

# The Zen and Art of Python

## Introductory WebOps club session on Python

Surajkumar Harikumar  
Tejasvin Srinivasan  
Abdeali Kothari  
Pragathi Praveena  
Shahidh Muhammed

The WebOps Club  
IIT Madras

26 August 2014

# Outline

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

- 1 Basics
  - Background
  - Basic Syntax
  - Functions, Classes
  - Imports, Miscellany
- 2 The Zen Of Python
  - The Zen of Python
- 3 Problem
  - Problem

# Outline

# The Zen and Art of Python

## Background

## 1 Basics

- Background

- Basic Syntax

- Functions, Classes

- Imports, Miscellany

## 2 The Zen Of Python

## ■ The Zen of Python

### 3 Problem

### ■ Problem

# What is Python

The Zen and  
Art of Python

Basics

Background

Basic Syntax

Functions,  
Classes

Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

## What Python Is:

- Interpreted
- Object-Oriented
- Versatile (owing to libraries and packages)
- Focuses on easy-to-read code
- Easy to learn, Fun to use

## Quotable

"Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live." - Martin Golding

# Why Python

The Zen and  
Art of Python

Basics

Background

Basic Syntax

Functions,  
Classes

Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

- Code is 2-10x shorter than C,C++ and Java
- Code is very easy to read, and easy to learn
- Great for running small experiments
- Interactive Interpreter

# Who Uses Python

# The Zen and Art of Python

## Basics

## Background

- Google
- DropBox
- GIMP
- NASA
- CIA
- NASA

<https://wiki.python.org/moin/OrganizationsUsingPython>  
[http://en.wikipedia.org/wiki/List\\_of\\_Python\\_software](http://en.wikipedia.org/wiki/List_of_Python_software)

# Outline

The Zen and  
Art of Python

Basics

Background

**Basic Syntax**

Functions,  
Classes

Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

- 1 Basics
  - Background
  - **Basic Syntax**
  - Functions, Classes
  - Imports, Miscellany
- 2 The Zen Of Python
  - The Zen of Python
- 3 Problem
  - Problem

# Hello World

The Zen and  
Art of Python

Basics

Background

**Basic Syntax**

Functions,  
Classes

Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

That's all folks!



# Syntax

The Zen and  
Art of Python

Basics

Background  
**Basic Syntax**  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- Comments

## Variables and Types:

- Integers / Floats
- Strings
- List / Tuple
- Dictionary
- Boolean

# Playing with Variables

The Zen and  
Art of Python

Basics

Background

**Basic Syntax**

Functions,  
Classes

Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- Arithmetic
- String Operations
- AND / OR / NOT
- Comparing variables

# Conditionals, Loops and more

The Zen and  
Art of Python

Basics

Background

**Basic Syntax**

Functions,  
Classes

Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

- Indents are the new Brackets
- If / Elif / Else
- For
- While
- List Comprehension, Inline Loops

# Outline

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- 1 Basics
  - Background
  - Basic Syntax
  - **Functions, Classes**
  - Imports, Miscellany
- 2 The Zen Of Python
  - The Zen of Python
- 3 Problem
  - Problem

# Functions

The Zen and  
Art of Python

Basics

Background  
Basic Syntax

Functions,  
Classes

Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- Defining Functions
- Function Arguments
- args,kwargs

Example

# Classes

The Zen and  
Art of Python

Basics

Background

Basic Syntax

Functions,  
Classes

Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

- Defining Classes and variables
- Functions on Classes
- Initialization
- Inheritance

# Outline

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
**Imports,  
Miscellany**

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- 1 Basics
  - Background
  - Basic Syntax
  - Functions, Classes
  - Imports, Miscellany
- 2 The Zen Of Python
  - The Zen of Python
- 3 Problem
  - Problem

# Imports, Miscellany

The Zen and  
Art of Python

Basics

Background

Basic Syntax

Functions,  
Classes

Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

- One of the Most Useful features of Python
- Code Re-use
- Super Modularity
- Datetime
- Custom Code Import
- Try-except
- Documentation



# Outline

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- 1 Basics
  - Background
  - Basic Syntax
  - Functions, Classes
  - Imports, Miscellany
- 2 The Zen Of Python
  - The Zen of Python
- 3 Problem
  - Problem

# The Zen of Python

## The Zen and Art of Python

### Basics

- Background
- Basic Syntax
- Functions, Classes
- Imports, Miscellany

### The Zen Of Python

- The Zen of Python

### Problem

- Problem

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one— and preferably only one —obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *\*right\** now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea — let's do more of those!

# Beautiful is better than ugly.

## The Zen and Art of Python

### Basics

- Background
- Basic Syntax
- Functions,  
Classes
- Imports,  
Miscellany

### The Zen Of Python

- The Zen of  
Python

### Problem

- Problem

# Beautiful is better than ugly.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

Beautiful is better than ugly.

- Simple Expressions
- Indentation and Consistent Syntax
- Programs are human-friendly

# Explicit is better than implicit.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

# Explicit is better than implicit.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- Boolean Type
- Reads like English
- No unnecessary iterators / variables

# Simple is better than complex.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

# Simple is better than complex.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- Print statements
- Interactive Interpreter



# Complex is better than complicated.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

# Complex is better than complicated.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

- Break Down complex tasks
- Many many many python libraries
- Don't have to Re-invent the wheel

# Sparse is better than dense.

## The Zen and Art of Python

### Basics

- Background
- Basic Syntax
- Functions,  
Classes
- Imports,  
Miscellany

### The Zen Of Python

- The Zen of  
Python

### Problem

- Problem

# Sparse is better than dense.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

Tons of software that lets you go beyond Barebones Python

- Django
- Scipy
- Mercurial

# Readability counts.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

# Readability counts.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

- Indentation
- PEP8 Standard

# Special cases aren't special enough to break the rules.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

# Special cases aren't special enough to break the rules.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- Everything is an object



# Although practicality beats purity.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

# Although practicality beats purity.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- OOP
- Functional Programming

# Errors should never pass silently.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

# Errors should never pass silently.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- Exception Handling
- Debugging

# Unless explicitly silenced.

## The Zen and Art of Python

### Basics

- Background
- Basic Syntax
- Functions,  
Classes
- Imports,  
Miscellany

### The Zen Of Python

#### The Zen of Python

### Problem

#### Problem

# Unless explicitly silenced.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

## ■ Try-Except

# And all the rest.

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

In the face of ambiguity, refuse the temptation to guess.

There should be one— and preferably only one —obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *\*right\** now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea – let's do more of those!

# The Zen of Python Again

## The Zen and Art of Python

### Basics

- Background
- Basic Syntax
- Functions, Classes
- Imports, Miscellany

### The Zen Of Python

- The Zen of Python

### Problem

- Problem

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one— and preferably only one —obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *\*right\** now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea — let's do more of those!



# Outline

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem  
Problem

- 1 Basics
  - Background
  - Basic Syntax
  - Functions, Classes
  - Imports, Miscellany
- 2 The Zen Of Python
  - The Zen of Python
- 3 Problem
  - Problem

# Relative Grading

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

Can you write a function which, given the mark distribution for the class, computes the mean + standard deviation, and tells you your grade ?

# Wrap Up

The Zen and  
Art of Python

Basics

Background  
Basic Syntax  
Functions,  
Classes  
Imports,  
Miscellany

The Zen Of  
Python

The Zen of  
Python

Problem

Problem

Questions ?