

Plone<sup>®</sup>  
5

Built with Passion

Modern, Powerful, and User-driven

The CMS of Theseus

Upgrade to Plone

Plone®  
5

who am i



**Upgrade to Plone**

who am i

**Cris Ewing**  
custom development and training

who am i

jazkarta

who am i

**W**

PROFESSIONAL &  
CONTINUING EDUCATION  

---

UNIVERSITY *of* WASHINGTON

who am i



**CODE FELLOWS**



"Many Hats" by Phoebe Baker, CC-BY  
[https://www.flickr.com/photos/phoebe\\_photo/2376458894/](https://www.flickr.com/photos/phoebe_photo/2376458894/)

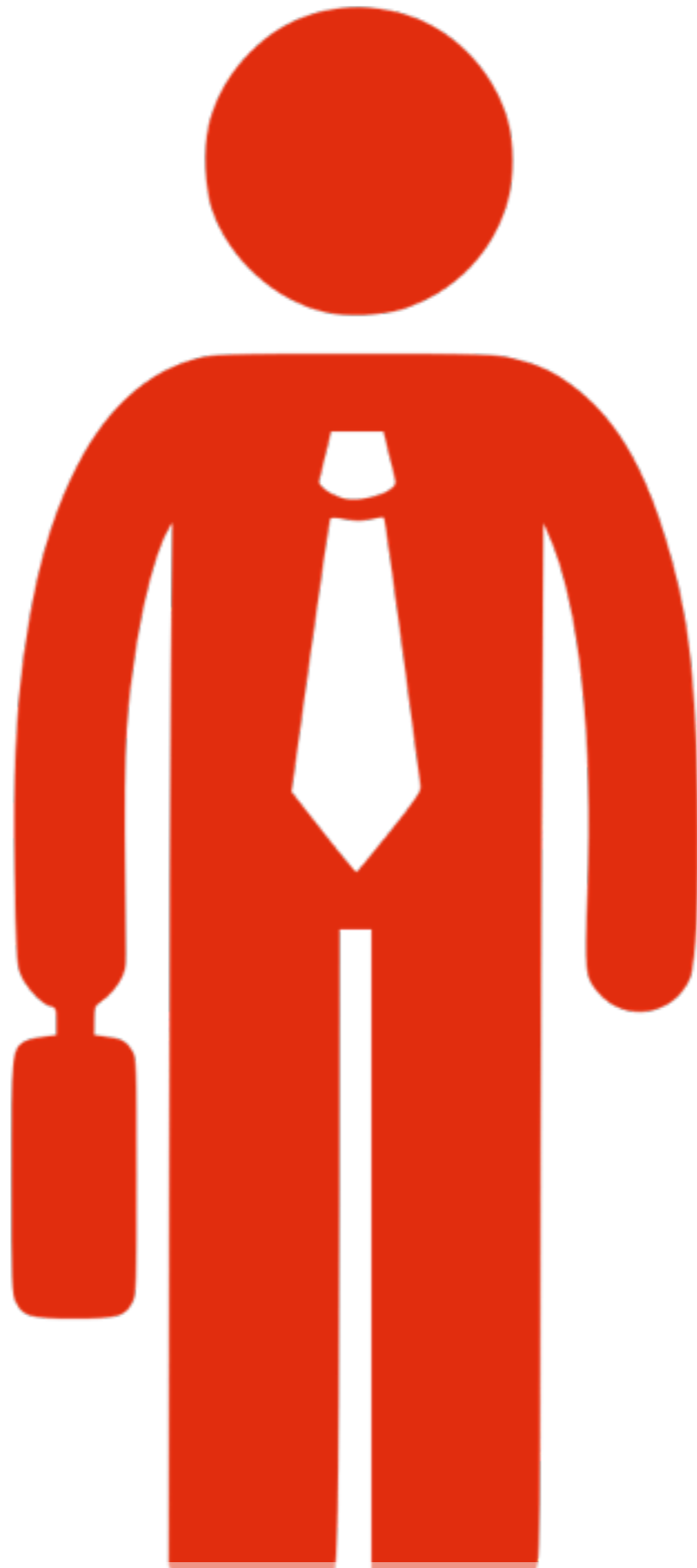


“A Famous Historian” from Monty Python and the Holy Grail





“Moss with Big Glasses” from The IT Crowd, BBC



By ben from Openclipart (Openclipart) [CC0], via Wikimedia Commons



Open Arms by Jodel Cuasay CC-BY 2.0  
<https://www.flickr.com/photos/jlcuasay/8444227885>

Think Back

Disk 1

CompuServe Information Manager  
for Windows Version 2.0.1

**CompuServe**

10 Hours  
Free\*  
online with  
this disk

To install WinCIM in Windows Program Manager, click on File, Run and type z388cp  
Note: for Windows 95, Select RUN from the START button and type z388cp in the Open box



# Jim Fulton

Plone®  
5



# Principia

Image by Paul Everitt, used with permission

Upgrade to Plone



# Paul Everitt

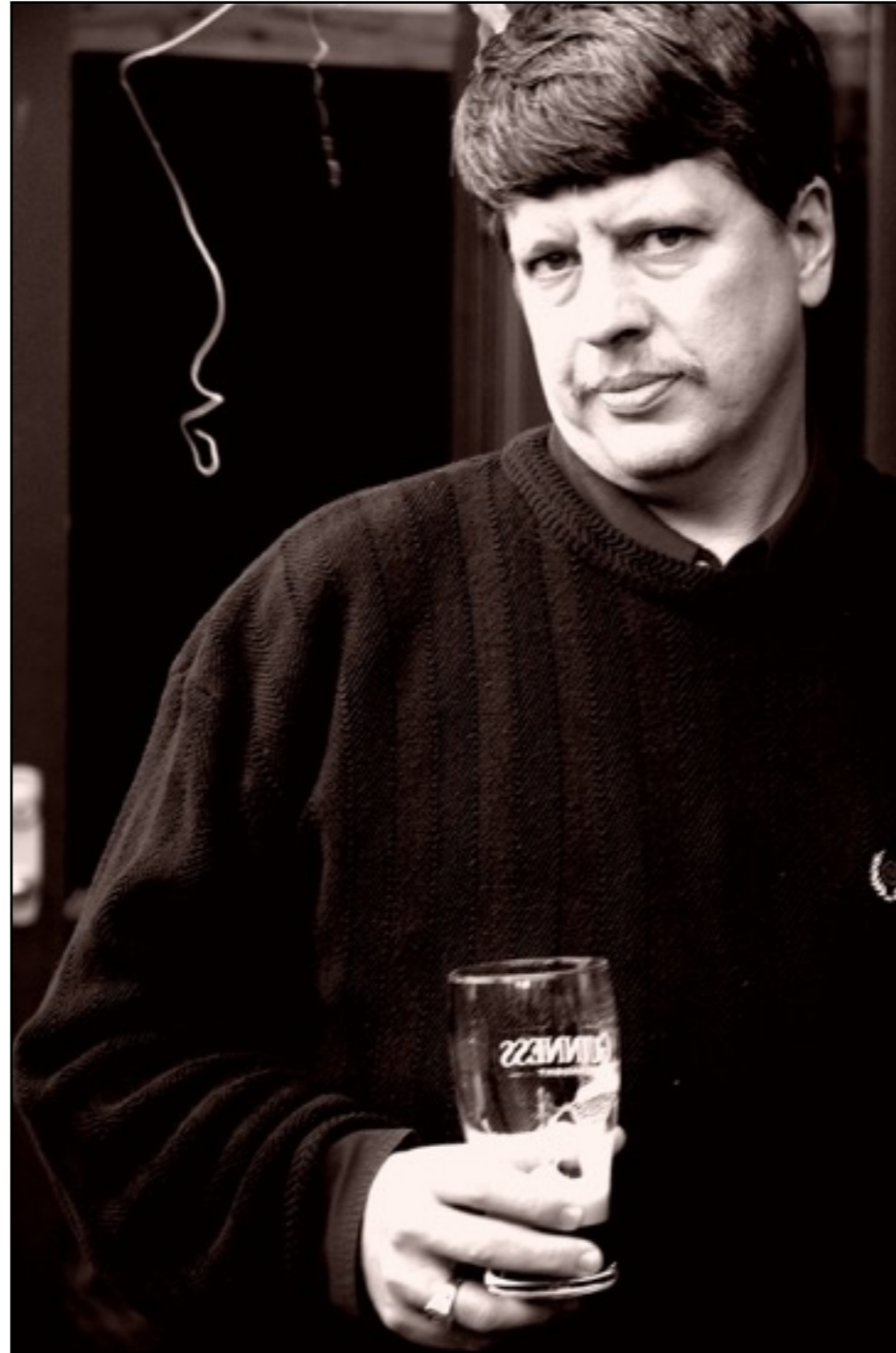
"Paul Everitt" by Christian Scholz, CC-BY-NC  
<https://www.flickr.com/photos/mrtopf/281507165/>



Plone®  
**5**



**Upgrade to Plone**



# Tres Seaver

Properties Aliases Actions

**Factory-based Type Information at /Plone/portal\_types/Discussion Item**

Properties allow you to assign simple values to Zope objects. To change property values, edit the v

Name	Value	Type
<b>Title</b>	<input type="text" value="Comment"/>	string
<b>Description</b>	<input type="text" value="Comments added to a content item."/>	text
<b>I18n Domain</b>	<input type="text" value="plone"/>	string
<b>Icon (Expression)</b>	<input type="text" value="string:\${portal_url}/discussionitem_icon.png"/>	string
<b>Product meta type</b>	<input type="text" value="Discussion Item"/>	string
<b>Product name</b>	<input type="text"/>	string
<b>Product factory</b>	<input type="text" value="plone.Comment"/>	string
<b>Add view URL (Expression)</b>	<input type="text"/>	string
<b>Add view link target</b>	<input type="text"/>	string
<b>Initial view name</b>	<input type="text"/>	string
<b>Implicitly addable?</b>	<input type="checkbox"/>	boolean
<b>Filter content types?</b>	<input checked="" type="checkbox"/>	boolean
<b>Allowed content types</b>	<input type="checkbox"/> Collection <input checked="" type="checkbox"/> Discussion Item <input type="checkbox"/> Document <input type="checkbox"/> Event <input type="checkbox"/> File <input type="checkbox"/> Folder <input type="checkbox"/> Image	multiple selection
<b>Allow Discussion?</b>	<input checked="" type="checkbox"/>	boolean

**Properties** Aliases Actions

**Factory-based Type Information at /Plone/portal\_types/Discussion Item**

Properties allow you to assign simple values to Zope objects. To change property values, edit the v

Name	Value	Type
Title	<input type="text" value="Comment"/>	string
Description	<input type="text" value="Comments added to a content item."/>	text

**Permissions** Groups Variables

**Workflow State at /Plone/portal\_workflow/simple\_publication\_workflow/states/private**

When objects are in this state they will take on the role to permission mappings defined below. Only the [permissions managed by this workflow](#) are shown.

