Débuter avec Zope

Présentation
Premières pages, premiers automatismes
Feuilles de styles

Université *Lille3-Charles de Gaulle*
D.Gonzalez – Mars 2003

Table des matières

1 Consultation et téléchargement **5**

2 Trouver de l'information sur le web **7**
2.1 L'aide en ligne de Zope 7
2.2 Initiation 7
2.3 Installation 7
2.4 Général 7
2.5 « Gourous » 8

3 Présentation de Zope **9**
3.1 Présentation générale 9
3.2 Particularités 9
3.3 Le langage de développement 9
3.4 Pourquoi utiliser Zope ? 9

4 Premiers pas avec Zope **11**
4.1 S'identifier 11
4.2 Espace de travail 11

5 Mon premier site **13**
5.1 Une page d'accueil 13
5.1.1 Créer les fichiers 13
5.1.2 Remplir les fichiers 14
5.2 Voir le résultat 15
5.3 Explication des mécanismes mis en œuvre 16
5.3.1 DTML 16
5.3.2 Héritage 16
5.3.3 Comment s'est construite votre page d'accueil ? 16
5.3.4 Détails sur l'en-tête 17
5.3.5 Détails sur le pied de page 17
5.4 Les balises DTML 17

6 Compliquons un peu les choses **19**
6.1 Structure 19
6.2 Galerie d'images 19
6.3 Remplir la galerie 19
6.4 Premier affichage automatique 20
6.4.1 Écrire 20
6.4.2 Comprendre 20
6.5 Améliorons notre galerie 21

7 Des articles **23**
7.1 Créer le répertoire 23
7.2 Des liens vers les articles 23
7.3 Voir directement les articles 24
7.4 Des efforts de présentation 24

8 Soigner la présentation **25**
8.1 Un menu arborescent 25
8.1.1 Créer le menu 25
8.1.2 Quelques modifications 25
8.2 Feuilles de style 26
8.2.1 Création de la feuille de style 26

Zope 1

Consultation et téléchargement

Vous pouvez consulter (et télécharger) ce document aux adresses suivantes :
- version HTML, organisée en pages séparées (une par chapitre), facile à consulter :
  `http ://www.grappa.univ-lille3.fr/~gonzalez/enseignement/commun/zope/index.html`
- version HTML, en une seule grande page, facile à imprimer ou à sauvegarder :
  `http ://www.grappa.univ-lille3.fr/~gonzalez/enseignement/commun/zope/zope.html`
- version Pdf :
  `http ://www.grappa.univ-lille3.fr/~gonzalez/enseignement/commun/zope/zope.pdf`
- version PostScript :
  `http ://www.grappa.univ-lille3.fr/~gonzalez/enseignement/commun/zope/zope.ps`

Zope 2

Trouver de l'information sur le web

2.1 L'aide en ligne de Zope

La première source d'information est **dans** Zope : Zope contient un système d'aide en ligne extrêmement bien fait. Commencez donc par chercher là :
– `http ://www.grappa.univ-lille3.fr :8080/HelpSys`

2.2 Initiation

– **Getting Started With Zope**
`http ://www.zope.org/Members/klm/GettingStarted`
– **Learning Zope**
`http ://www.os3b.org/web/docs/tutor/LearningZope.html`
– **Gotchas for Zope Beginners**
`http ://www.zope.org/Members/jens/docs/newbie_caveats`
– **Foire aux Questions (FAQ) Zope**
`http ://www.nuxeo.com/techno/faq-zope`
– **Présentation de Zope**
`http ://www.pilotsystems.net/content/technologies/intro_zope_html ?printpage=yes`
– **Zope**
`http ://www.math.jussieu.fr/ zoonek/UNIX/34_Zope/zope.html`

2.3 Installation

– **Une complète introduction à Zope en français**
`http ://www.zope.org/Members/ghuo/introduction_zope_fr`
– **ZopeNewbies FAQ**
`http ://weblogs.userland.com/zopeNewbies/ZopeFAQ/`

2.4 Général

– **Zope documentation**, documentation sur le site officiel de Zope :
`http ://www.zope.org/Documentation`
– **The Zope Book**, tout sur Zope
`http ://www.zope.org/Documentation/Books/ZopeBook/current/index_html`

`http ://www.summatech.ru/content/info/ZopeBook`
- **Zopera**, le portail de la communauté Zope francophone.
  `http ://www.zopera.org/`
- **Zope FAQ**
  `http ://zdp.zope.org/projects/zfaq`
- **Zope Tutorial Glossary**
  `http ://www.phoenixnewtimes.com/Control_Panel/Products/ZopeTutorial/Help/tutorialGlossary`
- **Zope solution provider** (hébergeurs avec support Zope)
  `http ://www.zope.org/Resources/ZSP`

