

The Plone logo is a light blue circle containing three smaller light blue circles arranged in a triangular pattern. A horizontal line extends from the left edge of the logo across the slide.

## Forms in Plone

Steve McMahon

[steve@dcn.org](mailto:steve@dcn.org)

SteveM on #plone

smcmahon on plone.org



There's more than one way to do it.

*Larry Wall*

# Plone's Major Form Methods

---

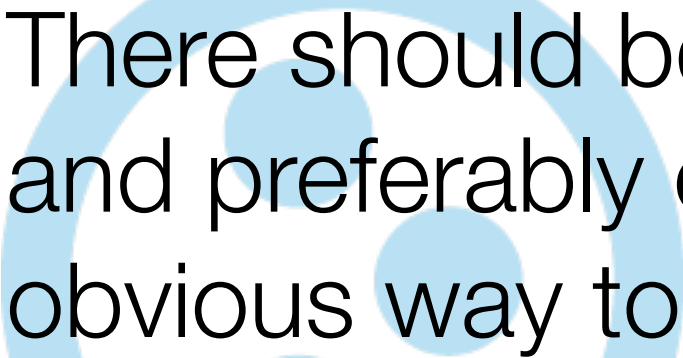
⊙ Archetypes

⊙ PloneFormGen

⊙ Templates / CMFFormController

⊙ Z3 schemas / formlib

⊙ z3c.form



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

*Tim Peters, Zen of Python*

# Archetypes — Strengths

---

- ③ Royal Road to Creating 1st-Class Portal Content
- ③ Rich Widget Set
- ③ Leverages KSS / Plone Form Conventions
- ③ ArchGenXML
- ③ Well-Known, Understood, Trained

# Archetypes — Weaknesses

---

- ⊙ Not good for standalone forms / RDBMS
- ⊙ Funky Codebase
- ⊙ Encourages Bad Habits
- ⊙ Useless Outside Plone
- ⊙ Requires Product Creation / Zope Restarts

# PloneFormGen — Strengths

---

- ⊙ Fast
- ⊙ Leverages Plone TTW conventions;  
Useful by Power Users
- ⊙ Plone-Style Forms
- ⊙ No Zope restarts

...

## PloneFormGen — Strengths, Continued

---

- ③ No File System Access Required
- ③ Flexible Security; Restricted Python
- ③ Growing List of Fields & Action Adapters
- ③ Supports Plone 2.1, 2.5, 3.x

# PloneFormGen — Weaknesses

---

- ③ Scary Hack; Inefficient
- ③ Business Logic Trapped in ZODB
- ③ Difficult to Control Formatting
- ③ Restricted Python
- ③ Requires Scripting for RDBMS
- ③ Useless Outside Plone

# Custom Templates / CMFFormController — Strengths

---

- ③ Only Solution That Works in ZODB & File System (skin layer)
- ③ Full Control of Presentation
- ③ Leverages TAL / TALES Knowledge
- ③ Restricted Python
- ③ Can Follow Plone Form Conventions Only

# Custom Templates / CMFFormController — Weaknesses

---

- ⊙ No Access to Widgets
- ⊙ Requires Knowledge Of:
  - ⊙ ZMI
  - ⊙ HTML
  - ⊙ TAL/TALES
  - ⊙ Python Scripting
  - ⊙ Plone Form Coding Conventions
- ⊙ Restricted Python

# Custom Templates / CMFFormController — Basics

---

- ③ Self-Submit Pattern
- ③ CMFFormController
  - ③ Controller Page Templates
  - ③ Validation Scripts
  - ③ Action Scripts
- ③ Plone Form Coding Pattern

# Zope 3 Style Form Generation: schemas & formlib — Strengths

---

- ③ Clean Logic
- ③ Standalone Forms or Content
- ③ Zope 3 Component Architecture / Adaptability
- ③ Unrestricted Python
- ③ Efficiency
- ③ Useful in Zope 3 Apps

# Zope 3 Style Form Generation: schemas & formlib — Weaknesses

---

- ③ Comparative Lack of Widgets
- ③ High Skill-Level Requirements
- ③ Hard to Control Formatting
- ③ Requirement to Create Packages / Restart Zope

# z3c.form / plone.z3cform — Strengths

---

- ⊙ All the strengths of formlib +
- ⊙ Widget Groups & SubForms
- ⊙ plone.z3cform Adds:
  - ⊙ Plone Templates
  - ⊙ WYSIWYG Widget
  - ⊙ CRUD Support
- ⊙ Lots of Development Energy

# z3c.form / plone.z3cform — Weaknesses

---

- ④ Too Young for Immediate Use
- ④ Requirement to Create Packages / Restart Zope

# Introduction to Form Security

---

④ Distrusting / Sanitizing Input

④ Your Form is Not Necessarily What's Posted

④ Mail Form Horror Stories

④ CSRF

# PloneFormGen Future

---

- ⊙ DC Sprint ?
- ⊙ z3c.form Implementation
- ⊙ Efficiency Improvement
- ⊙ CSRF