Permission	Roles									
	Anonymous	Authenticated	Contributor	Editor	Manager	Member	Owner	Reader	Reviewer	Site Administrator
<input type="checkbox"/> Acquire permission settings?										
<input type="checkbox"/> Access contents information	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Change portal events	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Modify portal content	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> View	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

<b>Initial view name</b>	<input type="text"/>	string
<b>Implicitly addable?</b>	<input type="checkbox"/>	boolean
<b>Filter content types?</b>	<input checked="" type="checkbox"/>	boolean
<b>Allowed content types</b>	<input type="text" value="Collection"/> Discussion Item Document Event File Folder Image	multiple selection
<b>Allow Discussion?</b>	<input checked="" type="checkbox"/>	boolean

Properties Aliases Actions

**Factory-based Type Information at /Plone/portal\_types/Discussion Item**

Properties allow you to assign simple values to Zope objects. To change property values, edit the v

Name	Value	Type
Title	<input type="text" value="Comment"/>	string

Workflow State at

When objects are in this state, they are:

Permissions

Acquire permission settings?

- Access
- Change
- Modify
- View

Customize

**Filesystem Page Template at /Plone/portal\_skins/plone\_content/folder\_summary\_view**

**Id** folder\_summary\_view  
**Size** 6,095 bytes  
**Last modified** 2015/09/28 08:32:13 GMT-7  
**Source file** /Users/cewing/projects/plone\_demos/testy/buildout-cache/eggs/Products.CMFPlone-5.0-py2.7.egg/Products/CMFPlone/skins/plone\_content/folder\_summary\_view.pt  
**Customize** Select a destination folder and press the button to make a copy of this template that can be customized.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
xmlns:tal="http://xml.zope.org/namespaces/tal"
xmlns:metal="http://xml.zope.org/namespaces/metal"
xmlns:i18n="http://xml.zope.org/namespaces/i18n"
lang="en"
metal:use-macro="context/main_template/macros/master"
i18n:domain="plone">
<body>
  <metal:content-core fill-slot="content-core">
    <metal:block define-macro="listing" extend-macro="context/folder_listing/macros/content-core">
      <!-- we don't want the dl -->
      <metal:entries fill-slot="entries">
        <metal:block use-macro="context/folder_listing/macros/entries">
          <metal:entry fill-slot="entry">
            <div class="tileItem visualIEFloatFix"
              tal:define="item_object item/getObject;">
              <a href="#"
                tal:condition="exists:item_object/image_thumb"
                tal:attributes="href python:test(item_type in use_view_action, item_url+'/view', item_url)">
                <img src="" alt=""
                  tal:replace="structure python: path('nocall:item_object/tag')(scale='thumb', css_class='tileImage')" />
              </a>
              <h2 class="tileHeadline"
                metal:define-macro="listitem">
                <a href="#"
                  class="summary url"
                  tal:attributes="href python:test(item_type in use_view_action, item_url+'/view', item_url);"
                  tal:content="item_title_or_id">
                  Item Title
                </a>
            </div>
          </metal:entry>
        </metal:block>
      </metal:entries>
    </metal:block>
  </metal:content-core>
</body>
```

## Factory-based Type Information

Properties allow you to assign

**Name**

**Title**

## Filesystem Page Template

**Id** folder\_summary

**Size** 6,095 bytes

**Last modified** 2015/09/28 08:32:13 GMT-7

**Source file** /Users/cewing/projects/plone\_demos/testy/buildout-cache/eggs/Products.CMFPlone

**Customize** Select a destination folder and press the button to make a copy of this script that

custom

**Acquire permission settings?**

Access

Change

Modify

View

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:tal="http://xml.zope.org/namespaces/tal"
xmlns:metal="http://xml.zope.org/namespaces/metal"
xmlns:i18n="http://xml.zope.org/namespaces/i18n"
lang="en"
metal:use-macro="content-core"
i18n:domain="plone">
<body>
<metal:content-core file="plone_login/login_next.pt"
<metal:block define-macro="plone_login/login_next"
<!-- we don't want to use the default login form -->
<metal:entries>
<metal:block define-macro="plone_login/login_next"
<div class="login"
tal:define="
<a href="#"
tal:define="
tal:define="
<img alt="login icon"
</a>
</a>
<h2 class="login"
metal:define-macro="plone_login/login_next"
<a href="#"
</a>
```

## Filesystem Controller Python Script at /Plone/portal\_skins/plone\_login/login\_next

**Id** login\_next

**Size** 2,558 bytes

**Last modified** 2015/09/28 08:32:13 GMT-7

**Source file** /Users/cewing/projects/plone\_demos/testy/buildout-cache/eggs/Products.CMFPlone

**Customize** Select a destination folder and press the button to make a copy of this script that

custom

```
## Script (Python) "login_next"
##bind container=container
##bind context=context
##bind namespace=
##bind script=script
##bind state=state
##bind subpath=traverse_subpath
##parameters=
##title=Login next actions
##
from Products.CMFPlone import PloneMessageFactory as _
import ZTUtils

REQUEST = context.REQUEST

util = context.plone_utils
membership_tool = context.portal_membership
if membership_tool.isAnonymousUser():
    REQUEST.RESPONSE.expireCookie('__ac', path='/')
    util.addPortalMessage(_(u'Login failed'), 'error')
    return state.set(status='failure')

came_from = REQUEST.get('came_from', None)
next = REQUEST.get('next', None)

# if we weren't called from something that set 'came_from' or if HTTP_REFERER
# is the 'logged_out' page, return the default 'login_success' form
if came_from is not None:
    scheme, location, path, parameters, query, fragment = util.urlparse(came_from)
    template_id = path.split('/')[-1]
    if template_id in ['login', 'login_success', 'login_password',
        'login_failed', 'login_form', 'logged_in', 'logout',
        'logged_out', 'registered', 'mail_password',
        'mail_password_form', 'register', 'require_login',
        'member_search_results', 'pwreset_finish',
        # We need localhost in the list, or Testing.testbrowser
```

The screenshot displays the 'Plone Skins Tool' interface within a Plone 5 environment. The tool is used to manage the stack of skins used by the site. The main window shows a list of skins and their layers. The 'Plone Default' skin is currently selected as the default.

**Plone Skins Tool at /Plone/portal\_skins**

**Skin selections**

Name	Layers (in order of precedence)
<input type="checkbox"/> Plone Default	custom cmfeditions_views CMFEditions mimetypes_icons PasswordReset LanguageTool plone_ecmascript plone_wysiwyg plone_prefs plone_templates plone_form_scripts plone_scripts plone_images plone_content plone_login

**Add a new skin**

Name	Layers
<input type="text"/>	<input type="text"/>

**Default skin**

**REQUEST variable name**

**Skin flexibility**  Allow arbitrary skins to be selected

**Skin Cookie persistence**  Make skin cookie persist indefinitely

**Workflow State at**

When objects are in this state:

Permissions
<input type="checkbox"/> Access
<input type="checkbox"/> Change
<input type="checkbox"/> Modify
<input type="checkbox"/> View

**Filesystem**

Id	Size	Last modified	Source file	Customize

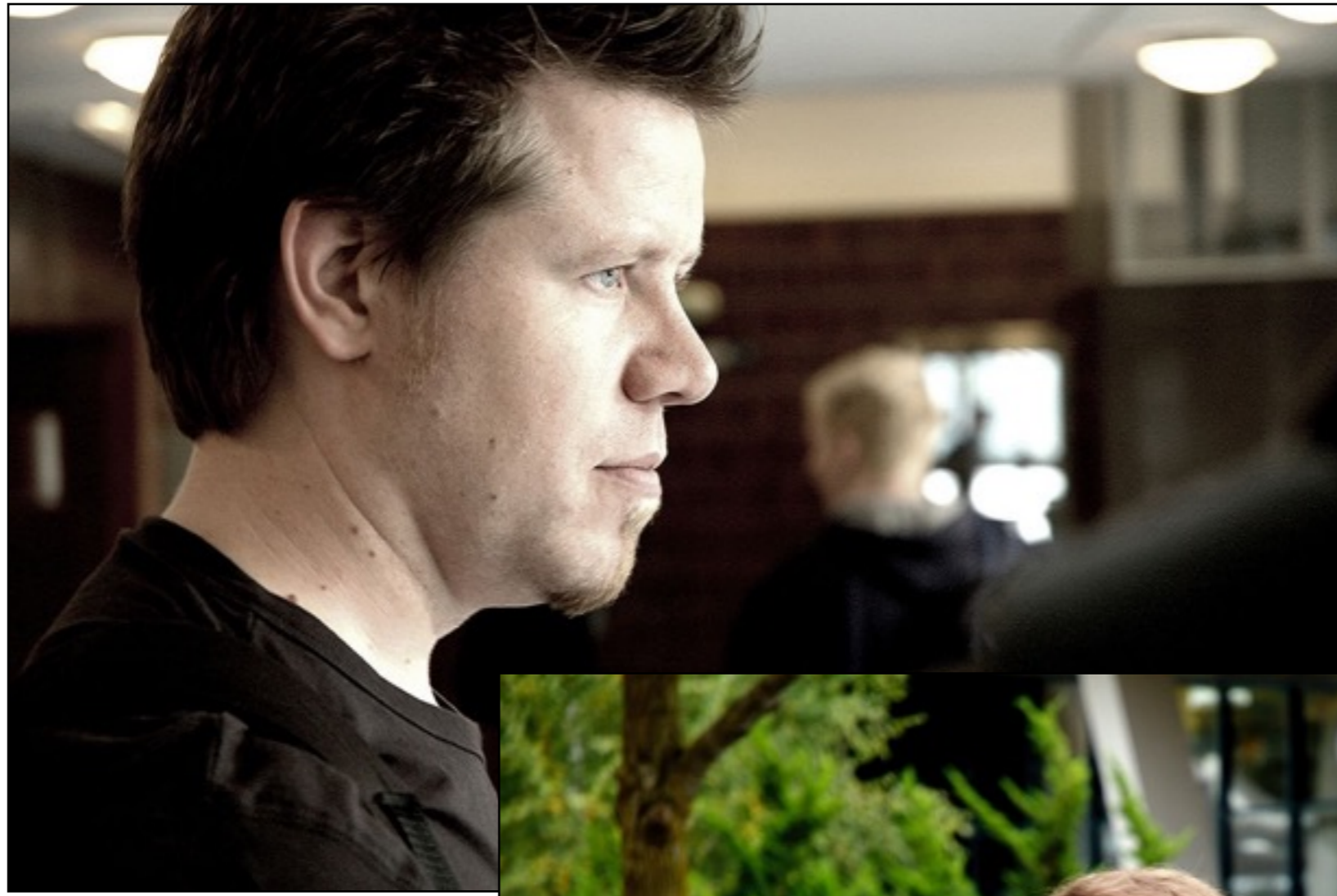
```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:metal="http://xml.zetoc.org/2002/01/metal" xmlns:i18n="http://xml.zetoc.org/2002/01/i18n" lang="en" metal:role="main" i18n:domain="plone">  
<body>  
<metal:content_view type="view">  
<metal:content_view type="view">
```