2.5 « Gourous »

- **Zope page template**, documentation for french developers
  `http ://www.zope.org/Members/tm/index_docfr_html`
- **Zope Quick Reference**
  `http ://zdp.zope.org/projects/zqr/ZopeQR`

Zope 3

Présentation de Zope

Ce chapitre est en grande partie tiré de « Une complète introduction à Zope en Français » dont vous pourrez trouver l'original à cette adresse :

`http ://www.zope.org/Members/ghuo/introduction_zope_fr`

3.1 Présentation générale

Zope signifie *Z Object Publishing Environment*.
Zope est une plateforme objet de développement de sites Web, développé en Python et open-source Zope permet donc de créer des sites web traditionnels mais aussi des sites web dynamiques, des applications Intranet, en manipulant des objets Zope prédéfinis ou dérivés.

3.2 Particularités

- Zope possède sa propre base de données orientée objet (ZODB) pour stocker les objets Zope. Cela signifie, le site étant créé à l'aide de ces objets, qu'aucun fichier ne figure sur le disque dur du serveur.
- Il intègre une gestion de la sécurité très performante.
- Il est particulièrement adapté pour faire de la maintenance de site :
  - il permet de détacher facilement l'information de sa présentation,
  - via un navigateur web, il est facile d'apporter des modifications aux instances d'objets Zope représentant la (ou les) application(s) web.

3.3 Le langage de développement

Un langage de script, basé sur le langage python, a été créé pour l'écriture de pages dynamiques : le langage **DTML**.

3.4 Pourquoi utiliser Zope ?

Les principales raisons qui pourront vous pousser à utiliser Zope sont (dans le désordre le plus absolu) :
- gratuit et open-source ;
- rapide et efficace (supporte la gestion de gros sites très fréquentés) ;
- facilite la gestion d'un site complexe (l'articulation des pages entre elles se déduit automatiquement de la façon dont vous avez structuré les données enregistrées ; eh oui, ça veut dire qu'il faut penser avant d'agir ; mais ça vous évitera bien du travail) ;

– simple de construire des sites dynamiques (grâce au langage simple DTML) ;
– simplification du traitement des formulaires ;
– interface simple avec la plupart des bases de données (PostgreSQL, MySQL, Oracle, Sybase, etc.).

Zope 4

Premiers pas avec Zope

4.1 S'identifier

Dans votre navigateur web préféré tapez l'adresse
`http ://www.grappa.univ-lille3.fr :8080/zope/`*`identifiant`*`/manage`
(*identifiant* vous a été donné en cours).
La première page qui s'affiche vous demande de vous identifier (voir fig. 4.1).

FIG. 4.1 – Identification

Il vous faut alors taper l'identifiant et le mot de passe qui vous ont été donnés en cours.

4.2 Espace de travail

Vous êtes alors dans votre répertoire personnel (voir fig. 4.2) sur lequel vous avez tous les droits. Vous ne pouvez aller nulle part ailleurs, sauf en tant que simple visiteur.

