Accelerating Information Technology Innovation

http://aiti.mit.edu

Nigeria Summer 2012
Lecture 1 – Introduction to Python
Agenda

• What is Python?
• Why Python, in general?
• Why Python, for us?
• The Development Cycle
• Basic Syntax
• Interactive Experimentation (Lab)!
What is Python?
Python is...

- **interpreted.** Languages like C/C++ require **compilers** to translate high-level code to machine code...

```
High-Level Code
a = b + c;

Compiler

Machine Code
...
ld $r1, a
ld $r2, b
add $r3, $r1, $r2
st a, $r3
...
```
Python is…

• …which means that a program has to be compiled separately for each type of machine:
Python is...

- In contrast, Python is compiled to an intermediate format called bytecode, which is understood by a virtual machine.

```
Python Source (.py)  Python Bytecode (.pyc)
```

- This model is similar to Java’s and is designed to allow you to ‘Write Once, Run Anywhere’
Python is…

• This is accomplished through the use of Python virtual machines, or *interpreters*, which are built on each type of machine.
• The interpreter simulates the VM bytecode on the actual hardware, translating the VM’s ‘native’ calls to machine code.
• This presents a standard interface to the language, allowing portability
Python is...

Python Program → compiler → Python bytecode → Interpreter

Win
Mac
Unix
Python is…

• Compilation-transparent in the reference implementation (CPython) – it happens automatically!

• Interestingly, implementations exist for other VMs on the same hardware:
  – Jython – compiles to Java VM bytecode
  – Iron Python – compiles to .NET bytecode
Python is...

- Dynamically typed; variable types are determined at runtime depending on what you assign to them:

```python
# int
a = 1

# string
a = "a"

# list
a = [1, 2, 3]

# dictionary
a = {1: 2, 3: 4}
```
Why Python?
Python because…

- Python’s interpreted nature makes it portable and architecture-agnostic; if a machine runs Python, it’ll run your code.

- Python, like Java, includes many convenient built-in functions and datastructures which are already optimized for its virtual machine.
Python because…

- Python’s syntax is designed to be readable and fast to write. In addition to dynamic typing, whitespace is used as a block delimiter, and semicolons are not usually necessary:

```python
if (x)
{
    if (y)
        a();
    b();
}
```

```python
if x:
    if y:
        a()
    b()
```
Python because...

- Useful data structures (dictionaries, tuples, lists, etc.) are built-in and do not need to be separately imported
- Lack of a separate compile step speeds rapid prototyping and debugging
- Dynamic typing speeds up development – no need to explicitly specify method argument types beforehand
Why Python, For Us?
Python for us, because…

• We want each of you to reach millions of users, and don’t want to waste time building the pipes and plumbing

• Python is supported by a number of good frameworks, led by
  – Google AppEngine, and its progenitor -
  – Django
The Development Cycle
The (Ideal) Development Cycle

- Clearly specify the problem:
  - Inputs, input manipulation, outputs
- Design the solution:
  - E.g what algorithms, data structures
- Implementation:
  - Coding!
- Test, test, test
  - Strongly suggest unit testing with PyUnit
The (Real) Development Cycle

• As above, but *faster*.
  – Python, as a dynamically typed, dynamic language is perfect for *rapid* prototyping

• Be prepared to throw away one (or more!) prototypes
  – Often you learn crucial things about the problem as you code which cannot be fixed without starting from scratch.
Strong Recommendations

• Use self-documenting variable names
  – e.g. “name” instead of “n”

• Use full length camelcase for class names
  – e.g. “CustomPresenter” not “custpres”
  – More style tips at http://www.python.org/dev/peps/pep-0008/

• Comment everything that’s not absolutely obvious
  – Can you read your own code in 10 years?
Basic Syntax
Syntax

• As mentioned before, blocks are delimited with whitespace: specifically, four spaces (and no tabs)

```python
if x:
    if y:
        a()
    b()

accum = 0
for i in range(0, 5):
    accum += i
```
Syntax

- Semicolons are only used to separate multiple statements on the same line, which is discouraged:

```java
if (x)
{
    a();
    b();
}
```

```java
if x:
    a(); b()
```
Syntax

- Single line comments are denoted with hash (#), multiline with three quotes """

```python
# This is a comment
foo()

"""
This is a longer comment
"""
foo()
```
Interactive Experimentation
Interaction

• Python has an interactive console which is great for tinkering

```bash
$ python
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:13:53)
[GCC 4.5.2] on linux2
Type "help", "copyright", "credits" or "license" for more information
>>> a = 1
>>> a
1
>>> type(a)
<type 'int'>
```
Lab