Plone®  
5




Upgrade to Plone





Mission Accomplished

[site map](#) [accessibility](#) [contact](#) [site setup](#)

 Plone™

---

home members news eventslimi preferences undo log out

you are here: home

### navigation

- Home
- Members
- News
- Events

contents view edit properties sharing

actions display add to folder state: public draft

## Welcome to Plone

by [Alexander Limi](#) — last modified 2005-08-10 04:09

**Congratulations! You have successfully installed Plone.**

If you're seeing this instead of the web site you were expecting, the owner of this web site has just installed Plone. Do not contact the Plone Team or the Plone mailing lists about this.

The first thing you should do is to set up your site by visiting the [Site Setup area](#). Become familiar with Plone by getting one of the [Plone books](#), and make sure you look at the available [add-on products](#) and [online documentation](#).

### More information

For more information on Plone:

- [The Plone Open Source Content Management System website](#)
- [What's new in Plone 2.1](#)
- [Plone documentation](#)
- [Plone add-on products](#)
- [Plone mailing lists and support forums](#)
- [Available Plone books](#) (recommended!)

### recent changes

Welcome to Plone	2005-08-10
Members	2005-08-09
News	2005-08-09
News	2005-08-09
Events	2005-08-09

### August 2005

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			





"Homeland" by Dietrich Wegner, used with permission



"Homeland" by Dietrich Wegner, used with permission

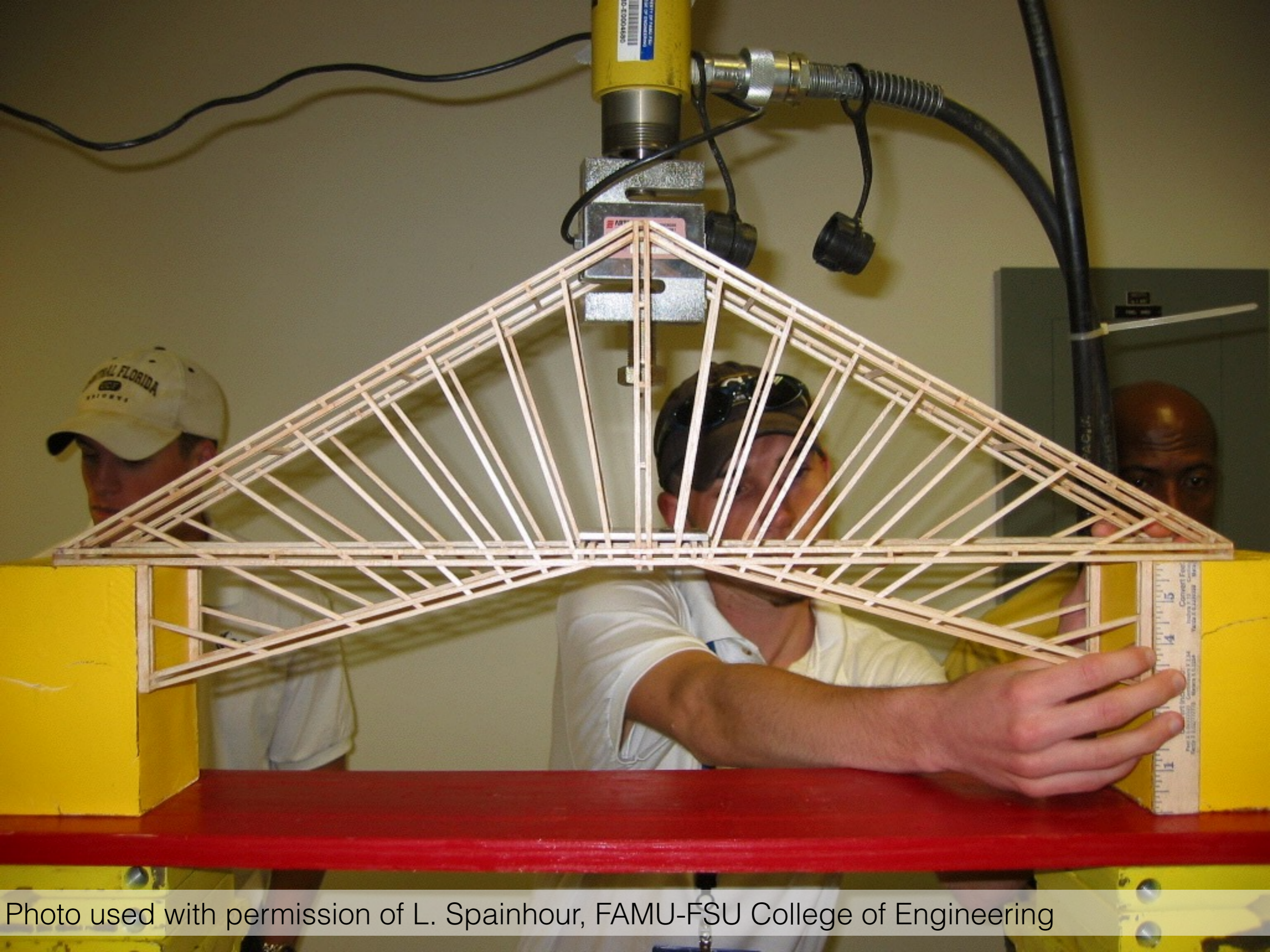


Photo used with permission of L. Spainhour, FAMU-FSU College of Engineering

**DANGER**

**EXPOSED**

**HIGH VOLTAGE**

**WATCH YOUR ASS!**



"warning sign" by Windell Oskay

<https://www.flickr.com/photos/oskay/237442629>





"Big Bada Boom" by Steve Jurveston, CC-BY  
<https://www.flickr.com/photos/jurvetson/15050728893>



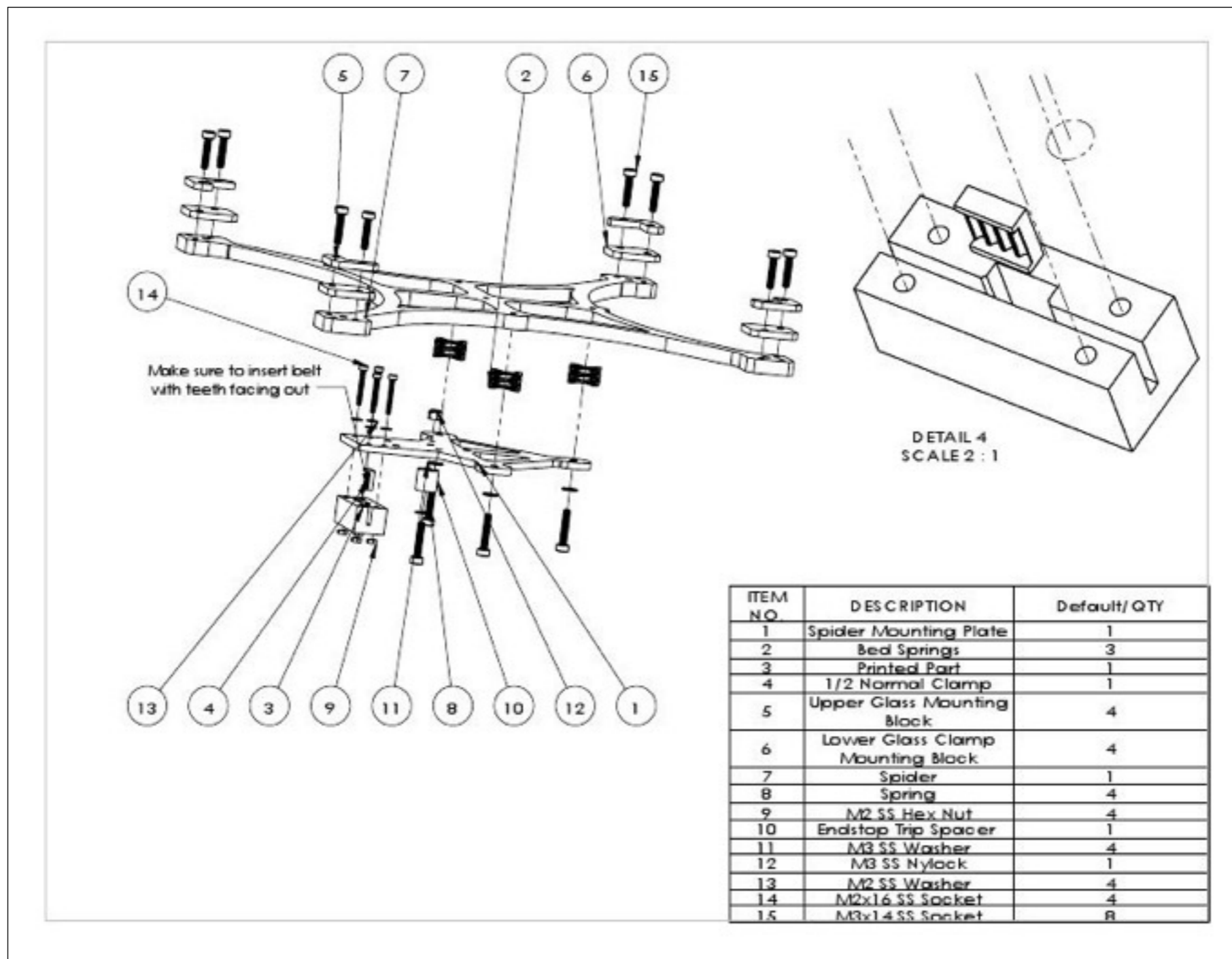
"pile of corruption" by Damian Gadai, CC-BY  
<https://www.flickr.com/photos/23024164@N06/9887010214>

reputation, reputation,  
reputation



"Not Me" by Caleb Roenigk, CC-BY  
<https://www.flickr.com/photos/crdot/6134455594>