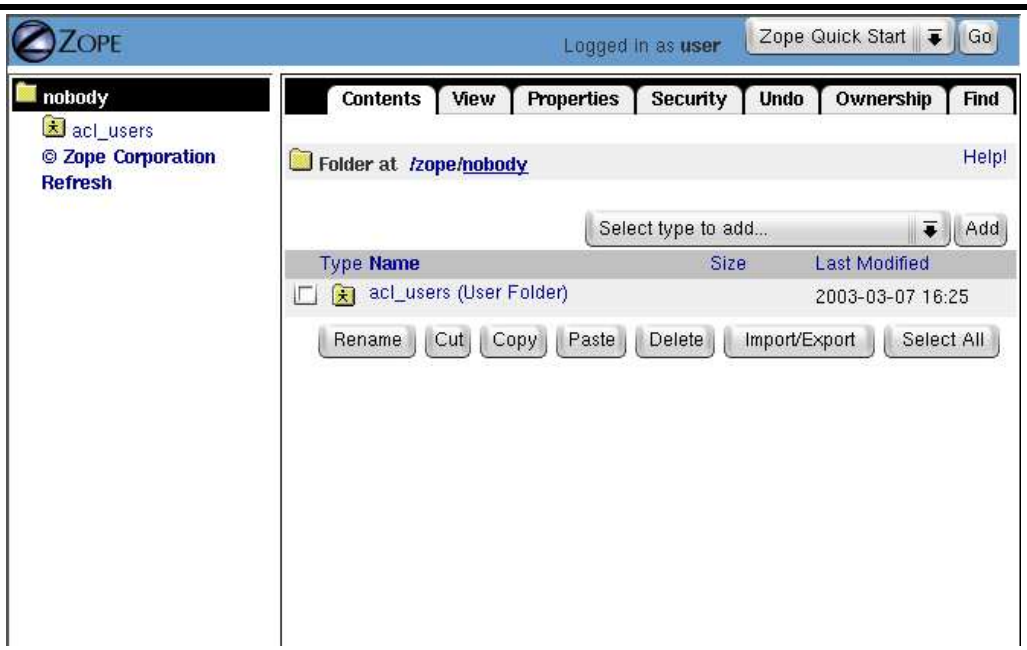

FIG. 4.2 – Votre espace de travail

La première chose à faire est de changer de mot de passe pour en choisir un qui vous soit personnel. Pour cela cliquez sur acl_users (User Folder) puis sur l'utilisateur qui y figure, saisissez votre nouveau mot de passe deux fois, une fois dans « new Password », et une fois dans « (Confirm) », puis cliquez sur Change.
C'est fait votre mot de passe est changé (ce qui fait qy'à votre première action le navigateur va vous redemander de vous identifier).

Zope 5

Mon premier site

5.1 Une page d'accueil

5.1.1 Créer les fichiers

1. Commencer par repérer le bouton

.
C'est celui que vous devrez utiliser chaque fois que vous voudrez créer un objet dans vos répertoires.
2. Utiliser ce bouton pour ajouter une **méthode** DTML (`DTML Method`).
3. Vous obtenez un écran qui ressemble à la figure 5.1.

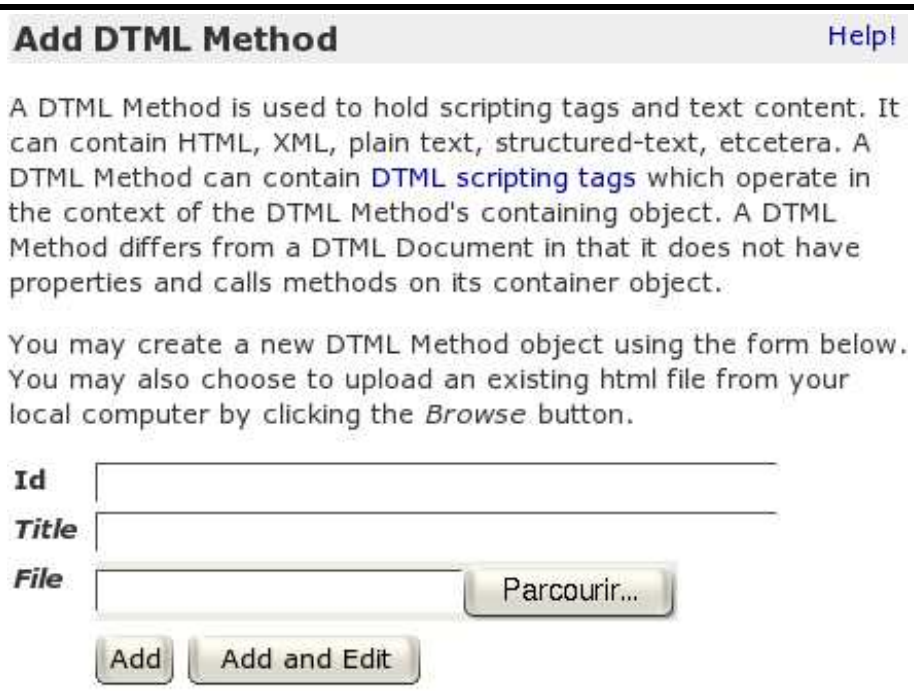

FIG. 5.1 – Add DTML Method

4. Dans le champ `Id` écrivez **exactement** « `standard_html_header` » (sans les guillemets bien entendu !) puis cliquez sur `Add`.

5. Répétez la même chose, mais en insérant une **méthode** DTML (`DTML Method`) que vous appellerez « `standard_html_footer` ».
6. Créer un **document** DTML que vous appellerez « `index_html` ».

> **Vous venez de créer les premières briques de votre première page : `index_html` sera la page d'accueil, tandis que `standard_html_header` et `standard_html_footer` seront les en-têtes et pieds de TOUTES vos pages.**

5.1.2 Remplir les fichiers

Pour l'instant les fichiers ne contiennent que des valeurs par défaut qui ne vont pas produire beaucoup d'effet. Ils risquent même peut-être de ne produire que des messages d'erreurs.
Il est donc temps de les remplir :

1. Cliquer sur `index_html`. Vous obtenez un écran qui vous permet d'en modifier le contenu. Il ressemblera sans doute à la figure 5.2.

FIG. 5.2 – Modifier le contenu de `index_html`

2. Supprimer ce que contient la zone d'édition et remplacez le par :

```
<dtml-var standard_html_header>
Voici la première page de <dtml-var title_or_id>.
<dtml-var standard_html_footer>
```

   Dans le champ `Title` écrivez vos nom et prénom (ou quoi que ce soit d'autre qui vous plairait comme titre !).

