ZOPE

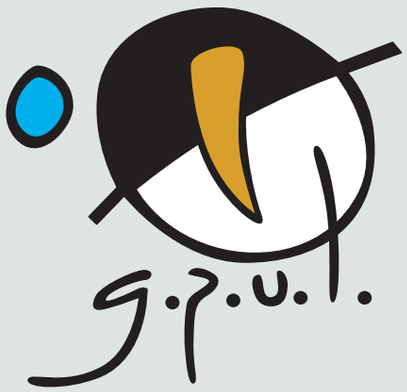
Sistema Operativo Linux
V Jornadas

Desarrollo de aplicaciones web con

Zope

Adrián Pérez de Castro – moebius@eml.cc

V Jornadas GPUL sobre el Sistema Operativo Linux



ZOPE

Sistema Operativo Linux
V Jornadas

Aplicaciones web con Zope

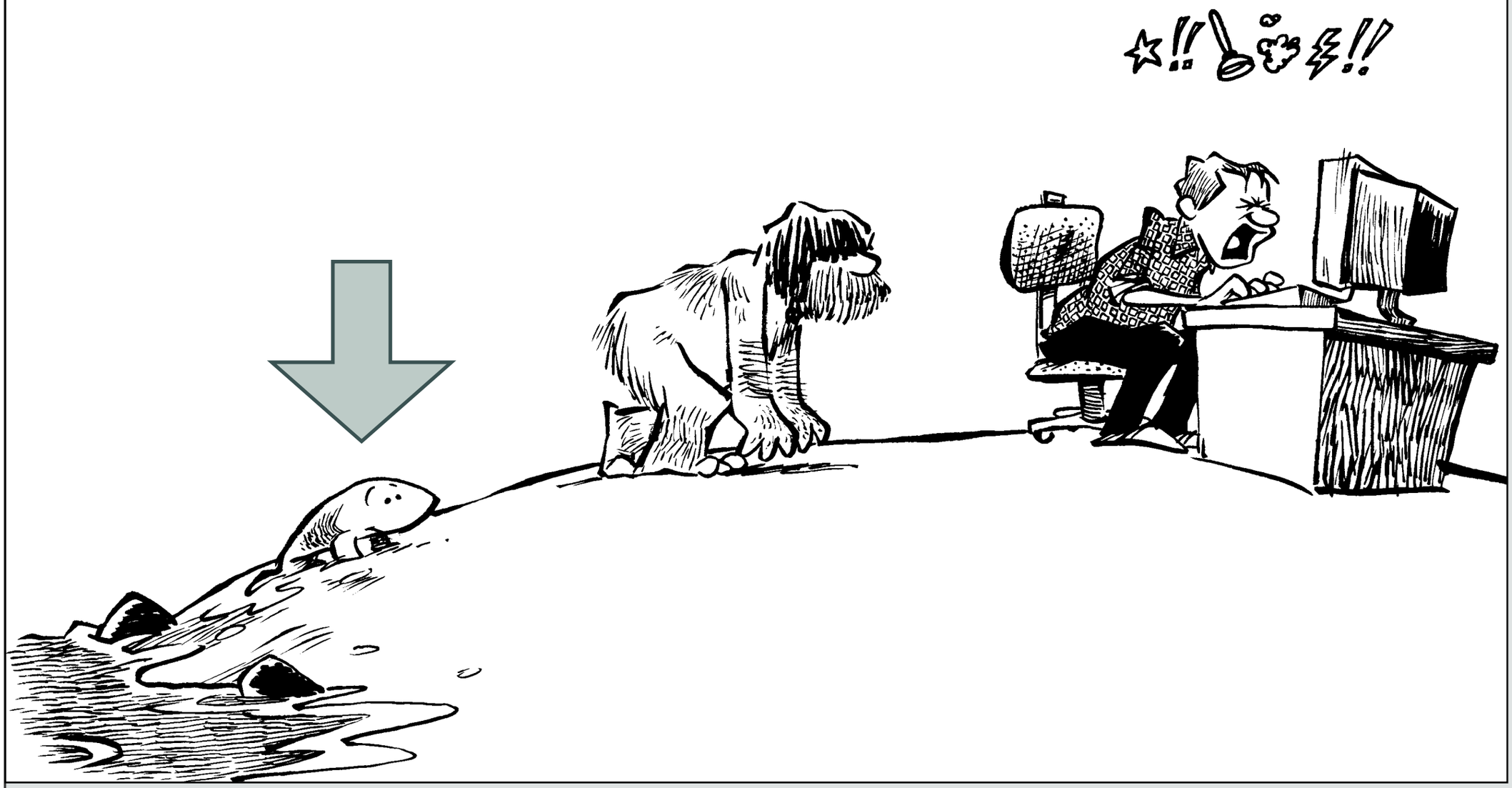
- ¿Qué es Zope?
- Zope Básico
  - El primer paso: Instanciar Zope
  - Creación y edición de contenido
  - Dinamismo: ZPT, DTML, Python
  - Al César lo que es del César: Adquisición
- El Zen de Zope
  - Más allá de lo predefinido: Productos
  - El ojo que todo lo ve: Usuarios y Roles

ZOPE

Sistema Operativo Linux

V Jornadas

¿Qué es Zope?



De cómo alguien que se encuentra con una tecnología decide mejorarla por considerarla poco útil

ZOPE

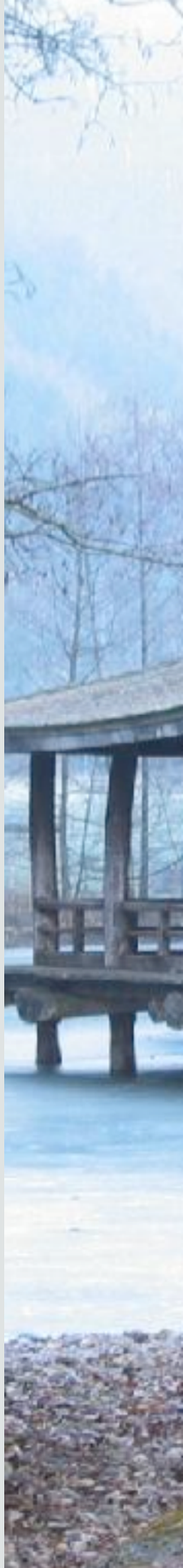
El nacimiento – 1

Charla sobre CGI de Jim Fulton, 1996

- Antes de la charla: Jim no sabía sobre CGI
- Camino de la charla: Jim leyó lo que pudo sobre CGI
- Después de la charla: Jim no quería saber nada de CGI

Viaje de vuelta

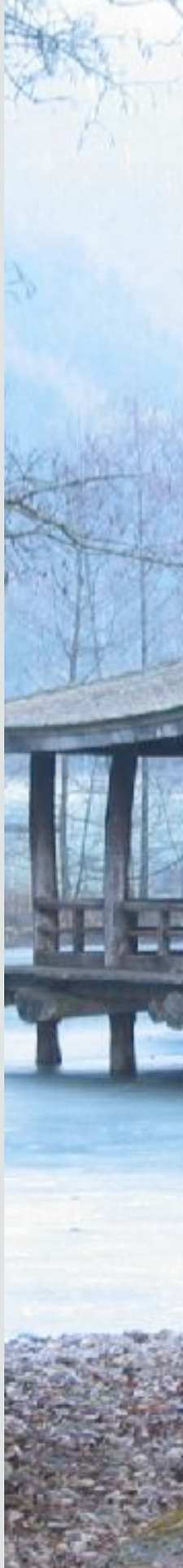
- Los aviones tienen mesas
- En las mesas se pueden apoyar portátiles
- Los portátiles sirven para programar
- Programar permite hacer nuevos programas