Lockdown



# Componentization



# Packaging



***TODAY  
IS A  
GOOD DAY***





"Mouse trap" by Wayne Silver, CC-BY

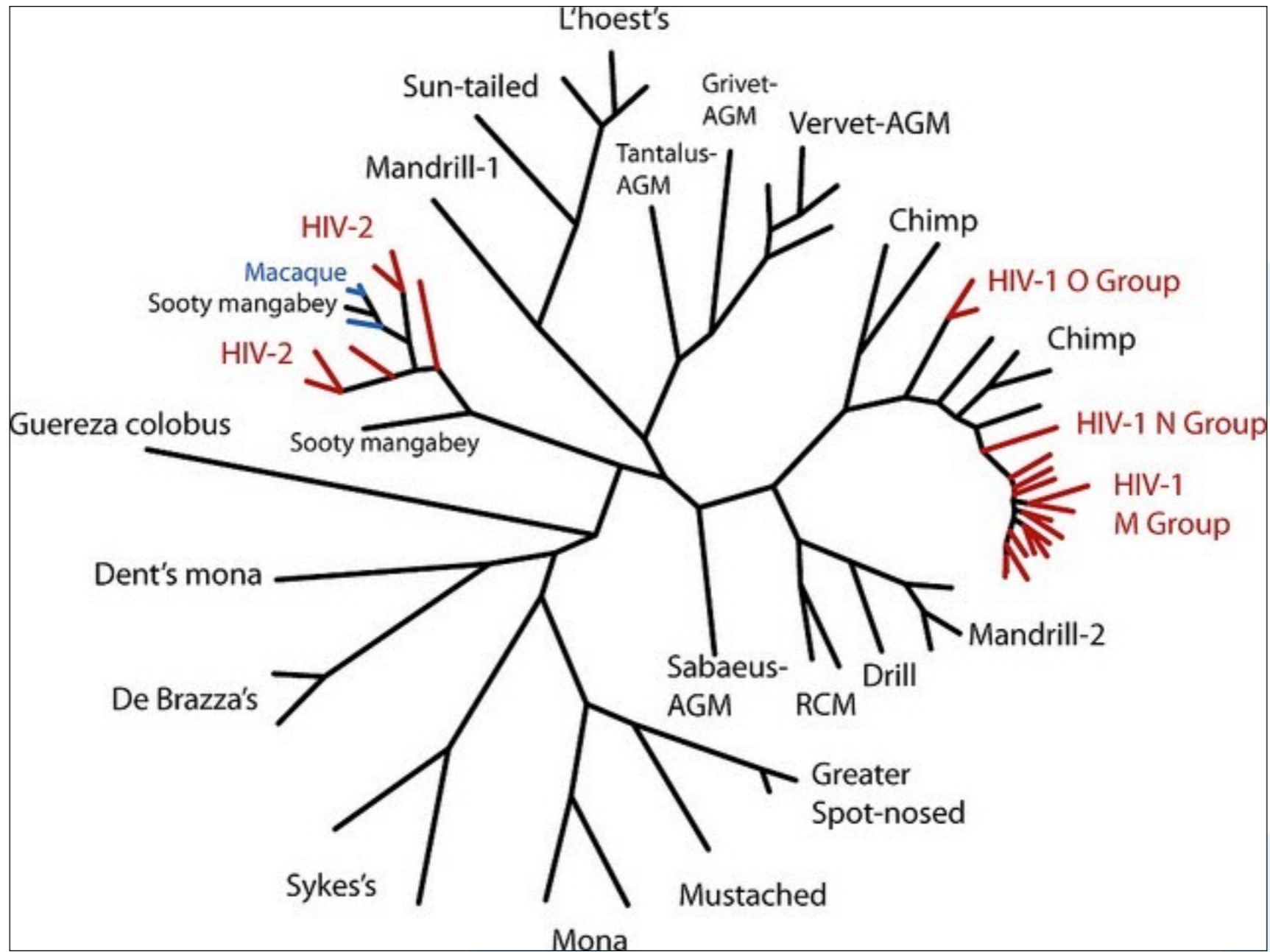
<https://www.flickr.com/photos/psycho-pics/2594617170>



# Tooling



# The Return of the Explorer



# Evolution

Plone®  
5



"The Winner" by Philo Nordlund, CC-BY  
<https://www.flickr.com/photos/philon/2420649326>

Upgrade to Plone

Plone®  
5

Plone®  
5

*Good reasons must,  
of course,  
give way to better...*



With the "Dexterity" engine, creating content types is easier than ever before.

Faster, simpler templates and dynamic theming options let you deliver content on all devices.

Full power to designers and developers: they can work with their preferred grid system and JavaScript tools.

Modern, and built to last.

## MODERN TECHNOLOGY

### Plone for today & tomorrow

Plone 5 brings the best of current technologies to Plone.

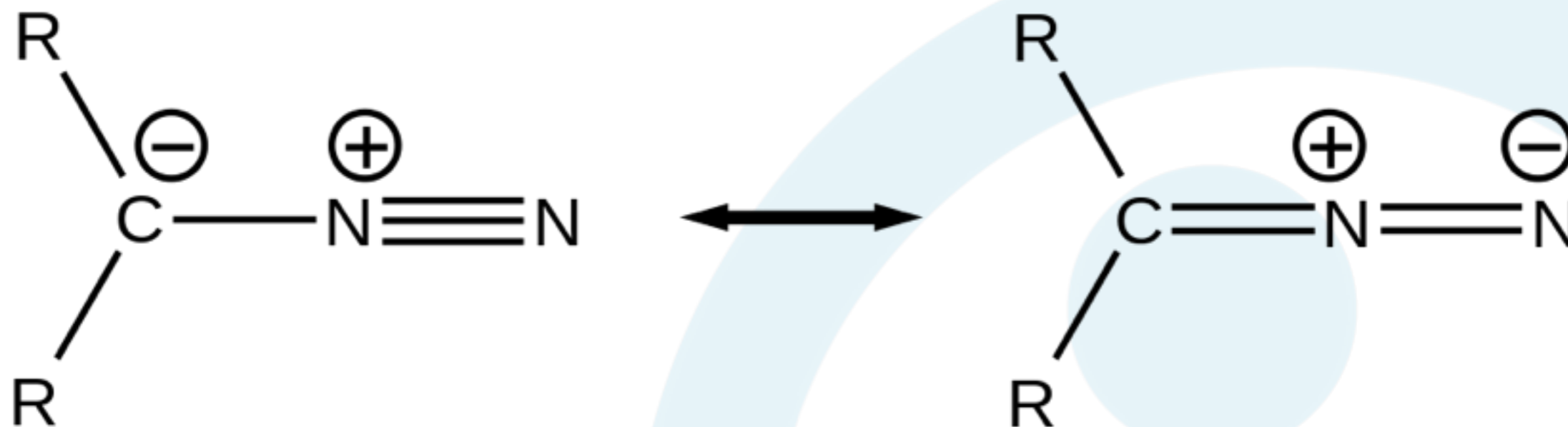
Modern, testable, standard Javascript. Responsive design. Social support.

Our quality and stability come with a clear upgrade path.

**Protect your investment.**

Upgrade to Plone

# Diazo



di-az-o (also di-az-o-type)

*noun*

a copying or coloring process using a diazo compound  
decomposed by ultraviolet light



# Traversal

<http://site/folder/page>

# Static and CGI

/site/folder/page

# Static and CGI

/site/folder/page

```
site  
└─ folder  
   └─ page
```

Plone®  
**5**

Zope

**Upgrade to Plone**



# Zope

```
import transaction
from ZODB import DB, FileStorage

connection = DB(FileStorage.FileStorage('./Data.fs')).open()
root = connection.root()

root['a_number'] = 3
root['a_string'] = 'Conference'
root['a_dictionary'] = {'10:00': 'Plone Talk', '11:00': 'REST API Talk'}

transaction.commit()
connection.close()
```

# Zope

```
import transaction
from ZODB import DB, FileStorage

connection = DB(FileStorage.FileStorage('./Data.fs')).open()
root = connection.root()
```

```
root['a_number'] = 3
root['a_string'] = 'C'
root['a_dictionary'] = {}

transaction.commit()
connection.close()
```

```
class SimpleContentObject():
    .. _container = {}

    .. def __getitem__(self, key):
    ..     return self._container[key]

    .. def __setitem__(self, key, value):
    ..     self._container[key] = value

    .. def __delitem__(self, key):
    ..     del self._container[key]
```

Zope

/site/folder/page

```
site  
└─ folder  
    └─ page
```



Zope

/site/folder/page

```
{'site': {'folder': {'page': page_object}}}
```

# Zope

`root['site']['folder']['page']`

```
{'site': {'folder': {'page': page_object}}}
```

# z Object Publishing Environment

```
def publish(request, module_name, after_list, debug=0,
            # Optimize:
            call_object=call_object,
            missing_name=missing_name,
            dont_publish_class=dont_publish_class,
            mapply=mapply,
            ):

    try:
        # ...
        object=request.traverse(path, validated_hook=validated_hook)
        # ...
        result=mapply(object, request.args, request,
                      call_object,1,
                      missing_name,
                      dont_publish_class,
                      request, bind=1)

        # ...
        if result is not response: response.setBody(result)
        # ...
        return response
    except:
        # ...
```

# z Object Publishing Environment

```
def publish(request, module_name, after_list, debug=0,
            # Optimize:
            call_object=call_object,
            missing_name=missing_name,
            dont_publish_class=dont_publish_class,
            mapply=mapply,
            ):
    try:
        # ...
        object=request.traverse(path, validated_hook=validated_hook)
        # ...
        result=mapply(object, request.args, request,
                      call_object,1,
                      missing_name,
                      dont_publish_class,
                      request, bind=1)
        # ...
        if result is not response: response.setBody(result)
        # ...
        return response
    except:
        # ...
```