   Cliquez ensuite sur « `Save Changes` ».

3. Repérer en haut de l'écran le chemin de votre fichier. Il ressemblera à la figure 5.3, mais à la place de `nobody` il y aura votre nom. Cliquer dessus (à l'endroit de votre nom). Vous vous retrouvez alors dans votre espace de travail.

> **À tout moment figure à cet endroit l'adresse de ce que vous visualisez actuellement. Chaque élément de l'adresse (entre les *slashs* « / ») est indépendant et vous emmène directement à l'endroit ainsi nommé.**

FIG. 5.3 – Chemin de `index_html`

4. Faites la même chose pour `standard_html_header` en remplaçant le texte par :

```
<HTML><HEAD>
<TITLE><dtml-var title_or_id></TITLE>
</HEAD><BODY BGCOLOR="white">
<H1><dtml-var title_or_id></H1>
<HR>
```

   Ne mettez pas de titre.

5. Faites la même chose pour `standard_html_footer` en remplaçant le texte par :

```
<HR>
<P ALIGN="right">
<FONT COLOR="gray" SIZE="-1">
Dernière modification :
<dtml-var bobobase_modification_time>
</FONT>
</P>
</BODY></HTML>
```

   Ne mettez pas de titre pas de titre non plus.

5.2 Voir le résultat

Pour voir résultat de ce que vous venez de faire, il suffit, quand vous êtes dans votre espace de travail de cliquer sur l'onglet

.
Si tout s'est bien passé vous devez obtenir une page qui ressemble à la figure 5.4. Bien entendu il

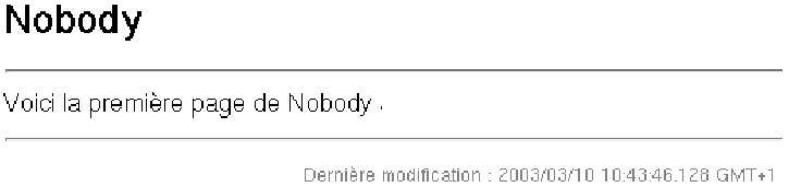

FIG. 5.4 – Votre première page

ne sera pas écrit `Nobody`, mais votre nom, et la date sera, à peu de chose près, l'heure du jour où aurez fait cet essai.

5.3 Explication des mécanismes mis en œuvre

5.3.1 DTML

Les objets que vous avez manipulés sont des objets DTML. Dans l'aide en ligne de Zope on trouve ceci :

> **A DTML Document contains web-editable content. A DTML Document roughly corresponds to a web page. DTML Documents can contain scripting commands in Document Template Markup Language (DTML), which allows for dynamic behavior.**

Ce qu'on peut traduire par :

> **Un document DTML contient de l'information à afficher sur le web. Un document DTML correspond à peu près à une page web. Les documents DTML contiennent en plus des commandes écrites en Document Template Markup Language (DTML, langage à balises pour les modèles de documents), ce qui permet un comportement dynamique.**

Qu'est-ce que le langage DTML ? Il s'agit d'une extension du langage HTML dans lequel on a ajouté quelques balises supplémentaires qui permettent de gérer des contenus dynamiques.
On reconnaît les balises DTML à ce qu'elles commencent par « `dtml-` ». Vous pourrez en trouver une liste exhaustive à la section 5.4.

5.3.2 Héritage

Les objets que vous allez construire vont être construits dans ce qui ressemble à une arborescence de répertoires et fichiers. (Ce n'en est absolument pas une car les objets sont en fait rangés dans la base de données intégrée à Zope. Mais nous ferons semblant d'y croire quand même car cela nous simplifiera beaucoup la vie...)
Quand un de vos objets voudra utiliser une méthode (ou une variable) DTML, il cherchera dans son propre (pseudo-)répertoire. S'il ne la trouve pas, il cherchera dans le (pseudo-)répertoire père. S'il ne la trouve pas, il continuera à remonter dans l'arborescence, jusqu'à trouver la méthode (ou la variable) demandée.
C'est grâce à ce mécanisme que les méthodes `standard_html_header` et `standard_html_footer` que vous avez créées seront utilisées dans toutes vos pages comme en-têtes et pieds de page, même si ces pages se trouvent dans les profondeurs de votre arborescence.

