner sc = new Scanner(new File("numbers.txt")); double total = 0; int items = 0; while (sc.hasNextDouble()) { total += sc.nextDouble(); items++; } double average = total / items; } catch (IOException e) { System.out.println("File Error"); }</pre>	<p style="text-align: center; font-size: small;">Python</p> <pre>myFile = open("numbers.txt") data = myFile.readlines() myFile.close() total = 0 for d in data: total = total + d average = total / len(data)</pre>
--	---

Python vs. Java

- Add numbers to a list, sort them, and output using a custom separator.

Java	Python
<pre>ArrayList<Integer> t = new ArrayList<Integer>(Arrays.asList(7, 3, 15)); Collections.sort(t); Iterator i = t.iterator(); while (i.hasNext()) { System.out.print(i.next()); if (i.hasNext()) { System.out.print(" and "); } }</pre>	<pre>t = [7, 3, 15] t.sort() print(" and ".join(map(str, t)))</pre>

Python vs. Java

- Object for latitude/longitude coordinates.

Java	Python
<pre>public class Coord { private float latitude; private float longitude; public Coord() { } public Coord(float lat, float lon) { this.latitude = lat; this.longitude = lon; } public float getLatitude() { return latitude; } public float getLongitude() { return longitude; } public void setLatitude(float lat) { this.latitude = lat; } public void setLongitude(float lon) { this.longitude = lon; } }</pre>	<pre>class Coord(object): def __init__(self, lat = 0, lon = 0): self.latitude = lat self.longitude = lon</pre>

Resources

- Built-in documentation (F1)
- *Think Python* by Allen B. Downey
<http://www.greenteapress.com/thinkpython/thinkpython.html>
- Google



Questions?

- E-mail me: jamesallen@hb.edu