# z Object Publishing Environment

```
def publish(request, module_name, after_list, debug=0,
            # Optimize:
            call_object=call_object,
            missing_name=missing_name,
            dont_publish_class=dont_publish_class,
            mapply=mapply,
            ):
    try:
        # ...
        object=request.traverse(path, validated_hook=validated_hook)
        # ...
        result=mapply(object, request.args, request,
                      call_object,1,
                      missing_name,
                      dont_publish_class,
                      request, bind=1)
        # ...
        if result is not response: response.setBody(result)
        # ...
        return response
    except:
        # ...
```

# z Object Publishing Environment

```
def publish(request, module_name, after_list, debug=0,
            # Optimize:
            call_object=call_object,
            missing_name=missing_name,
            dont_publish_class=dont_publish_class,
            mapply=mapply,
            ):

    try:
        # ...
        object=request.traverse(path, validated_hook=validated_hook)
        # ...
        result=mapply(object, request.args, request,
                      call_object,1,
                      missing_name,
                      dont_publish_class,
                      request, bind=1)

        # ...
        if result is not response: response.setBody(result)
        # ...
        return response
    except:
        # ...
```

```
def __call__(self):  
    '''  
    Invokes the default view.  
    '''  
    view = _getViewFor(self)  
    if getattr(aq_base(view), 'isDocTemp', 0):  
        return apply(view, (self, self.REQUEST))  
    else:  
        return view()
```

```
def __call__(self):  
    '''  
    Invokes the default view.  
    '''  
    view = _getViewFor(self)  
    if getattr(aq_base(view), 'isDocTemp', 0):  
        return apply(view, (self, self.REQUEST))  
    else:  
        return view()
```



```
def __call__(self):  
    '''  
    Invokes the default view.  
    '''  
    view = _getViewFor(self)  
    if getattr(aq_base(view), 'isDocTemp', 0):  
        return apply(view, (self, self.REQUEST))  
    else:  
        return view()
```











Contents Properties View

Plone Skins Tool at [/Plone/portal\\_skins](#)

### Skin selections

Name	Layers (in order of precedence)
<input type="checkbox"/> <b>Plone Default</b>	custom cmfeditions_views CMFEditions mimetypes_icons PasswordReset LanguageTool plone_ecmascript plone_wysiwyg plone_prefs plone_templates plone_form_scripts plone_scripts plone_images plone_content plone_login

Delete Save

### Add a new skin

Name	Layers
<input type="text"/>	<input type="text"/>

Add

---

**Default skin**

**REQUEST variable name**

**Skin flexibility**  Allow arbitrary skins to be selected

**Skin Cookie persistence**  Make skin cookie persist indefinitely

Save

Contents Properties View

Plone Skins Tool at [/Plone/portal\\_skins](#)

### Skin selections

Name	Layers (in order of precedence)
<input type="checkbox"/> <b>Plone Default</b>	custom cmfeditions_views CMFEditions mimetypes_icons PasswordReset LanguageTool plone_ecmascript plone_wysiwyg plone_prefs plone_templates plone_form_scripts plone_scripts plone_images plone_content plone_login

Delete Save

### Add a new skin

Name	Layers
<input type="text"/>	<input type="text"/>

Add

---

**Default skin**

**REQUEST variable name**

**Skin flexibility**  Allow arbitrary skins to be selected

**Skin Cookie persistence**  Make skin cookie persist indefinitely

Save



Contents Properties View

Plone Skins Tool at [/Plone/portal\\_skins](#)

### Skin selections

Name	Layers (in order of precedence)
<input type="checkbox"/> <b>Plone Default</b>	custom cmfeditions_views CMFEditions mimetypes_icons PasswordReset LanguageTool plone_ecmascript plone_wysiwyg plone_prefs plone_templates plone_form_scripts plone_scripts plone_images plone_content plone_login

Delete Save

### Add a new skin

Name	Layers
<input type="text"/>	<input type="text"/>

Add

---

**Default skin**

**REQUEST variable name**

**Skin flexibility**  Allow arbitrary skins to be selected

**Skin Cookie persistence**  Make skin cookie persist indefinitely

Save

 Filesystem Page Template at [/Plone/portal\\_skins/plone\\_content/folder\\_summary\\_view](#) 

**Id** folder\_summary\_view

**Size** 6,095 bytes

**Last modified** 2015/09/28 08:32:13 GMT-7

**Source file** /Users/cewing/projects/plone\_demos/testy/buildout-cache/eggs/Products.CMFPlone-5.0-py2.7.egg/Products

**Customize** *Select a destination folder and press the button to make a copy of this template that can be customized.*

custom



Customize

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
xmlns:tal="http://xml.zope.org/namespaces/tal"
xmlns:metal="http://xml.zope.org/namespaces/metal"
xmlns:i18n="http://xml.zope.org/namespaces/i18n"
lang="en"
metal:use-macro="context/main_template/macros/master"
i18n:domain="plone">

<body>

  <metal:content-core fill-slot="content-core">
    <metal:block define-macro="listing" extend-macro="context/folder_listing/macros/content-core">
      <!-- we don't want the dl -->
      <metal:entries fill-slot="entries">
        <metal:block use-macro="context/folder_listing/macros/entries">
          <metal:entry fill-slot="entry">
            <div class="tileItem visualIEFloatFix"
              tal:define="item_object item/getObject;">

              <a href="#"
                tal:condition="exists:item_object/image_thumb"
                tal:attributes="href python:test(item_type in use_view_action, item_url+'/view', it
                  <img src="" alt=""
                    tal:replace="structure python: path('nocall:item_object/tag')(scale='thumb',
              </a>

            <h2 class="tileHeadline"
              metal:define-macro="listitem">
```

Filesystem Page Template at [/Plone/portal\\_skins/plone\\_content/folder\\_summary\\_view](#)

**Id** folder\_summary\_view

**Size** 6,095 bytes

**Last modified** 2015/09/28 08:32:13 GMT-7

**Source file** /Users/cewing/projects/plone\_demos/testy/buildout-cache/eggs/Products.CMFPlone-5.0-py2.7.egg/Products

**Customize** *Select a destination folder and press the button to make a copy of this template that can be customized.*

custom



Customize

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
xmlns:tal="http://xml.zope.org/namespaces/tal"
xmlns:metal="http://xml.zope.org/namespaces/metal"
xmlns:i18n="http://xml.zope.org/namespaces/i18n"
lang="en"
metal:use-macro="context/main_template/macros/master"
i18n:domain="plone">
```

```
<body>
```

```
<metal:content-core fill-slot="content-core">
```

```
<metal:block define-macro="listing" extend-macro="context/folder_listing/macros/content-core">
```

```
<!-- we don't want the dl -->
```

```
<metal:entries fill-slot="entries">
```

```
<metal:block use-macro="context/folder_listing/macros/entries">
```

```
<metal:entry fill-slot="entry">
```

```
<div class="tileItem visualIEFloatFix"
```

```
tal:define="item_object item/getObject;">
```

```
<a href="#"
```

```
tal:condition="exists:item_object/image_thumb"
```

```
tal:attributes="href python:test(item_type in use_view_action, item_url+'/view', it
```

```
<img src="" alt=""
```

```
tal:replace="structure python: path('nocall:item_object/tag')(scale='thumb',
```

```
</a>
```

```
<h2 class="tileHeadline"
```

```
metal:define-macro="listitem">
```



aye, there's the rub...

Sarah Bernhardt as Hamlet, with Yorick's skull  
(Photographer: James Lafayette, c. 1885–1900)