5.3.3 Comment s'est construite votre page d'accueil ?

Votre page commence par :

```
<dtml-var standard_html_header>
```

Elle va donc chercher la variable DTML qui s'appelle `standard_html_header`, c'est à dire l'en-tête par défaut que vous avez défini (on verra plus loin les détails de cet en-tête).
Puis on trouve :

```
Voici la première page de <dtml-var title_or_id>.
```

C'est-à-dire du texte (*Voici la première page de*) et une autre variable (`title_or_id`). Cette variable cherche dans le *répertoire* courant si `Titre` ou `Id` sont définis. Elle trouve le titre que vous aviez défini (votre nom en principe), et c'est ce qu'elle affiche.
La page se termine par :

```
<dtml-var standard_html_footer>
```

Comme pour l'en-tête, c'est le pied de page par défaut que vous avez créé qui s'affiche.

5.3.4 Détails sur l'en-tête

L'entête contient du HTML classique ainsi que la même variable DTML `title_or_id`.
Il va créer l'en-tête de la page web (`<HEAD>`), définir le fond blanc, écrire le titre de la page (`<H1>`) et le faire suivre d'une ligne horizontale sur toute la largeur de la page (`<HR>`).
Si vous n'en précisez pas d'autre dans votre arborescence, c'est celui qui sera utilisé dans **toutes** vos pages. C'est donc une garantie d'homogénéité de la présentation de votre site.
Si vous voulez modifier la présentation globale du site, c'est donc ici qu'il faudra revenir.

5.3.5 Détails sur le pied de page

Le pied de page, lui, à une exception, ne contient que du HTML standard : une ligne horizontale suivie d'un paragraphe aligné à droite, de couleur grise, sur lequel se termine la page.
La seule chose nouvelle est l'utilisation de la variable « `bobobase_modification_time` » : elle indique la date de modification du document depuis lequel elle est appelée.

5.4 Les balises DTML

Voici une liste **exhaustive** des balises DTML :
– `dtml-call` : appeler une méthode DTML.
– `dtml-comment` : insérer un commentaire.
– `dtml-functions` : utiliser une fonction Python prédéfinie.
– `dtml-if` : *SI*, pour tester une condition.
– `dtml-in` : créer des boucles, pour répéter, parcourir, etc.
– `dtml-let` : définir une variable et lui donner une valeur.
– `dtml-mime` : créer un type MIME (principalement utilisé pour le mail).
– `dtml-raise` : lever une exception (pour gérer les erreurs survenues lors de l'exécution).
– `dtml-return` : fin d'une méthode, et éventuellement renvoyer un résultat).
– `dtml-sendmail` : envoyer un mail via SMTP.
– `dtml-sqlgroup` : créer un groupe dans une requête SQL.
– `dtml-sqltest` : insérer un test dans du SQL.
– `dtml-sqlvar` : insérer des variables dans du SQL.
– `dtml-tree` : afficher un arbre dynamique.
– `dtml-try` : permet la gestion des exceptions (voir `dtml-raise`).
– `dtml-unless` : permet de tester facilement des conditions négatives.
– `dtml-var` : insérer une variable (ou plutôt sa valeur).
– `dtml-with` : *pousse* un objet dans l'espace de nommage de DTML ; les variables seront cherchées d'abord dans les objets *poussés*.

Zope 6

Compliquons un peu les choses

Dans ce chapitre (et dans la suite) nous parlerons de fichiers et de répertoires, et nous ferons semblant de croire que tout ce que nous faisons est rangé sous cette forme sur le serveur.

6.1 Structure

La première chose à avoir en tête c'est :

C'est la structure de ce que nous écrirons qui en définira l'apparence.

Cela signifie qu'il ne faudra pas tout écrire à plat, mais d'abord penser à une structure, une hiérarchie, qui permettront d'induire presque automatiquement l'interface de navigation.
Autrement dit :

Il faut penser avant d'agir.

6.2 Galerie d'images

Nous allons créer une galerie d'images dans notre site.
Commencez par aller dans votre espace personnel et ajouter un dossier (*Folder* en anglais).
Donnez lui « `galerie` » comme `Id`, et « Ma galerie à moi » comme `Title`.
Puis cliquez sur « `Add` ».
Entrez dans ce nouveau dossier.

6.3 Remplir la galerie

Assurez vous d'avoir quelques images sur le disque dur de votre machine.
Puis ajoutez une image.
Cela vous amène dans un écran qui doit ressembler à la figure 6.1.
En cliquant sur le bouton « `Parcourir` », choisissez une image de votre disque.
Donnez lui ensuite un nom (« `Id` ») et un titre (« `Title` »).
Quand tout cela est fait, cliquez sur « `Add` ».
Recommencez pour quelques autres photos (au moins 3, mais pas trop, disons pas plus de 6 ou 7).
Attention : les « `Id` » doivent être tous différents.