ZOPE

El nacimiento – 2

- Jim Fulton trabajaba en Digital Creations
  - Prototipo implementado «on-the-fly»
  - Componentes «Open Source»:
    - Bobo (sistema de publicación)
    - Document Template (plantillas de texto)
    - BoboPOS (base de datos)
  - Producto comercial: Principia
- Principia se hace libre: Noviembre 1998
  - Hadar Pedhazur convence a a Digital Creations para liberarlo
  - Los componentes de Principia forman la base de Zope



ZOPE

Zope es... – 1

...un «framework»

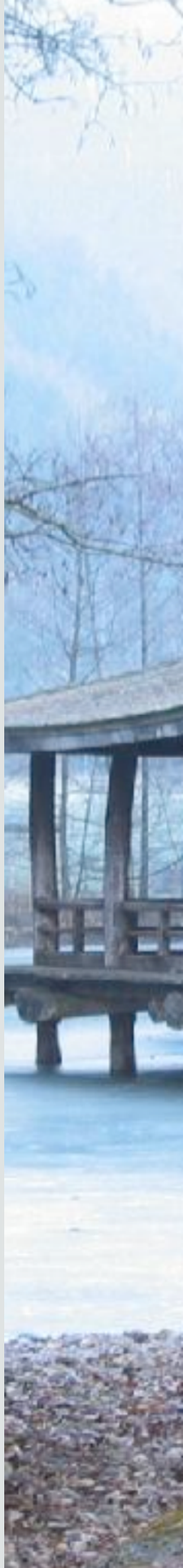
- Incluye de serie servicios que toda aplicación web necesita
- El programador no necesita saber los detalles escabrosos

...orientado a objetos

- Python es orientado a objetos
- Todo es un objeto

...publicación de objetos

- Petición = búsqueda de un objeto + invocación de método
- Invocar un método típicamente devuelve HTML



ZOPE

Zope es... – 2

...manejable a través de la web

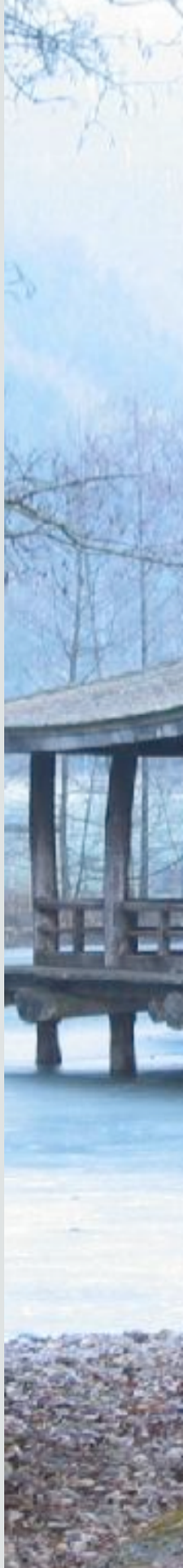
- Zope se puede administrar con (casi) cualquier navegador
- Se maneja sin acceso al sistema de archivos del servidor

...capaz de delegar responsabilidades

- Permite especificar políticas de seguridad complejas
- Permisos según: objeto, clase del objeto, rol del usuario...

...persistente

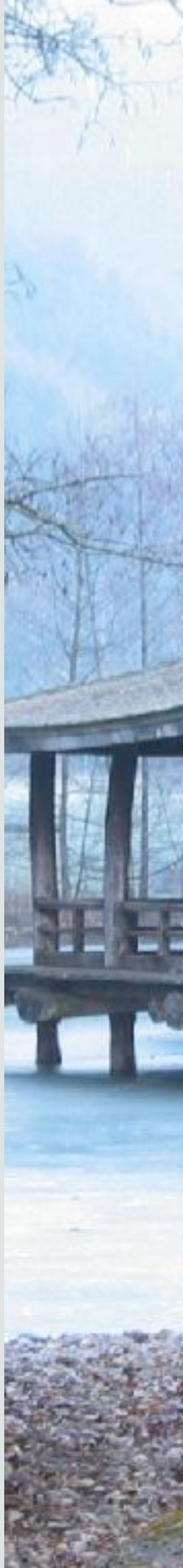
- BBDD almacena objetos de forma transparente
- Cada petición es una transacción: soporte para deshacer



ZOPE

Zope es... — 3

- ...extensible
  - A través de la web
  - Creando nuevas clases en Python
- ...«adquisitivo»
  - Un objeto puede «adquirir» atributos y comportamiento
  - Fácil especialización de partes de una aplicación web
- ...software libre
  - Esto es lo más importante, 8^)



ZOPE

Zope **NO** es... – 1

...fácil de aprender y enseñar

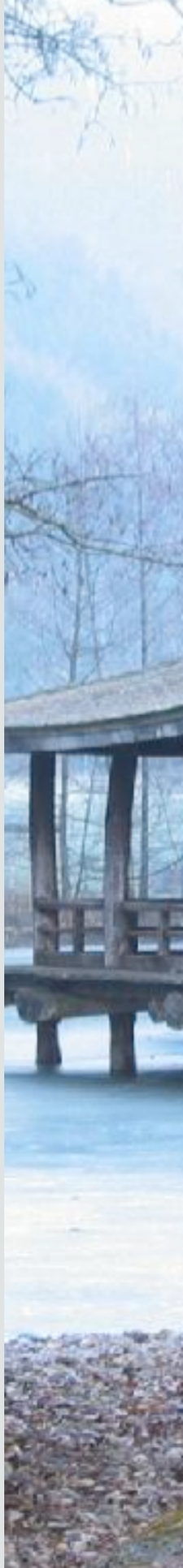
- Hay conceptos a los que cuesta habituarse
- Es un sistema complejo
- Poca documentación sobre ciertos aspectos

...extremadamente rápido

- Pero funciona mejor que otros con recursos limitados
- Python es rápido, pero podría serlo más
- Más velocidad con Psyco (JIT), a costa de utilizar más RAM

...«chascarrillo» diario de la cafetería

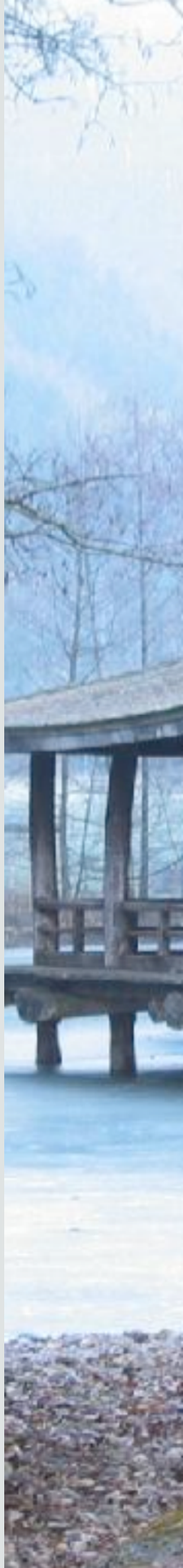
- Quizás a partir de hoy B^)



ZOPE

Zope **NO** es... – 2

- ...la solución a todos los males
  - Sigue haciendo falta diseñar y programar
  - Puede haber gente con «Pyfobia» (nota: hay PHP y Perl)
- ...un servidor web al uso
  - Para esto ya tenemos Apache
  - Es mejor combinarlo con Apache: ¡más complejidad aún!
- ...especialmente respetuoso con «áéíóúñ»
  - El soporte internacional es relativamente reciente
  - Quedan componentes que trabajan mal con texto no–ASCII