Don't Do That

ZCA

**“Program to an interface, not an implementation”**

–Design Patterns: Elements of Reusable Object-Oriented Software

Consider a Faucet



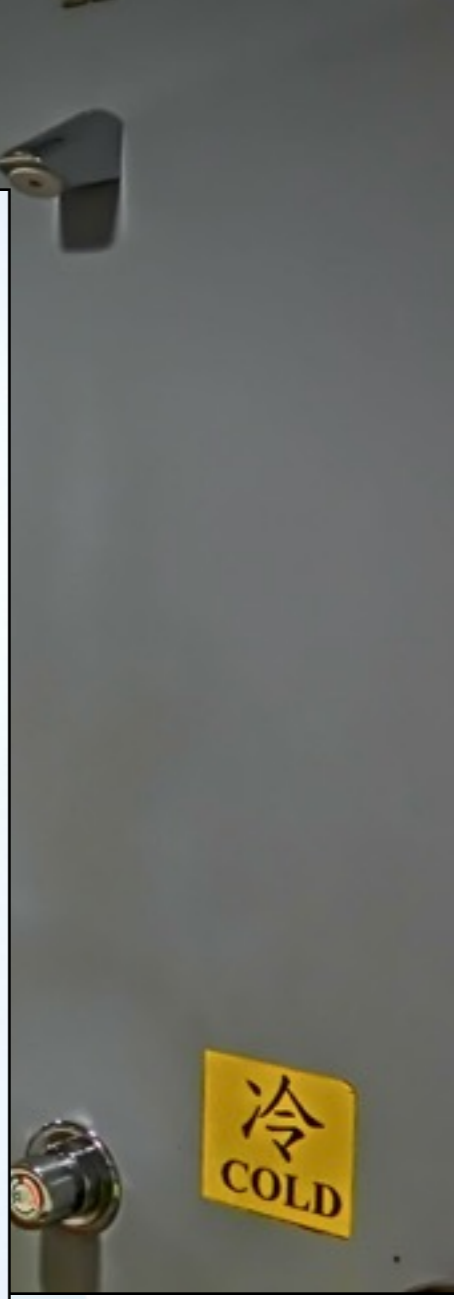


# Different Contexts





Plone®  
51



Upgrade to Plone

# Adapters

```
from zope.interface import Interface, Attribute
from zope.interface import implements
from zope.component import adapts, getAdapter
from zope.component import getGlobalSiteManager
```

```
from zope.interface import Interface, Attribute
from zope.interface import implements
from zope.component import adapters, getAdapter
```

```
class IWaterSupply(Interface):
    hot = Attribute("hot water")
    cold = Attribute("cold water")

class IFaucet(Interface):
    hot_water_percentage = Attribute("percent of water flow from hot")
    flow_rate = Attribute("flow rate of water in liters per second")

    def change_temperature(delta_hot):
        """update hot_water_percentage by delta_hot"""

    def change_flow(target_flow_rate):
        """update flow rate to target from current"""
```



```
from zope.interface import Interface, Attribute
from zope.interface import implements
from zope.component import adapts, getAdapter
```

```
class IWaterSupply(Interface):
    hot = Attribute("hot water")
    cold = Attribute("cold water")
```

```
class IFaucet(Interface):
    hot_water_percent = Attribute("hot water percent")
    flow_rate = Attribute("flow rate")

    def change_temperature(self, delta_hot):
        """up"""

    def change_flow(self, target_flow_rate):
        """up"""
```

```
class House(object):
    implements(IWaterSupply)
    # details of implementation
```

```
class KitchenSinkFaucet(object):
    implements(IFaucet)
    adapts(IWaterSupply)
```

```
def __init__(self, water_supply):
    self.supply = water_supply
```

```
def change_temperature(self, delta_hot):
    self.hot_water_percent += delta_hot
```

```
def change_flow(self, target_flow_rate):
    self.flow_rate = target_flow_rate
```

```
from zope.interface import Interface, Attribute
from zope.interface import implements
from zope.component import adapters, getAdapter
```

```
class IWaterSupply(Interface):
    hot = Attribute("hot water")
    cold = Attribute("cold water")
```

```
class IFaucet(Interface):
    hot_water_percent = Attribute("hot water percent")
    flow_rate = Attribute("flow rate")

    def change
```

```
class House(object):
    implements(IWaterSupply)
    # details of implementation

    def change_temperature(self, delta_hot):
    def change_flow(self, target_flow_rate):
```

```
gsm = getGlobalSiteManager()
gsm.registerAdapter(KitchenSinkFaucet, (IWaterSupply), name='sink')

>>> my_house = House()
>>> my_sink = getAdapter(my_house, IFaucet, 'sink')
>>> my_sink.__class__ == KitchenSinkFaucet
True
```

```
    def change_temperature(self, delta_hot):
        self.hot_water_percent += delta_hot

    def change_flow(self, target_flow_rate):
        self.flow_rate = target_flow_rate
```

# Browser Views are Adapters

# Browser Views are Adapters

They adapt  
Context *and* Request

Plone®  
**5**



**Upgrade to Plone**

```
from zope.browser.interfaces import IBrowserView
```

```
class MyView(object):
```

```
    implements(IBrowserView)
```

```
    def __init__(context, request):
```

```
        self.context = context
```

```
        self.request = request
```

```
    def __call__(self):
```

```
        text = "I am MyView called with {request} in the context of {context}"
```

```
        return text.format(**{request: self.request, context: self.context})
```

```
from zope.browser.interfaces import IBrowserView
```

```
class MyView(object):
```

```
    implements(IBrowserView)
```

```
    def __init__(context, request):
```

```
        self.context = context
```

```
        self.request = request
```

```
    def __call__(self):
```

```
        text = "I am MyView called with {request} in the context of {context}"
```

```
        return text.format(**{request: self.request, context: self.context})
```

```
<configure
```

```
    xmlns="http://namespaces.zope.org/zope"
```

```
    xmlns:browser="http://namespaces.zope.org/browser"
```

```
>
```

```
    <browser:page
```

```
        for="*"
```

```
        name="myview"
```

```
        permission="zope2.Public"
```

```
        class=".views.MyView"
```

```
    />
```

```
</configure>
```

```
from zope.browser.interfaces import IBrowserView
```

```
class MyView(object):
```

```
    implements(IBrowserView)
```

```
    def __init__(context, request):
```

```
        self.context = context
```

```
        self.request = request
```

```
    def __call__(self):
```

```
        text = "I am MyView called with {request} in the context of {context}"
```

```
        return text.format(**{request: self.request, context: self.context})
```

```
<configure
```

```
    xmlns="http://namespaces.zope.org/zope"
```

```
    xmlns:browser="http://namespaces.zope.org/browser"
```

```
>
```

```
    <browser:page
```

```
        for="*"
```

```
        name="myview"
```

```
        permission="zope2.Public"
```

```
        class=".views.MyView"
```

```
    />
```

```
>>> PloneSite = request.Traverse('/')
```

```
>>> view = getAdapter((PloneSite, request), IBrowserView, name="myview")
```

```
>>> view()
```

```
"I am MyView called with <request> in the context of <Plonesite>"
```



```
<configure
  xmlns="http://namespaces.zope.org/zope"
  xmlns:browser="http://namespaces.zope.org/browser"
  >

  <browser:page
    for="*"
    name="myview"
    permission="zope2.Public"
    class=".views.MyView"
  />

</configure>
```

```
<configure
  xmlns="http://namespaces.zope.org/zope"
  xmlns:browser="http://namespaces.zope.org/browser"
  >
```

```
<browser:page
```

```
<configure
  xmlns="http://namespaces.zope.org/zope"
  xmlns:browser="http://namespaces.zope.org/browser"
  >
```

```
</configure>
  <browser:page
    for="IDocument"
    name="myview"
    permission="zope2.Public"
    class=".views.MyView"
    template="templates/myview.pt"
  />
</configure>
```

```
<configure
  xmlns="http://namespaces.zope.org/zope"
  xmlns:browser="http://namespaces.zope.org/browser"
  >
```

```
<browser:page
```

```
<configure
  xmlns="http://namespaces.zope.org/zope"
  xmlns:browser="http://namespaces.zope.org/browser"
  >
```

```
</con... <browser:page
  for="IDocument"
```

```
<configure
  xmlns="http://namespaces.zope.org/zope"
  xmlns:browser="http://namespaces.zope.org/browser"
  >
```

```
<browser:page
  for="IDocument"
  name="myview"
  permission="zope2.Public"
  class=".views.MyView"
  template="templates/myview.pt"
  layer=".interfaces.IMyPackageIsActive"
  />
```

```
</configure>
```

# So What?

# So What?

- Content Objects have a default view name: “view”

# So What?

- Content Objects have a default view name: “view”
- The publisher finds browser view adapters by name

# So What?

- Content Objects have a default view name: “view”
- The publisher finds browser view adapters by name
- Make a more specific registration, yours is picked

# So What?

- Content Objects have a default view name: “view”
- The publisher finds browser view adapters by name
- Make a more specific registration, yours is picked
- Adapters mean clean decoupling

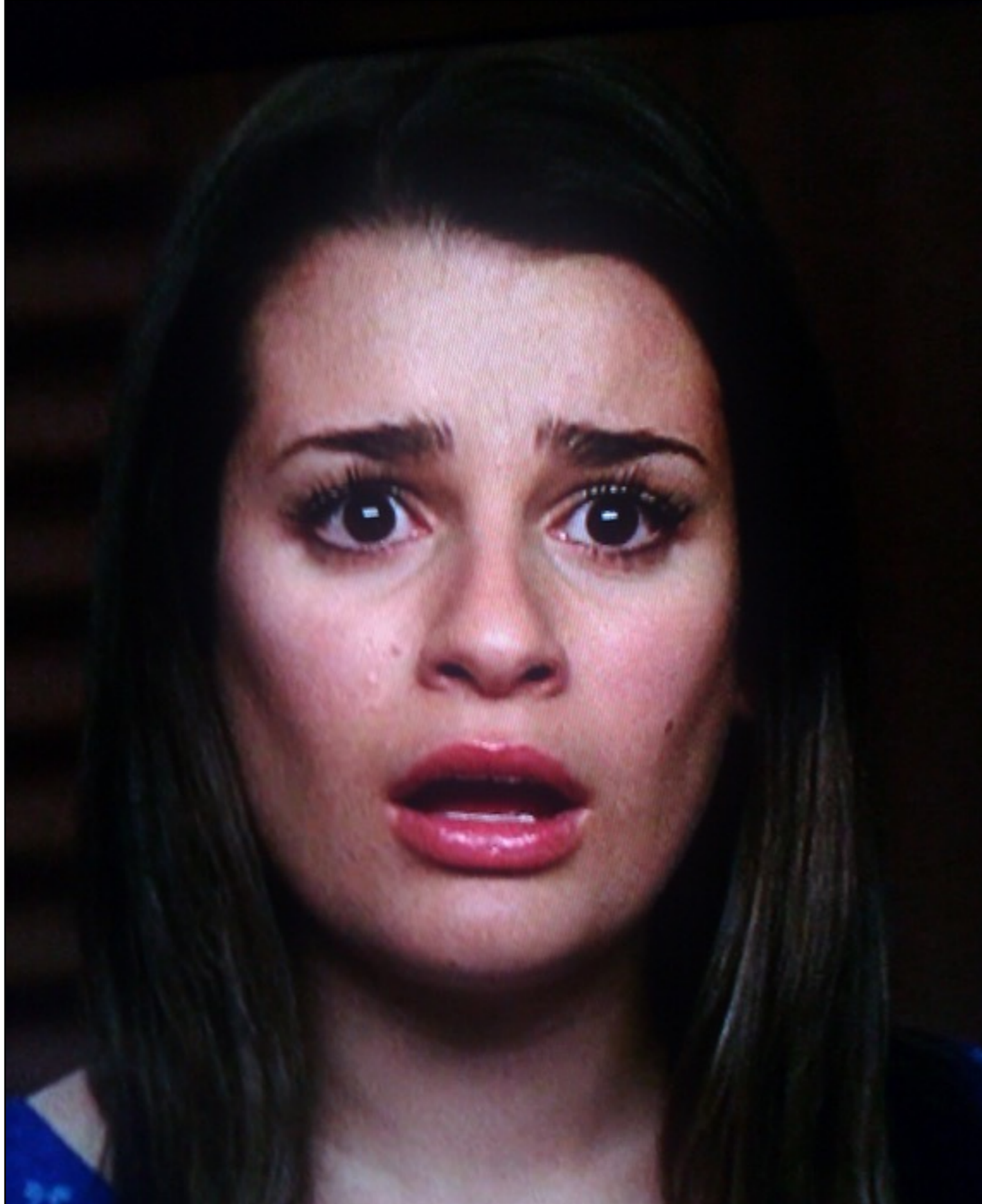




"Applauded" by Matthew Wilkinson, CC-BY  
<https://www.flickr.com/photos/manc72/7964710328>

except...