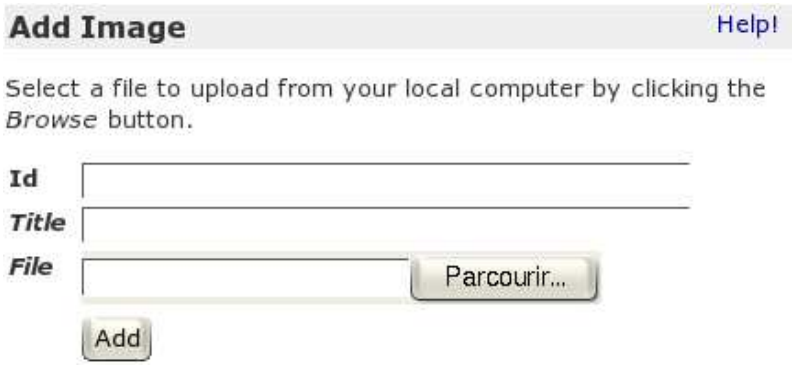

FIG. 6.1 – Ajouter une image

6.4 Premier affichage automatique

6.4.1 Écrire

Ajoutez, en restant toujours dans votre dossier « `galerie` », un document DTML que vous appellerez `index_html`, ouvrez le, et supprimez tout ce qu'il contient pour le remplacer par :

```
<dtml-var standard_html_header>
<dtml-in expr="galerie.objectValues()">
    <a href="<dtml-var absolute_url>"><dtml-var title></a>
    <dtml-var getSize> octets)<BR>
</dtml-in>
<dtml-var standard_html_footer>
```

Respectez les majuscules, surtout sur « `galerie.objectValues()` ».

Sauvegardez et testez cette page (`View`).
Vous devez obtenir une liste de liens qui permettent d'afficher vos photos. Chacun de ces liens est suivi de la taille en octets (*bytes*) du fichier correspondant.
Si vous cliquez sur l'un de ces liens, cela affiche l'image correspondante.

6.4.2 Comprendre

Le fichier `index_html` commence par

```
<dtml-var standard_html_header>
```

et finit par

```
<dtml-var standard_html_footer>
```

comme celui du chapitre précédent.
Le reste est nouveau.

```
<dtml-in expr="galerie.objectValues()">
    <a href="<dtml-var absolute_url>"><dtml-var title></a>
    (<dtml-var getSize> octets)<BR>
</dtml-in>
```

Les première (`<dtml-in ...>`) et quatrième (`</dtml-in>`) lignes permettent de définir une boucle qui va parcourir tous les objets se trouvant dans « `galerie` » (c'est ce que signifie l'expression « `galerie.objectValues()` »).
À chaque *tour* de cette boucle, les appels de variables se feront en référence à l'objet en cours :

– `<dtml-var absolute_url>` désignera l'adresse (ou URL, Uniform Resource Locator) de l'objet ;
– `<dtml-var title>` désignera le tire de l'objet ;
– `<dtml-var getSize>` désignera la taille de l'objet (en octets).
Après ces explications, allez voir le code source la page obtenue pour mieux comprendre.

6.5 Améliorons notre galerie

Vous avez peut-être remarqué que dans notre galerie précédente figurait une taille en octets qui ne correspondait à aucune de nos images.
C'est celle de la page `index` elle-même.
Pour éviter ce problème on peut introduire un test :

```
<dtml-var standard_html_header>
    <dtml-in expr="galerie.objectValues()">
        <dtml-if expr="id()!='index_html'">
            <a href="<dtml-var absolute_url>"><dtml-var title></a>
            (<dtml-var getSize> octets)<BR>
        </dtml-if>
    </dtml-in>
<dtml-var standard_html_footer>
```

Le test `<dtml-if expr="id()!='index_html'">` signifie que ce qui est entre lui et la fin du test (`</dtml-if>`) ne sera affiché que si l'`id` de l'objet en cours n'est pas `index_html`.
Et ça suffit...

Zope 7

Des articles

Nous allons ici utiliser le même genre de méthode pour obtenir un affichage automatique de tous les articles qui seront mis dans un répertoire.

7.1 Créer le répertoire

Placez vous dans votre espace de travail personnel et créez un dossier qui s'appelle « `journal` ». Dans ce dossier ajoutez quelques documents DTML. Vous êtes libres d'en choisir le contenu. Mais ce serait bien d'y mettre un peu de HTML pour en contrôler un peu la présentation.