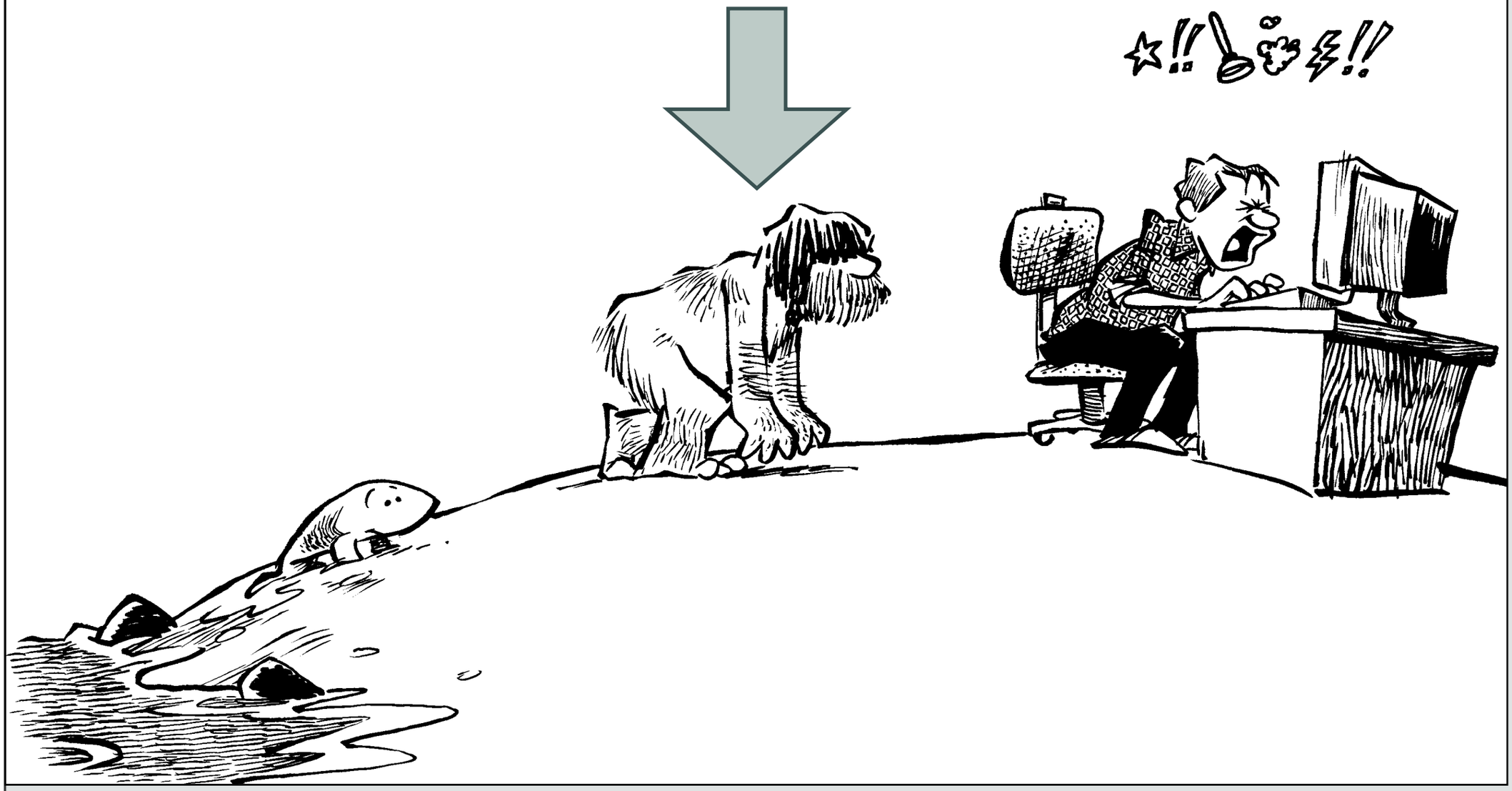


ZOPE

Sistema Operativo Linux

V Jornadas

Zope Básico



Instanciación, creación, edición, adquisición, dinamismo

Instanciación

Nuevo sitio = «Instancia» de Zope

- % mkzopeinstance.py -d .
- % $EDITOR etc/zope.conf

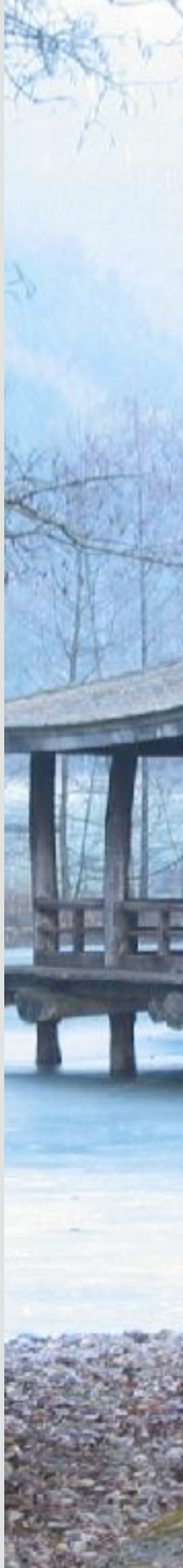
Ejecutar ZServer (método 1)

- % bin/runzope
- Ctrl-C

Ejecutar ZServer (método 2)

- % bin/zopectl start
- % bin/zopectl stop

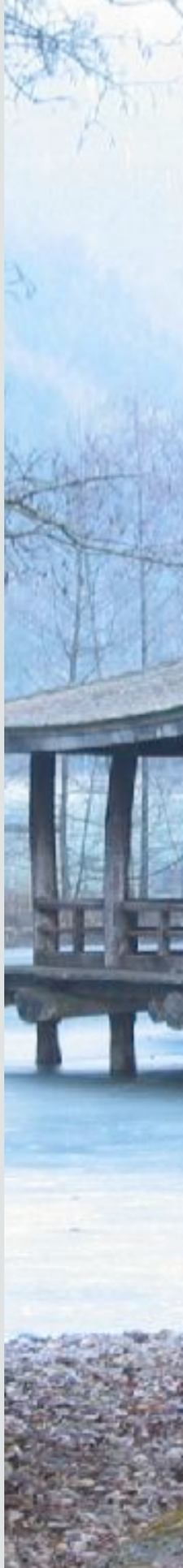
Demo



Cómo acceder a Zope

- Vía web – Zope Management Interface
  - Basta con apuntar a http://localhost:8080
- Con un cliente de FTP (!)
  - `% ftp localhost 8021`
  - Ideal para subir imágenes y archivos
- Con un cliente WebDAV (!!)
  - `% cadaver http://localhost:8080`
  - No se obtiene el fuente: WebDAV-source en «zope.conf»
    - `% cadaver http://localhost:9800`
  - Especialmente cómodo 8^)

