

Programmation en Collorg

La commande cog

Permet de gérer une application collorg (initialisation, mises à jour, ...).

Pour créer une nouvelle application collorg

cog init -d my_app Initialise une application. Crée un répertoire correspondant au dépôt collorg de l'application.

Dans le dépôt collorg de l'application

cog make Propage les modifications apportées à l'application.

cog struct Affiche la structure de la base de donnée de l'application ou d'une table (-t).

cog graph Affiche le graph de la base de donnée ou d'un schéma de la base de donnée (-s).

API collorg

Le contrôleur

Permet d'initialiser la connexion à la base de donnée. Exemple de script listant les FQTNs de la base:

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
```

```
from collorg.controller.controller import Controller
relation = Controller().model.relation
```

```
types = relation('collorg.core.data_type')
for type_ in types:
    print(type_.fqtn_)
```

Les relations

relation.select() Déclenche l'extraction des données de la base. Une relation est itérable. L'invocation de **select** est implicite dans un contexte de liste (ex : `for elt in relation:`)

relation.insert() Insert la donnée dans la base.

relation.update(nval) Met à jour la donnée dans la base avec **nval**.

relation.delete() Détruit l'ensemble référencé de la base.

relation.exists() retourne **True** si l'ensemble défini contient au moins un élément.

Les opérateurs algébriques sur les relations

Soient **rel_a** et **rel_b** deux relations portant sur un même ensemble.

Python		Set
$rel_a + rel_b$	\iff	$rel_a \cup rel_b$
$rel_a * rel_b$	\iff	$rel_a \cap rel_b$
$rel_a - rel_b$	\iff	$rel_a \setminus rel_b$

Les attributs

Templates collorg

Les balises du langage de templates

>>> Entrée dans du code python.

--- Zone d'affichage et marque d'indentation hors code python. Le code qui suit sera affiché tel quel.

+++ <variable> Zone d'affichage. Le code qui suit (jusqu'à la balise **---** **<variable>** est stocké tel quel (pour affichage) dans la variable **<variable>**

{% code_python %} Code embarqué dans une zone d'affichage.

Exemple de template :

La template qui suit reçoit un argument **names** contenant une liste de noms. Si la liste n'est pas vide, elle l'affiche au format HTML.

```
# >>>
names = kwargs['names']
if len(names):
    # ---
    <ul>
    # >>>
    for name in names:
        # ---
        <li>{% name %}</li>
    # ---
    </ul>
```