7.2 Des liens vers les articles

Nous allons faire une première version du journal en utilisant le même principe que pour la galerie d'image.
Allez dans le **répertoire de la galerie**, repérez le document `index_html`, cliquez dans la case situé à sa gauche (pour le sélectionner), puis cliquez sur `Copy`.
Retournez ensuite dans le dossier `journal` où vous cliquez sur `Paste`.
Vous venez de faire un *copier/coller*.
Si vous demandez à visualiser (*View*) le fichier, vous n'obtiendrez que le contenu du dossier `galerie`. C'est normal, souvenez vous que nous y avons écrit en toutes lettres

```
<dtml-in expr="galerie.objectValues()">
```

Il suffit de remplacer cette ligne par

```
<dtml-in expr="PARENTS[0].objectValues()">
```

et ce sera utilisable n'importe où.
Les objets nommés « `PARENTS[`*i*`]` » représentent les parents de l'objet en cours (ici `index_html`) au *i*-ème niveau au dessus de lui.
Visualisez cette première version pour voir...
Si vous avez pris la peine d'encadrer vos articles chacun d'un couple

```
<dtml-var standard_html_header>

<dtml-var standard_html_footer>
```

il suffira de cliquer sur le titre pour voir apparaître votre article. (Si vous n'avez pas mis d'en-tête il y a de fortes chances que votre navigateur ne reconnaisse pas le texte comme une page HTML. Il peut alors arriver n'importe quoi : le navigateur vous propose de sauvegarder le texte, il peut afficher le code source sans l'interpréter, etc.).

7.3 Voir directement les articles

Plutôt que d'obtenir une pages de liens, nous allons essayer d'obtenir une page unique qui regroupe la totalité des articles. (J'espère que vous n'en avez pas mis trop...)
Commencer par recopier (*Copy/Paste*) le dossier `journal`.
L'`id` du nouveau dossier devra être « `nouveau_journal` » (utiliser `Rename`).
Le titre du nouveau dossier devra être « `Le nouveau journal` » (utiliser l'onglet `Properties` **quand vous êtes dans le dossier**).
La première chose à y faire est d'ôter les en-têtes et pieds de page que vous auriez pu mettre dans vos articles. En effet ils n'ont aucune raison d'être si vos articles ne constituent pas des pages par eux même.
Allez modifier le contenu de `index_html` en remplaçant

```
<a href="<dtml-var absolute_url>"><dtml-var title></a>
(<dtml-var getSize> bytes)<BR>
```

par

```
<dtml-var sequence-item><HR>
```

L'effet obtenu est simple : `<dtml-var sequence-item>` est remplacé par l'article lui-même. C'est un peu spartiate dans la présentation mais c'est efficace.

7.4 Des efforts de présentation

Modifions encore `index_html`.
Nous allons à nouveau remplacer

```
<dtml-var sequence-item><HR>
```

par

```
<dtml-if sequence-odd>
     <DIV STYLE="background:#EEEEEE">
<dtml-else>
     <DIV STYLE="background:#BBBBBB">
</dtml-if>
<H2><dtml-var sequence-var-title></H2>
<dtml-var sequence-item>
</DIV>
```

Quelques mots d'explications :
- `<dtml-if sequence-odd> ... <dtml-else> ... </dtml-if>` :
  Ceci permet d'avoir un comportement différent suivant qu'on est dans un élément de numéro impair (*odd* en anglais) ou pair.
  Le test inverse (pair) existe : `<dtml-if sequence-even>`
- `<DIV STYLE="background:#EEEEEE">` et `<DIV STYLE="background:#CCCCCC">` :
  Ben oui ! Faut pas oublier qu'on est toujours en train de parler de pages web, et qu'on peut utiliser les feuilles de style !
  En résumé : un article sur deux aura un fond gris clair, les autres ayant un fond gris plus foncé.
- `<H2><dtml-var sequence-var-title></H2>` :
  Le titre de l'article en cours de traitement (en `<H2>`).
- `<dtml-var sequence-item>` :
  Comme dans la version précédente.
- `</DIV>` :
  Pour fermer le `<DIV>` ouvert 4 ou 6 lignes au-dessus.

Zope 8

Soigner la présentation

8.1 Un menu arborescent

Tenir à jour un menu n'est pas la partie la plus intéressante de la gestion d'un site, mais c'est une des tâches les plus importantes : sans menu correct vous aurez beau remplir votre site de choses intéressantes, personne ne les trouvera.
Mais si votre site est régulièrement mis à jour, le maintien du menu devient une opération fastidieuse.
Heureusement avec Zope, pour peu que vous ayez bien structuré votre site, tout est simple.

8.1.1 Créer le menu

Allez modifier votre fichier `standard_html_header` pour le transformer ainsi (on a ajouté 4 lignes avant `<H1>`) :

```
<HTML><HEAD>
  <TITLE><dtml-var title_or_id></TITLE>
</HEAD><BODY BGCOLOR="white">
  <dtml-tree expr="nobody" branches=objectValues>
     <A HREF="<dtml-var absolute_url>"><dtml-var title_or_id></A>
  </dtml-tree>
  <HR>
  <H1><dtml-var title_or_id></H1>
  <HR>
```

À la place de `nobody` écrivez le nom de votre site.

Visualisez ce que cela donne. Étonnant, non ?