Demo



Crear y editar – 1

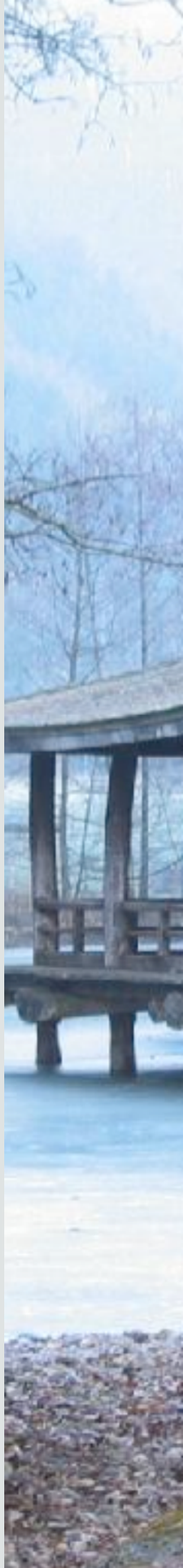
Creación

- Desplegar el menú que reza «Select type to add...»
- Elegir una opción
- Rellenar el formulario

Edición

- Navegar hasta el elemento que se desea editar
- Elegir la pestaña «Edit»
- Guardar los cambios

Demo



Crear y editar – 2

A través de FTP

```
ftp> mkdir fotos1
ftp> cd fotos
ftp> mput *.jpg
```

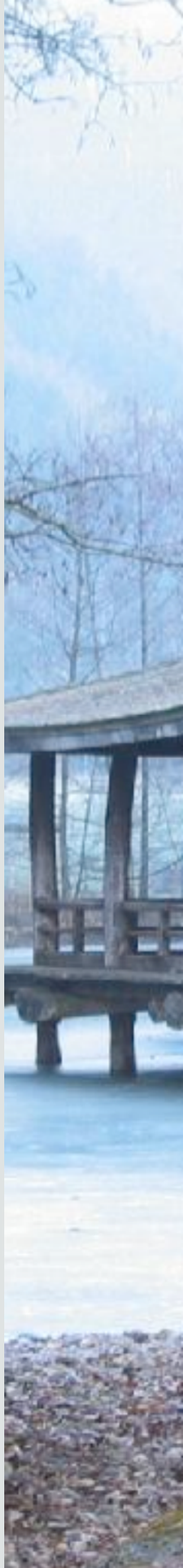
A través de WebDAV – cadaver

```
dav:/> mkdir fotos2
dav:/> cd fotos2
dav:/fotos2/> mput *.jpg
```

La prueba

- http://localhost:8080/fotos1/lovers.jpg

Demo



Crear y editar – 3

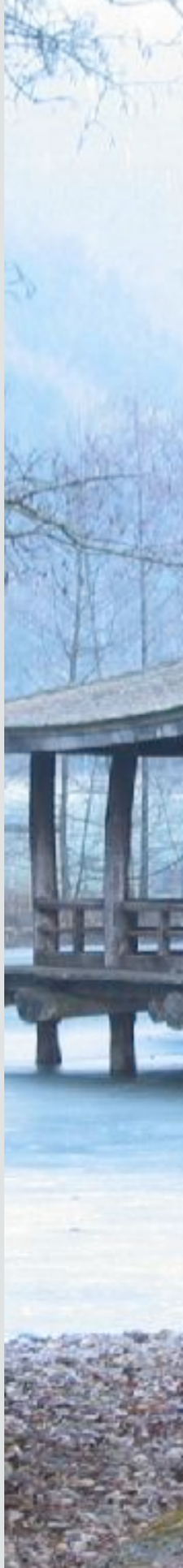
Un poco de HTML

dav:/fotos2/> edit index_html

```
<html>
  <head><title>Carátulas</title></head>
  <body>
    <p><img src="xx.jpg"/></p>
    <p><img src="lovers.jpg"/></p>
    <!-- ... -->
    <p><img src="wideworld.jpg"/></p>
  </body>
</html>
```

- Un poco trabajoso escribir cada nombre de archivo
  - Lo solucionamos luego... 8^)

Demo



ZOPE

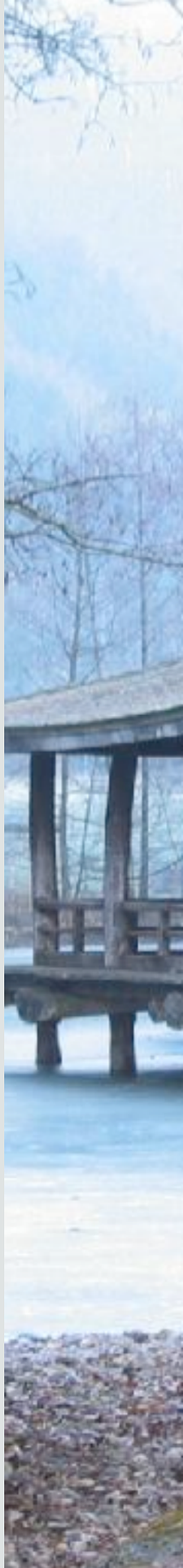
Dinamismo: ZPT – 1

ZPT = Zope Presentation Templates

Creamos «/ver_caratulas» con el siguiente código:

```
<html>
 <head><title>Carátulas</title></head>
 <body>
  <p tal:repeat=”i here/objectValues”>
    <span tal:replace=”i/title_or_id”>
     TITULO_O_ID
    </span>
    <br/><img src=”URL”
      tal:attributes=”src i/absolute_url”/>
  </p>
 </body>
</html>
```

Demo



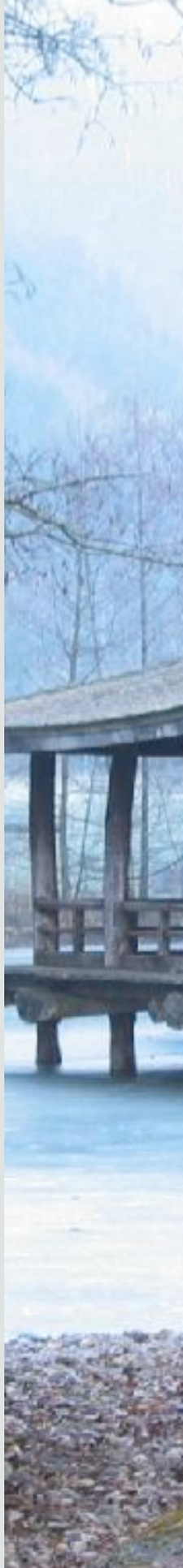
Dinamismo: ZPT – 2

Problema: se recorren todos los objetos

Seleccionar sólo las imágenes: expresión Python

```
<html>
 <head><title>Carátulas</title></head>
 <body>
  <p tal:repeat="i python:
          here.objectValues('Image')">
    <span tal:replace="i/title_or_id">
     TITULO_O_ID
    </span>
    <br/><img src="URL"
      tal:attributes="src i/absolute_url"/>
  </p>
 </body>
</html>
```

Demo



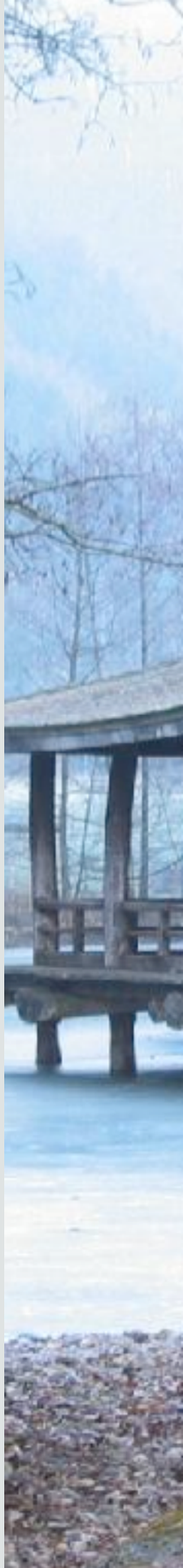
Dinamismo: DTML – 1

DTML – Document Template Markup Lang.