Plone®  
5



"Glee - Wait what?" by karlnorling, CC-BY  
<https://www.flickr.com/photos/karlnorling/5513291897>

Upgrade to Plone



“All problems in computer science can be solved by another layer of indirection”

—David Wheeler

Templates == Layout

Plone®  
**5**



**Upgrade to Plone**

## Important News!

Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Quam euismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit dui vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel. In molestie elit dui dictum, praesent nascetur pulvinar sed, in dolor pede in aliquam, risus nec error quis pharetra. Eros metus quam augue suspendisse, metus rutrum risus erat in. In ultrices quo ut lectus, etiam vestibulum urna a est, pretium luctus euismod nisl, pellentesque turpis hac ridiculus massa.

Venenatis a taciti dolor platea, curabitur lorem platea urna odio, convallis sit pellentesque lacus proin.



**This is a caption**

### **Contact:**

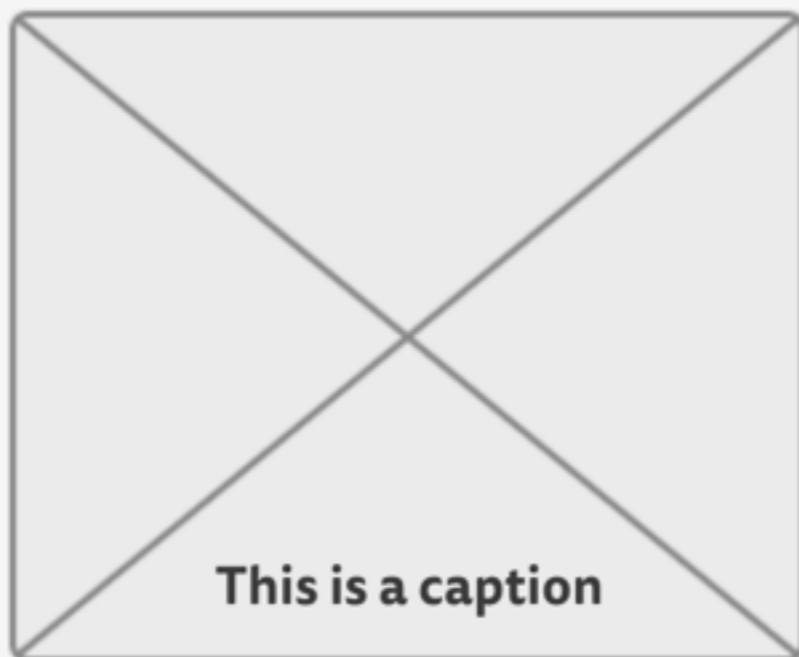
[C J Cregg, Press Secretary](#)



## Important News!

Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at

nunc.  
pellen  
vestib  
nullan  
praese  
aliqua  
quam  
in. In  
urna a  
pellen  
Vener  
platea  
proin.



## Important News!

Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Quam eismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit duis vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel. In molestie elit dui dictum, praesent nascetur pulvinar sed, in dolor pede in aliquam, risus nec error quis pharetra. Eros metus quam augue suspendisse, metus rutrum risus erat in. In ultrices quo ut lectus, etiam vestibulum urna a est, pretium luctus eismod nisl, pellentesque turpis hac ridiculus massa. Venenatis a taciti dolor platea, curabitur lorem platea urna odio, convallis sit pellentesque lacus proin.

For More, Contact [CJ Cregg, Press Secretary](#)

Templates == Layout

Templates  $\neq$  Layout?

Layout == Layout

transform layout

# xslt

**XSLT** (Extensible Stylesheet Language Transformations)

a language for transforming XML documents into other XML documents

Diazo: we write xslt so  
you don't have to

```
<theme href="index.html" />
```



```
<replace  
  content="selector"  
  theme="selector" />
```

# Selectors

# Selectors

**XPath** theme="/html/head/title"

# Selectors

**XPath** theme="/html/head/title"

**CSS** css:content="div#title"

# Diazo Rules

# Diazo Rules

- theme

# Diazo Rules

- theme
- replace

# Diazo Rules

- theme
- replace
- before



# Diazo Rules

- theme
- replace
- before
- after

# Diazo Rules

- theme
- replace
- before
- after
- drop

# Diazo Rules

- theme
- replace
- before
- after
- drop
- strip

# Diazo Rules

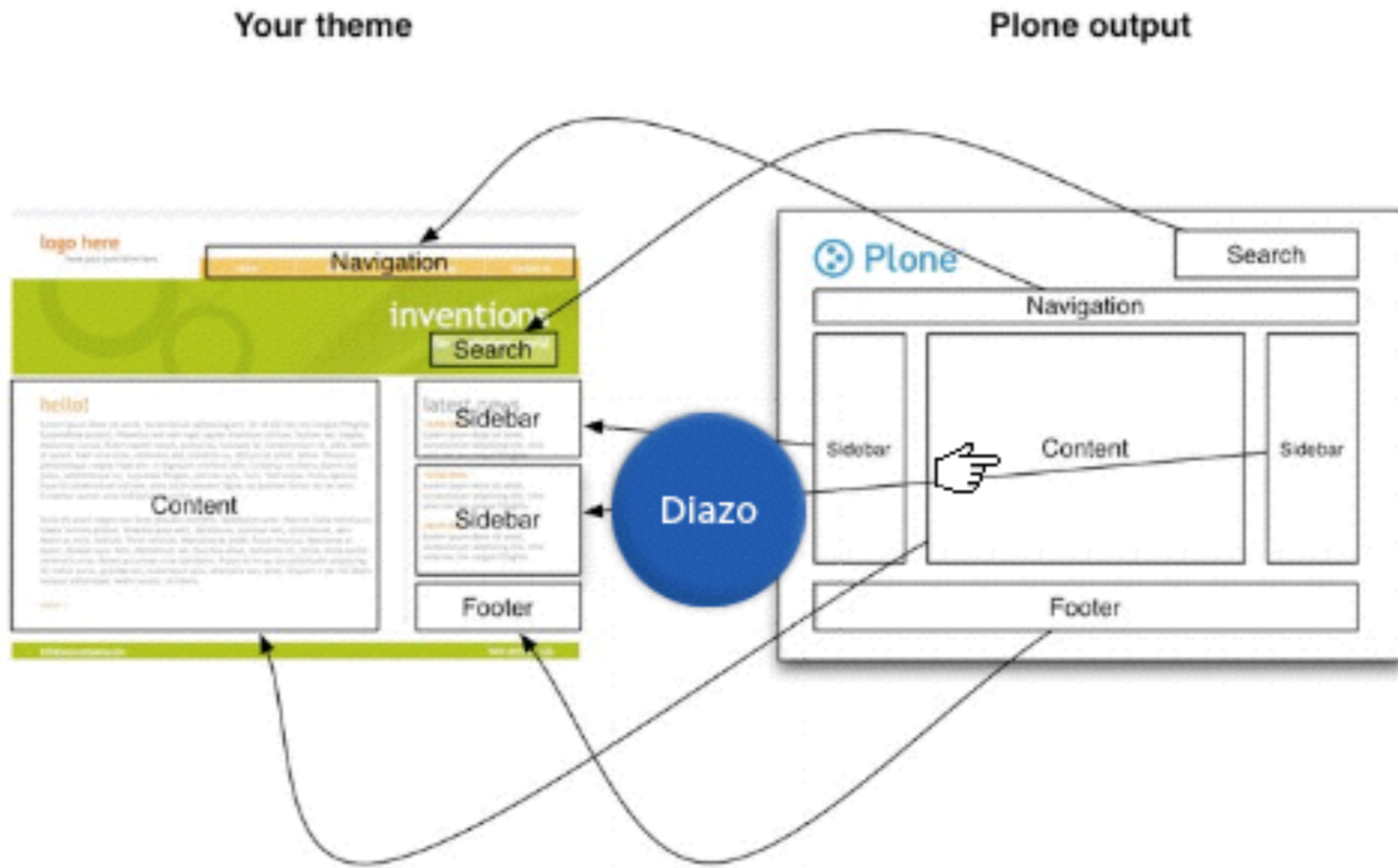
- theme
- replace
- before
- after

- drop
- strip
- merge

# Diazo Rules

- theme
- replace
- before
- after

- drop
- strip
- merge
- copy



```
mytheme/  
├── css  
│   └── mytheme.css  
├── index.html  
├── js  
│   └── mytheme.js  
└── rules.xml
```

# Use Your Own Tools



But I **HATE** xml

# Theming Demo

Plone®

5



Plone®  
**5**

**Upgrade to Plone**



- Patternslib

**Upgrade to Plone**

# Plone<sup>®</sup> 5

- Patternslib
- Dexterity

**Upgrade to Plone**

# Plone<sup>®</sup> 5

- Patternslib
- Dexterity
- Behaviors

**Upgrade to Plone**



- Patternslib
- Dexterity
- Behaviors
- Multi-lingual Content

A decorative graphic in the bottom right corner consisting of a large, light blue arc and three smaller, light blue circles of varying sizes, arranged in a pattern that suggests a stylized 'P' or a similar shape.

**Upgrade to Plone**



# Plone<sup>®</sup> 5

- Patternslib
- Dexterity
- Behaviors
- Multi-lingual Content
- Plone API

**Upgrade to Plone**

# Plone<sup>®</sup> 5

- Patternslib
- Dexterity
- Behaviors
- Multi-lingual Content
- Plone API
- and much, much more....!