8.1.2 Quelques modifications

1. Le texte `branches=objectValues` n'était pas nécessaire (il serait même plutôt en trop). Il n'était là que pour vous montrer que c'est vraiment une arborescence (ce qui n'aurait pas été visible car notre site est encore trop peu étoffé).
   Comme notre site est bien fait (bien pensé et bien structuré), chaque dossier a une fonctionnalité unique. On peut donc enlever cela et ne laisser que

   ```
   <dtml-tree expr="nobody">
   ```

2. L'entrée `User Folder` ne doit pas figurer dans notre site. Il suffit d'ajouter un test :

```
<HTML><HEAD>
  <TITLE><dtml-var title_or_id></TITLE>
</HEAD><BODY BGCOLOR="white">
  <dtml-tree expr="nobody">
     <dtml-if expr="id!='acl_users'">
        <A HREF="<dtml-var absolute_url>"><dtml-var title_or_id></A>
     </dtml-if>
  </dtml-tree>
  <HR>
  <H1><dtml-var title_or_id></H1>
  <HR>
```

3. Il manque le retour à la page d'accueil dans le menu. Ajoutons les 3 lignes nécessaires :

```
<HTML><HEAD>
  <TITLE><dtml-var title_or_id></TITLE>
</HEAD><BODY BGCOLOR="white">
   <dtml-with nobody>
      <A HREF="<dtml-var absolute_url>">(Racine)</A>
   </dtml-with>
   <dtml-tree expr="nobody">
     <dtml-if expr="id!='acl_users'">
        <A HREF="<dtml-var absolute_url>"><dtml-var title_or_id></A>
     </dtml-if>
  </dtml-tree>
  <HR>
  <H1><dtml-var title_or_id></H1>
  <HR>
```

Comme précédemment, à la place de `nobody` écrivez le nom de votre site.

8.2 Feuilles de style

Nous avons vu lors du chapitre consacré au *Nouveau Journal* que les feuilles de style sont utilisables. Nous ne les avions alors utilisées que sous leur forme la plus faible (attribut `STYLE` dans une balise HTML).
Nous allons voir comment les utiliser aussi dans leur forme la plus classique (fichier externe commun à tout un site).

8.2.1 Création de la feuille de style

Retournez dans votre espace de travail personnel. Ajoutez y un **fichier** (`File`) dont l'`Id` sera `style_css`.
Puis recopiez dans ce fichier le texte suivant (que je vous laisse comrendre seuls !) :

```
BODY {
   margin: 0cm;
   background: white;
   color: black;
}
div.menuflottant {
   float: left;
   margin-left: 0.5em;
   margin-top: 0.5em;
   padding: 0.2em 0.2em 0.2em 0.2em;
```

```
    position: fixed;
    width: 15%;
}
div.page {
    margin-left: 16%;
    margin-right: 2%;
    padding-bottom: 0.5em;
    padding-left: 2em;
    padding-top: 0.5em;
    text-align: justify;
}
H1 {
    background: #EEEEEE;
    color: white;
    text-align: center;
    padding: 5px;
    margin-bottom: 0cm;
}
H2 {
    margin-top: 0cm;
    background: #BBBBBB;
    color: white;
    text-align: center;
    padding: 3px;
    margin-bottom: 0cm;
}
H3 {
    margin-top: 0cm;
    background: #999999;
    color: white;
    text-align: center;
    padding: 1px;
}
```

Il ne reste plus qu'à modifier l'en-tête et le pied de page standard pour que cela soit pris en compte.

– standard_html_header :

```
<HTML><HEAD>
<TITLE><dtml-var title_or_id></TITLE>
<LINK REL="stylesheet" TYPE="text/css" HREF="style_css">
</HEAD><BODY>
<DIV class="menuflottant">
<dtml-with nobody>
        <A HREF="<dtml-var absolute_url>">(Racine)</A>
    </dtml-with>
   <dtml-tree expr="nobody">
       <dtml-if expr="id!='acl_users'">
           <A HREF="<dtml-var absolute_url>"><dtml-var title_or_id></A>
       </dtml-if>
   </dtml-tree>
</DIV>
<DIV CLASS="page">
<H1><dtml-var title_or_id></H1>
```

– standard_html_footer :

```
<HR>
```

```
<P ALIGN="right">
<FONT COLOR="gray" SIZE="-1">
Dernière modification :  <dtml-var bobobase_modification_time>
</FONT>
</P></DIV>
</BODY></HTML>
```

Table des figures

4.1 Identification 11
4.2 Votre espace de travail 12

5.1 Add DTML Method 13
5.2 Modifier le contenu de `index_html` 14
5.3 Chemin de `index_html` 15
5.4 Votre première page 15

6.1 Ajouter une image 20