- Permite generar datos no-XML (i.e: CSS, LaTeX, Texto ASCII...)
- Creamos «/caratulas.txt» con el siguiente código:

```
Listado de caratulas
=====================
<dtml-in “here/objectValues”>
 * <dtml-var sequence-item>
</dtml-in>
```

Demo



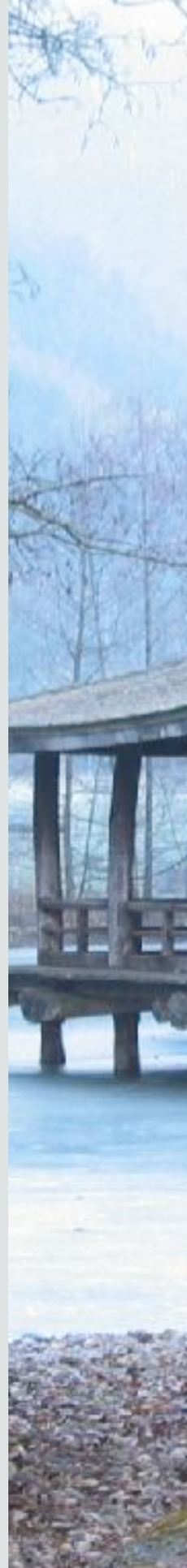
Dinamismo: DTML – 2

Generar TeX con DTML

Creamos «/caratulas.tex» con el siguiente código:

```
\documentclass[11pt]{article}
\title{Listado de Caratulas \\
\begin{small} <dtml-var URL> \end{small}}
\author{Zope}

\begin{document} \maketitle
\begin{itemize}
<dtml-in expr="objectValues('Image')">
    \item <dtml-var title_or_id>
</dtml-in>
\end{itemize}
\end{document}
```

Demo

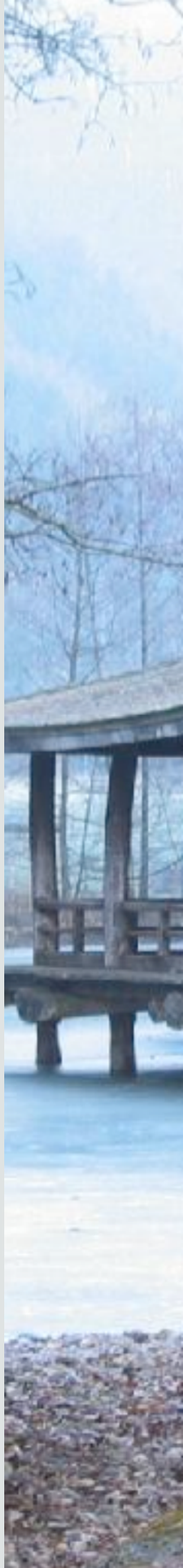


ZOPE

ZPT vs. DTML

Mejor usar ZPT, salvo cuando no aplicable

ZPT	DTML
Siempre genera XML válido	Genera texto de cualquier tipo
Compatible con editores visuales	Necesariamente editado a mano
Promueve modelo MVC	Mezcla lógica con presentación
Plantillas con «slots»	Plantillas anidadas por inclusión
Más fácil de aprender	Permite enviar e-mail

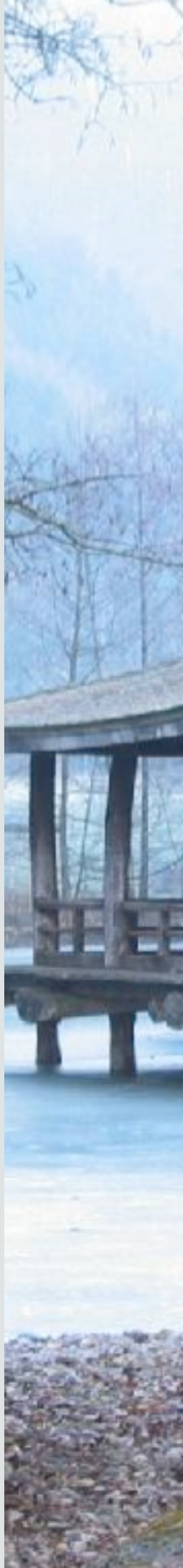


ZOPE

Adquisición – 1

Adquisición: búsqueda de objetos

- Cuando un objeto existe: todo funciona según lo esperado
- No existe: se intenta buscar el objeto
  - Buscando componentes de la URL «hacia arriba»
  - Al encontrar un componente, buscar «hacia abajo»
  - La petición se procesa en el contexto de otro objeto
- Permite:
  - Reutilizar objetos
  - Organizar objetos
  - Especializar objetos

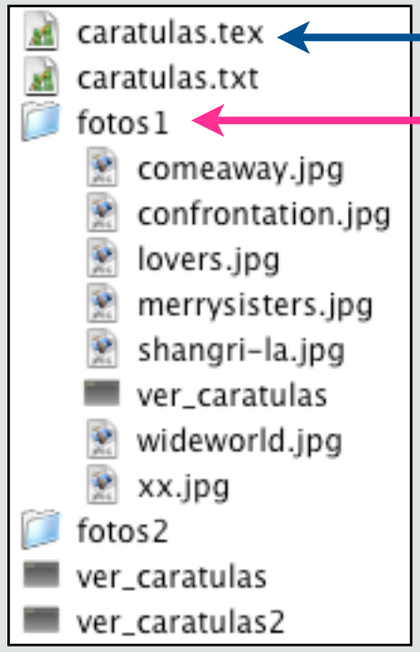


ZOPE

Adquisición – 2

Adquisición: Un ejemplo

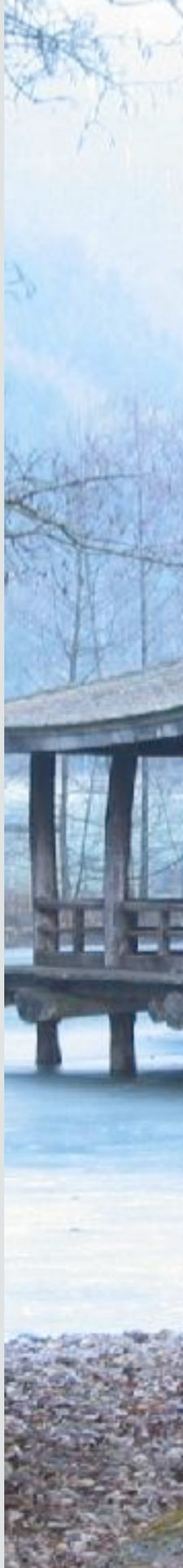
http://localhost:8080/fotos1/caratulas.tex



objeto adquirido

contexto

El código de «caratulas.tex» (adquirido) se interpreta con los datos de «fotos1» (contexto)



ZOPE

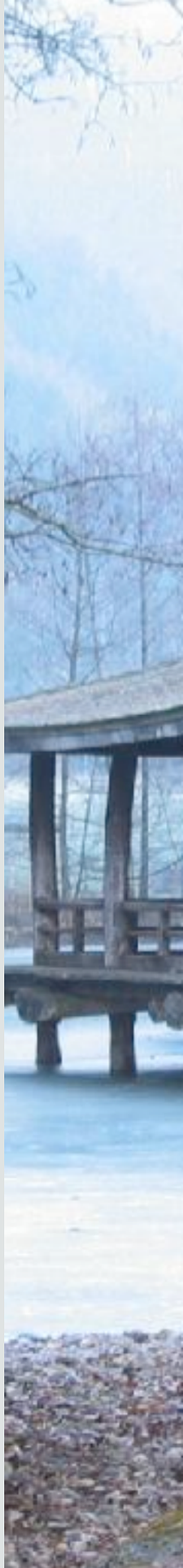
Dinamismo: Python – 1

No podía faltar el «clásico»