**Upgrade to Plone**

# Theseus' CMS

what's old is new again

Much Work Remains

# Python 3 Compatibility

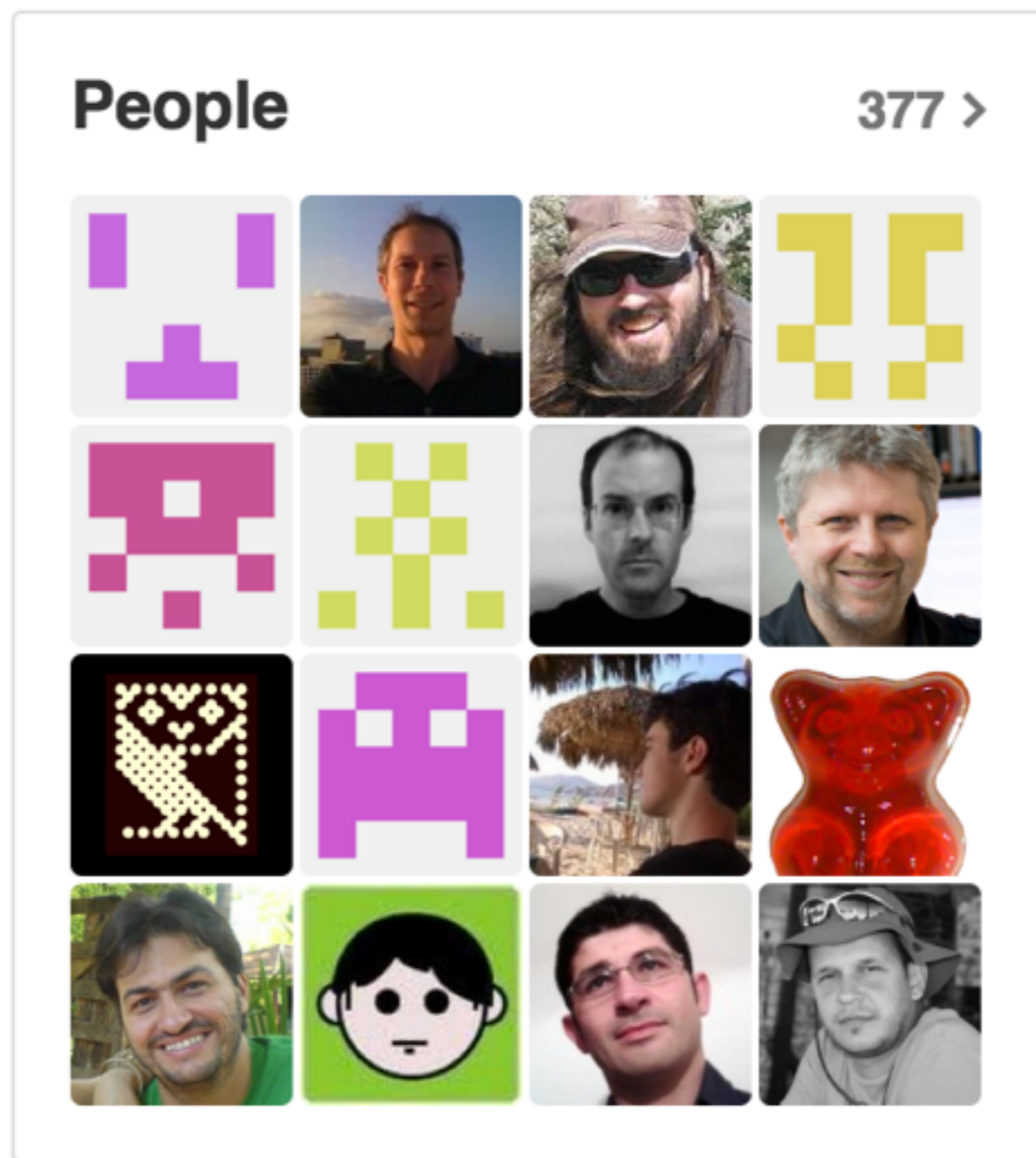
# Layer Cleanup

# Opportunity Knocks

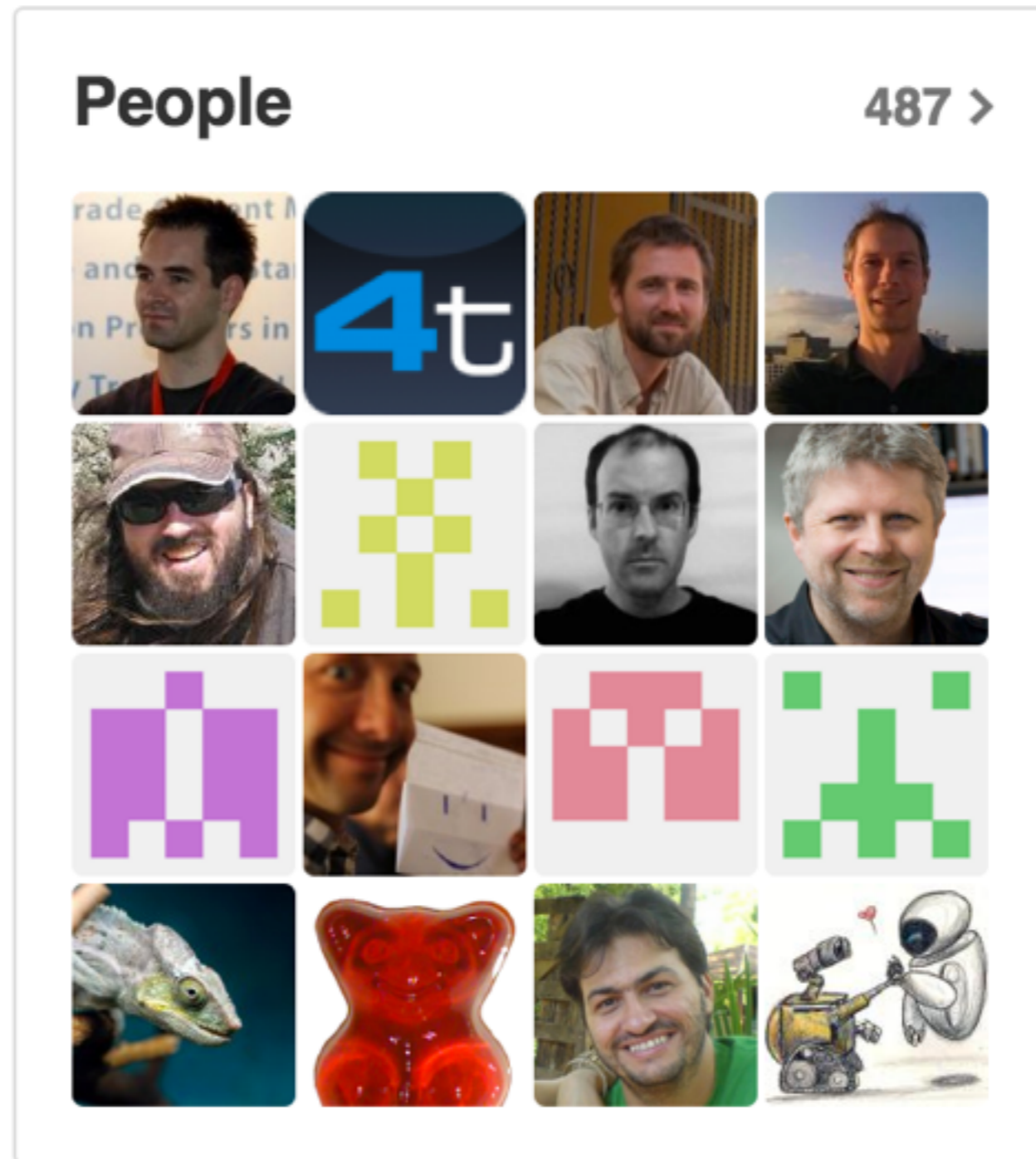


# Community

# Core Contributors



# Plone Collective



And You??

# Demos on Request

<http://plone.com>



Thanks Due



**Upgrade to Plone**



# Thanks Due

- Our PiterPy Organizers

**Upgrade to Plone**





# Thanks Due

- Our PiterPy Organizers
- Eric Steele

**Upgrade to Plone**



# Thanks Due

- Our PiterPy Organizers
- Eric Steele
- Paul Roeland

A decorative graphic in the bottom right corner consisting of a large, light blue arc and three smaller, light blue circles arranged in a pattern.

**Upgrade to Plone**



# Thanks Due

- Our PiterPy Organizers
- Eric Steele
- Paul Roeland
- Brandon Rhodes

**Upgrade to Plone**

# Thanks Due

- Our PiterPy Organizers
- Eric Steele
- Paul Roeland
- Brandon Rhodes
- Eric Brehault, Jens Klein, T. Kim Nguyen, Philip Bauer, Martin Aspeli, Johannes Raggam, Christine McNeill, Dylan Jay

# Additional Credits

"Limi" by Christian Scholz, CC-BY-NC

<https://www.flickr.com/photos/mrtopf/22101363>

"Alan Runyan" by Christian Scholz, CC-BY-NC

<https://www.flickr.com/photos/mrtopf/280446628>

"the kitchen sink" by Alan Cleaver, CC-BY

<https://www.flickr.com/photos/alancleaver/3727870484>

"Shower at Victoria Park Swimming Pool" by Malesworldoa, CC-BY-SA

[https://commons.wikimedia.org/wiki/File:HK\\_CWB\\_%E7%B6%AD%E5%A4%9A](https://commons.wikimedia.org/wiki/File:HK_CWB_%E7%B6%AD%E5%A4%9A%E5%88%A9%E4%BA%9E%E5%85%AC%E5%9C)  
[%E5%88%A9%E4%BA%9E%E5%85%AC%E5%9C](https://commons.wikimedia.org/wiki/File:HK_CWB_%E7%B6%AD%E5%A4%9A%E5%88%A9%E4%BA%9E%E5%85%AC%E5%9C)

[%92%E6%B8%B8%E6%B3%B3%E6%B1%A0\\_old\\_Victoria\\_Park\\_Swimming\\_P](https://commons.wikimedia.org/wiki/File:HK_CWB_%E7%B6%AD%E5%A4%9A%E5%88%A9%E4%BA%9E%E5%85%AC%E5%9C)  
[ool\\_%E7%94%B7%E6%9B%B4%E8%A1%A3%E5%AE](https://commons.wikimedia.org/wiki/File:HK_CWB_%E7%B6%AD%E5%A4%9A%E5%88%A9%E4%BA%9E%E5%85%AC%E5%9C)

[%A4\\_Changing\\_Room\\_shower\\_bathing\\_heads\\_Sept-2013.JPG">%A4\\_Changing\\_Room\\_shower\\_bathing\\_heads\\_Sept-2013.JPG](https://commons.wikimedia.org/wiki/File:HK_CWB_%E7%B6%AD%E5%A4%9A%E5%88%A9%E4%BA%9E%E5%85%AC%E5%9C)

"Cat in a washing machine" by James Cridland, CC-BY

<https://www.flickr.com/photos/jamescridland/2663785397>



# Plone Conference 2016

October 17-24, 2016  
Boston, MA, USA  
[ploneconf.org](http://ploneconf.org)

Upgrade to Plone

# Cris Ewing

@crisewing  
cris@crisewing.com  
cewing