

**Python: why settle for snake oil
when you can have the whole
snake?**

Shabda Raaj



About

- This is a workshop, not a talk.
- You are expected to code along.
- So pull out your laptops.
- Install python, ipython and komodo edit. (Or editor of your choice.)

Pre-requisittes

- Assume that you know programming in any language.
- So we don't spend a lot of time explaining basics.
- BUT, stop and ask if something doesn't make sense.
- AND Definitely Stop me if I am going too slow, too fast, or making no sense.

Python

- Dynamically but strongly typed.
- Very object oriented - everything is an object.
- But pragmatic - Objects aren't everything.
- Allows various paradigms of programming - OO, procedural, functional.
- Shallow learning curve, but powerful powerful capabilities available, when you need them.
- `import this`
- We will come back to this slide.

Python basics

- Hello world
- `>>> print "Hello World"`

Control Structures

- for, while, if, else, break, continue
 - -Yeah they are available, surprised?
- We will use them in a moment, but after we see the data structures available.

The Data Structures

- List - Like ArrayList in Java
- Tuple - Like List, but immutable
- Dict - Like Hashmaps in Java

The control structures

- For Loop
 - for el in iterable:
 - [block statement]
 - the classic for loop
 - for (int i; i < n; i++){}
 - for i in range(n):
 - #Work with I
 - While condition:
 - [block]
- break, continue. Normal operation - break out of current loop.

Conditionals

- If: elif: else:
 - if condition:
 - [block]
 - else:
 - [block]

The fizzbuzz test

- You have enough information now to write a solution
- Problem statement:
- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

Functions

- `def function_name(argument_list):`
 - `[block]`
- Functions can have default value.
 - `def fizzbuzz(till=100, fizz='fizz', buzz='buzz'):`
 - `#fizzbuzz code`
- Functions can have variable length values.
 - ex multiply all values passed to a function.
- Functions are first class - They are objects too.
They can be passed to other
- functions, assigned to variables etc.

Classes

- `class ClassName(base_classes):`
 - `[block]`
- Classes are first class too
- They can be passed to function, and assigned to variables.

Project Euler: problem 1

- If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.
- Find the sum of all the multiples of 3 or 5 below 1000

List Comprehensions

- The last solution was needlessly verbose
- List comprehension: Take a list and transform it.
- Standard list comprehension syntax - `[expr(i) for i in iterable if condition]`
- List of all squares: `[i*i for i in range(1,11)]`
- List of all squares of even numbers: `[i*i for i in range(1,11) if i%2 == 0]`
- So solution to last problem is just
- `sum([i*i for i in range(1,1001) if i%3 == 0 or i%5==0])`

Some functional programming

- List comprehensions are python way to do functional programming constructs
- `[function(i) for i in iterable if condition]` is `filter(func2, map(func1, iter))`
- Lets see how this list comprehension maps to functional concepts
- Get the list of squares of even numbers

File Handling

- Open a file with - `open('location')` or `file('location')`
- or give a mode - `open('location', 'rw')`
- iterate as
 - `for line in open_file.readlines():`
 - `print line` #Or whatever
- or
- `string = open_file.read()`

Some actual work

- Find the most commonly used word in the Alice in wonderland text.

A tour of the standard library

- Batteries included
 - math
 - datetime
 - string
 - re
 - random
 - os
 - pickle
- Do a `dir()` and see for yourself.
- And a lot, lot more: <http://docs.python.org/library/>

Recap of Slide 4

- Dynamically but strongly typed.
- Very object oriented - everything is an object.
- But pragmatic - Objects aren't everything.
- Allows various paradigms of programming - OO, procedural, functional.
- Shallow learning curve, but powerful powerful capabilities available, when you need them.
- `import this`

Decorators

- Syntactic sugar for
 - `foo_func = decorator_func(foo_func)`

Web development with Python

- Many useful frameworks
 - Tornado,
 - Pylons,
 - Turbogears
 - Django
 - GAE
- **Django**: Most actively developed and largest community participation

Other Good Libraries

- PIL
- Mechanize
- Beautiful Soup
- Element Tree

Resources

- python.org
- diveintopython.org
- uswaretech.com/blog

Thank you!

- Thank You.
- You can give feedback, ask questions at shabda@uswaretech.com