Un «hola mundo» en Python con Zope

```
##parameters=name="Zope"

header = """<html><head>
<title>Hello!</title></head><body>"""

footer = """</body></html>"""

print header
print "<p>Hello,", name, "</p>"
print footer

return printed
```

Demo



Dinamismo: Python – 2

«Sandboxing» automático de scripts

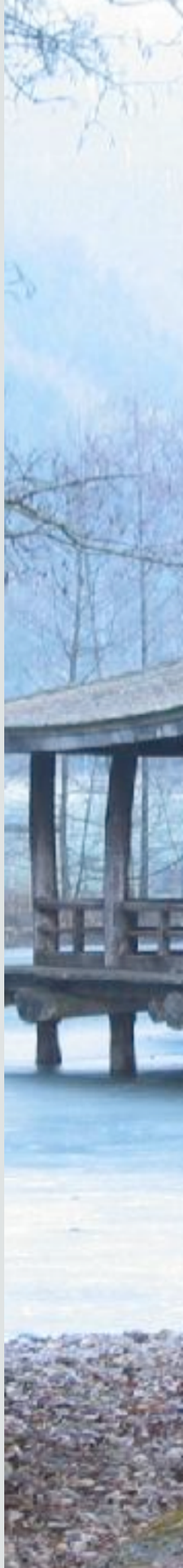
- Limita a los usuarios que pueden crear scripts via web
- Medida de seguridad necesaria y conveniente
- Impide realizar ciertas acciones (i.e: lanzar programas)

Alternativas sin «sandboxing»

- Necesitan acceso al sistema de archivos del servidor
- Métodos externos: controlados, pero con menos restricciones
- Productos: ¡el poder de Greyskull!
- ...para Masters del Universo 8^]

ZOPE

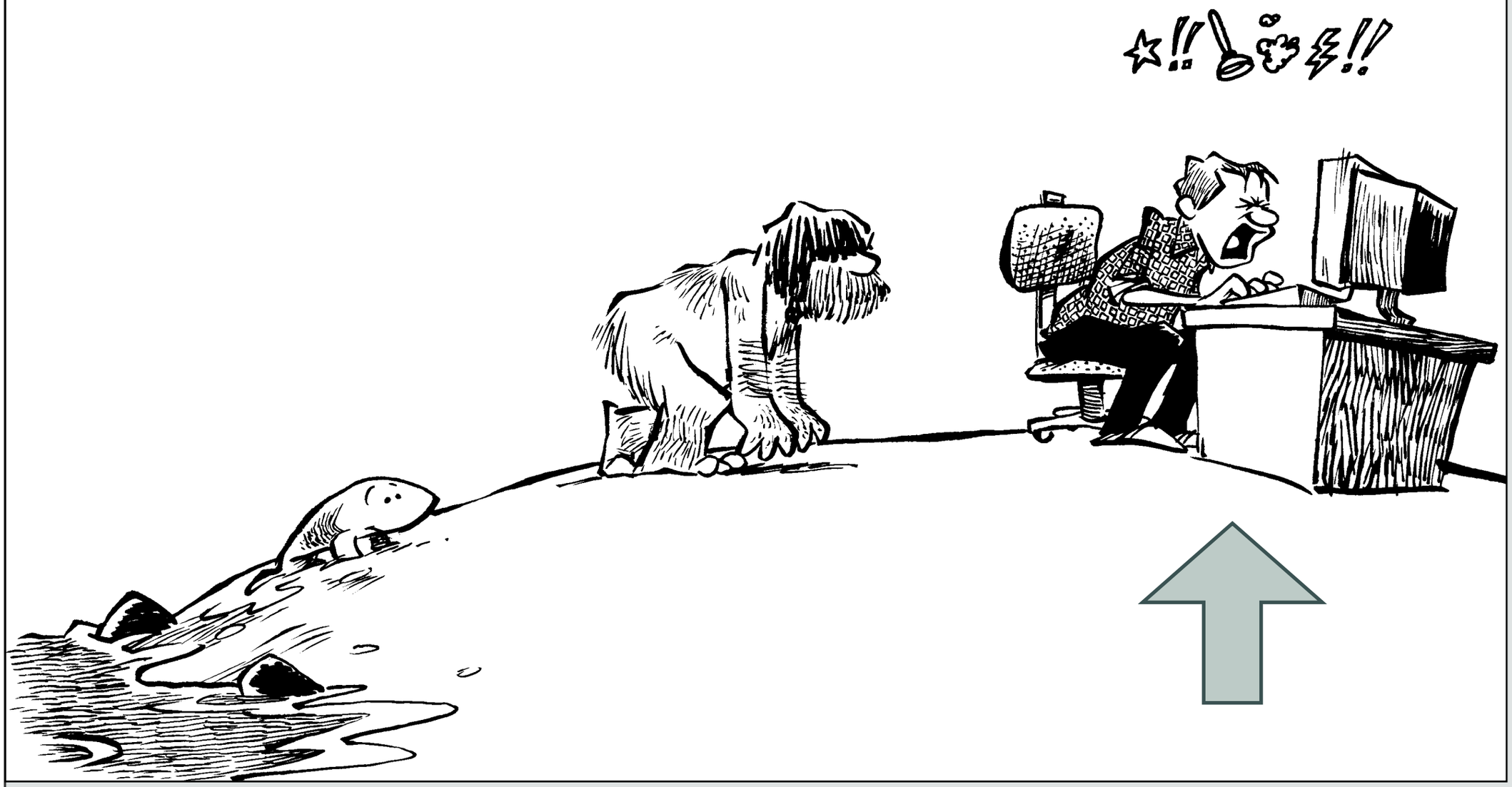
Sistema Operativo Linux
V Jornadas



ZOPE

Sistema Operativo Linux
V Jornadas

El Zen de Zope

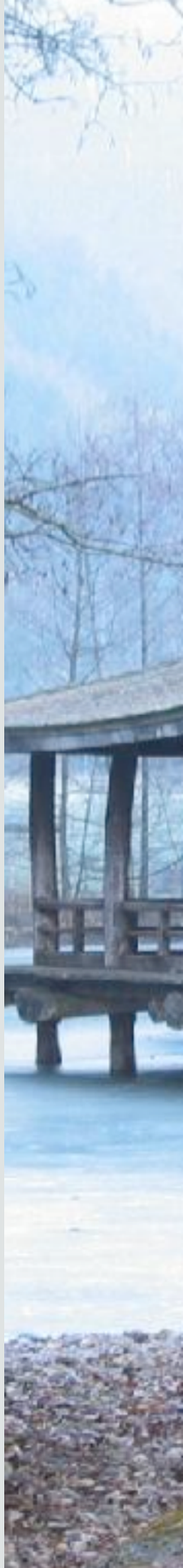


Usuarios, roles, productos

ZOPE

Productos – 1

- Permiten crear nuevos tipos de objetos
  - Los objetos «viven» en la base de datos (ZODB)
  - ZODB «persiste» objetos Python de forma automática
- Ejemplo: Un listín telefónico
  - Una carpeta con instancias de la clase «PhoneListingEntry»
  - «PhoneListingEntry» genérico: no sabe nada acerca de Zope
  - Hipótesis: «PhoneListingEntry» se reutiliza de otra aplicación
  - Realmente se almacenan instancias de «ZPhoneListingEntry»
    - Subclase de «PhoneListingEntry»
    - Adapta «PhoneListingEntry» al API de Zope

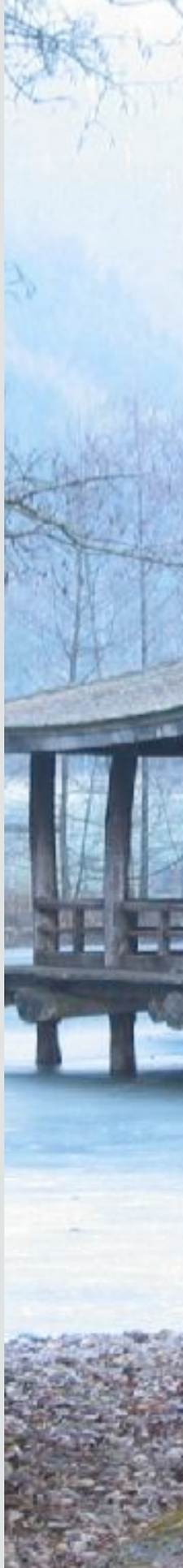


ZOPE

Productos – 2

Código de «PhoneListingEntry» (base.py)

```
class PhoneListingEntry:
  def __init__(self, phone, name):
    self._phone = str(phone)
    self._name = str(phone)

  def phone(self):
    return self._phone

  def name(self):
    return self._name

  def setName(self, newname):
    self._name = str(newname)
```



ZOPE

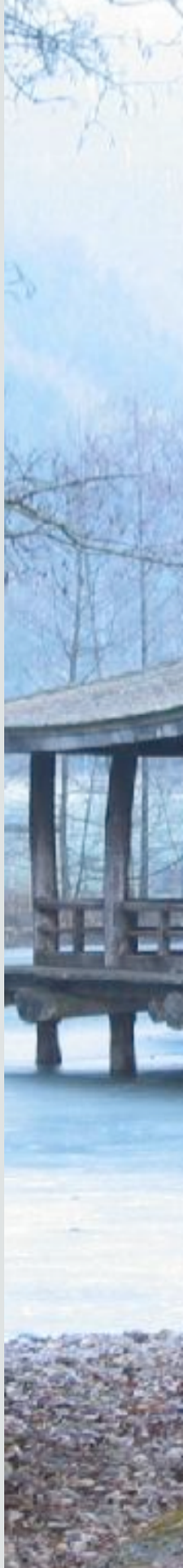
Productos – 3

Importar el API de Zope (adaptor.py)

```
from Globals import InitializeClass
from Globals import DTMLFile
from Persistence import Persistent
from OFS.SimpleItem import Item
```

Definir cómo crear instancias (adaptor.py)

```
manage_addEntryForm =
  DTMLFile('addEntry', globals())

def manage_addEntry(ctx, phone, name,
  REQUEST=None):
    phone = str(phone)
    ctx._setObject(phone,
       ZPhoneListingEntry(phone, str(name)))
    return ctx.manage_main(ctx, REQUEST)
```

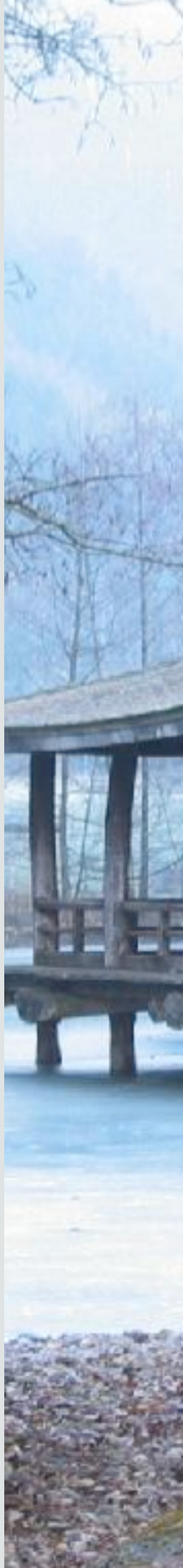


ZOPE

Productos – 4

«ZPhoneListingEntry» (adaptor.py)

```
class ZPhoneListingEntry(PhoneListingEntry,
                                    Item, Persistent):

  meta_type = 'Phone Listing Entry'

  def setName(self, newname):
    PhoneListingEntry.setName(self, newname)
    self._p_changed = 1


InitializeClass(ZPhoneListingEntry)
```



ZOPE

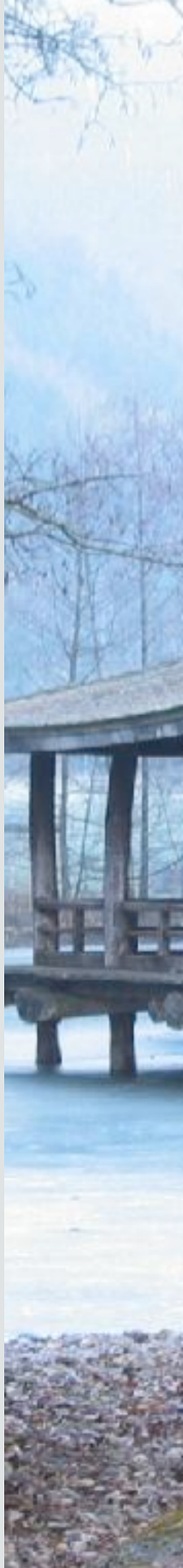
Productos – 5

Inicialización (__init__.py)

```
import adaptor
def initialize(ctx): ctx.registerClass(
  adaptor.ZPhoneListingEntry,
  constructors=(adaptor.manage_addEntryForm,
                adaptor.manage_addEntry))
```

«addEntry.dtml»

```
<form action="manage_addEntry" method="post">
  <p>Telefono: <input type="text"
     name="phone" size="16"/></p>
  <p>Nombre: <input type="text"
     name="name" size="40"/></p>
  <p><input type="submit"
     name="submit" value="Add"/></p>
</form>
```



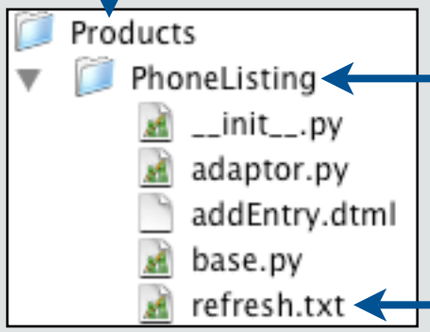
ZOPE

Productos – 6

Instalación del producto

- Subdirectorio dentro «Products» en la instancia de Zope
- Basta con reiniciar Zope tras instalar el producto
- Productos de terceros: descomprimir en «Products»

Dentro del directorio de instancia, o bien dentro del directorio «lib/python» de la instalación de Zope



Importante:
no usar espacios o guiones

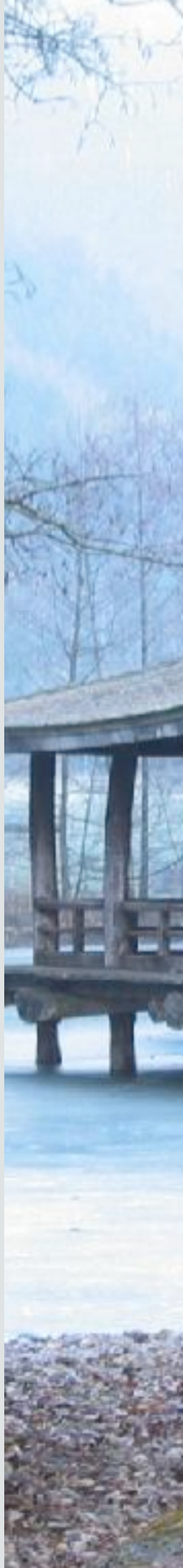
Truco: permite recargar el producto sin necesidad de reiniciar Zope

ZOPE

Usuarios y Roles – 1

Utilidad: definir esquemas de seguridad

- Granularidad fina a nivel de método según:
  - El usuario
  - Roles asignados al usuario
  - Objeto al que se accede
- Los permisos también se propagan por adquisición
- Predefinidos: Anonymous, Authenticated, Manager, Owner
  - http://localhost:8080/manage_access
- El usuario que crea la instancia es administrador
  - Sus roles: Manager + Owner = Control total

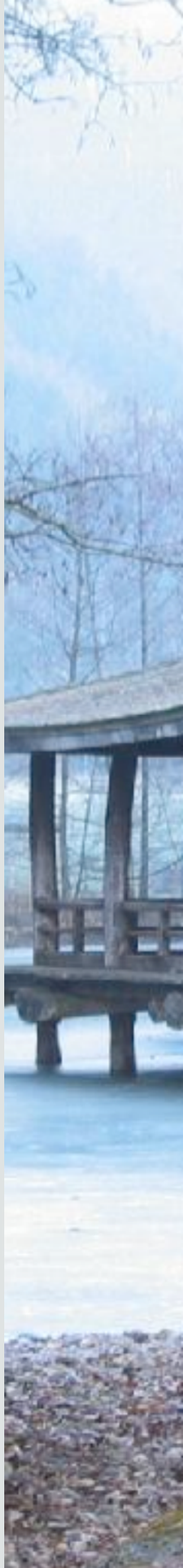


ZOPE

Usuarios y Roles – 2

Utilidad: definir esquemas de seguridad

- Granularidad fina a nivel de método según:
  - El usuario
  - Roles asignados al usuario
  - Objeto al que se accede
- Los permisos también se propagan por adquisición
- Predefinidos: Anonymous, Authenticated, Manager, Owner
  - http://localhost:8080/manage_access
- El usuario que crea la instancia es administrador
  - Sus roles: Manager + Owner = Control total



ZOPE

Sistema Operativo Linux
V Jornadas

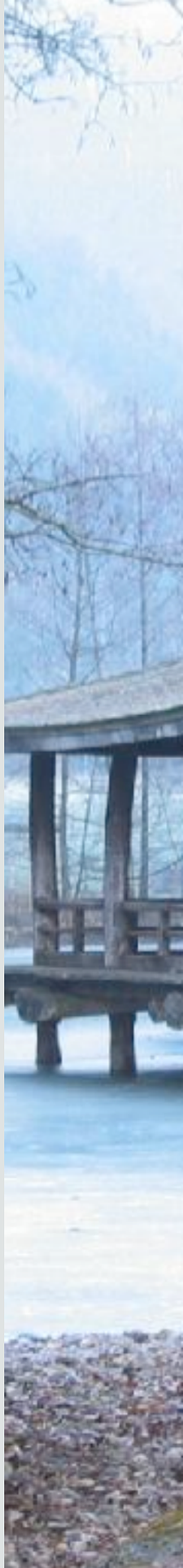
Usuarios y Roles – 3

Importar el API de seguridad

```
from AccessControl import ClassSecurityInfo
```

Declarar permisos en la clase

```
access_method_blah = ‘Access to blah()’

class ZPhoneListingEntry(...):
  security = ClassSecurityInfo()

  security.declareProtected(
      access_method_blah, ‘blah’)
  def blah(self):
    return “you, fool!”
```



Referencias

- Web oficial – http://zope.org
- Plone – http://plone.org
- Recursos adicionales:
  - Productos Zope/Plone - http://www.contentmanagementsoftware.info
  - Noticias e información general - http://planetzope.org

ZOPE

Sistema Operativo Linux

V Jornadas

Que el Zen les acompañe

(gracias por su asistencia)

ZOPE

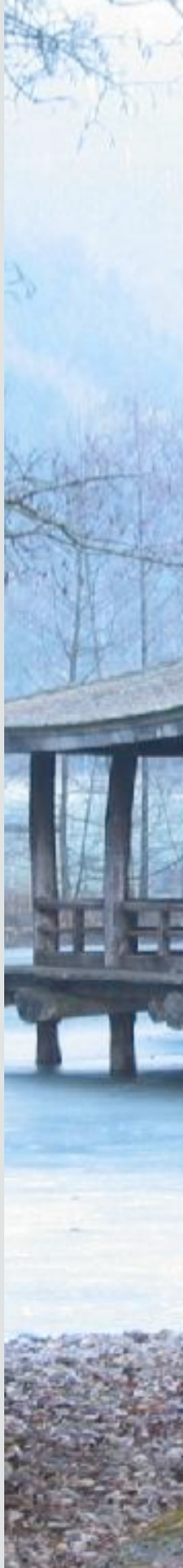
Integración

Autenticación y cuentas de usuario

- LDAP User Folder
- Group User Folder (GRUF)
- ActiveDirectory UserFolder
- Cookie Crumbler (autenticación mediante «cookies»)
- RPC Auth (autenticación para invocaciones XML-RPC)

Otros servicios web

- Interfaces PCGI y FastCGI
- Uso habitual: ZServer+Apache+mod_proxy+mod_rewrite
- Soporte SSL: basta con poner Apache por delante de ZServer

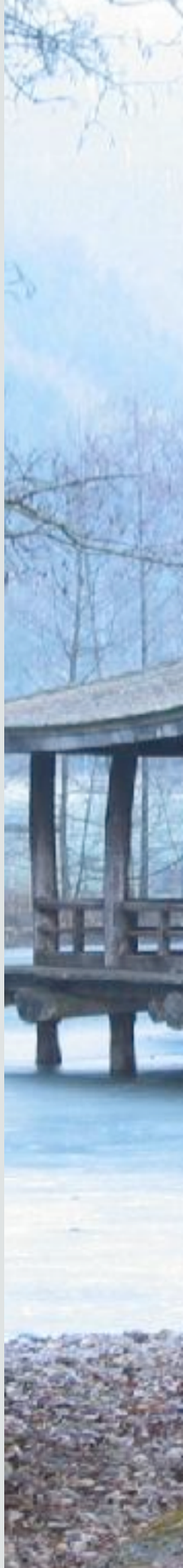


ZOPE

Trabajar cómodamente

Consejito: Navegador + Cliente DAV

- Navegador: basado en Gecko (Firefox) o KHTML (Safari)
- Soporte DAV: activar WebDAV-source
  - Cadaver + Vim (¡mi preferido!)
  - Konqueror, Finder de MacOS X, WebFolders...
- Crear con el navegador (para poder seleccionar tipo)
- Editar mediante DAV (es más cómodo)
- FTP: interesante para subir archivos de forma masiva
  - «PUT_factory» es tu amigo
- En caso de error: «Undo» tiene el poder



ZOPE

Mix 'n Match

Productos ya «prefabricados»

- Contenido: ZWiki, COREBlog, VarImage, Photo, fcForum...
- Utilidades: ExtFile, UserTrack, RPCAuth, TextIndexNG...
- Contenedores: BTreeFolder2, BlogFace...
- Commodities: DocFinderTab, Quickselect, ExternalEditor...

Trabajar a lo grande: Zope+CMF

- CMF = Content Management Framework
- Plone 2, CPS, ZWook, Silva: todos usan CMF
- Integrado de serie en Zope X3