G. Hunault
Master 1 Informatique

Angers, mars 2017

Langages de scripts et Frameworks de développement Web

1. Programmation RePePhPyRu

Dans le but de comparer les langages REXX/REGINA, PERL, PYTHON et RUBY on se propose ici d'écrire le même sous-programme dans ces quatre langages. On suppose qu'on dispose au départ d'un fichier-texte nommé par exemple **molq.txt** qui contient une expérimentation moléculaire de chimie quantique par ligne. Voici un extrait d'un tel fichier qui contient en général 3 000 lignes :

```
[...]
   C106H43N5O4S_0_1_1830_20345.7027198155_Gas_OPT.txt
   C109H44N6O4S4_0_1_1949_21630.1508935124_Gas_OPT.txt
   C109H44N6O4S4_0_1_1949_21705.8768459568_Gas_OPT.txt
   C10H10N2_0_1_324_628.3900149426_Gas_V_D2h_tdcam.txt
   C10H10N2_0_1_324_628.3900149428_Gas_onlyS.txt
   C10H10N2_0_1_324_628.3900149428_Gas_only.txt
[...]
```

On admet qu'on dispose d'une chaine de caractères nommée cible que l'on veut chercher dans ce fichier. Ce pourrait être par exemple `C10H10N2` ou `Gas_OPT`. Le sous-programme sera nommé `rchExp` et doit réaliser deux actions : donner le nombre de lignes correspondant à la cible, puis le nombre de molécules associées sachant qu'une molécule est définie comme le premier mot de la ligne (donc la sous-chaine juste avant le premier symbole souligné). Les paramètres de ce sous-programme sont respectivement le nom du fichier et la chaine-cible.

Ainsi, pour l'extrait proposé et avec le nom de fichier fourni, l'appel du sous-programme `rchExp("molq.txt","Gas_only")` doit renvoyer respectivement 2 et 1 car il y a deux lignes qui contiennent la cible, mais une seule molécule (`C10H10N2`) est impliquée.

a) Ecrire une fonction REXX/REGINA qui réalise cette tâche ;

b) Ecrire une fonction PERL qui réalise cette tâche ;

c) Ecrire une fonction PHP qui réalise cette tâche ;

d) Ecrire une fonction PYTHON 3 qui réalise cette tâche ;

   Pourrait-on facilement en faire une méthode pour la classe *String* ?

e) Ecrire une **méthode** RUBY dans la classe *String* qui réalise cette tâche.

Est-ce qu'un autre langage serait plus adapté pour effectuer ces deux comptages ? Si oui, lequel ?

2. Un peu de base de données en Python sans Django

Avec Python il n'y a pas que Django comme *framework*. Ainsi, l'excellent Flask mérite une attention toute particulière. Une fois bien installé et bien configuré, ce *microframework* peut aider à réaliser de nombreux prototypes.

On voudrait, dans le cadre d'une application de relations entre personnes âgées, proposer des services d'aide à la personne. On pourra admettre qu'une personne est définie dans une table Pers principalement par les champs

- id (integer) identifiant d'une personne, champ auto-incrémenté,
- surnom (text) chaine utilisée pour la connexion,
- email (text) chaine utilisée pour l'envoi de mails,
- age (integer) age de la personne exprimée en années.

On admettra également que les services d'aide sont principalement définis dans une table Serv par les champs

- id (integer) identifiant du service, champ auto-incrémenté,
- service (text) nom du service d'aide à la personne proposé,
- minage (text) age minimal à partir duquel on peut proposer le service,
- maxage (text) age maximal jusqu'auquel on peut utiliser le service.

On suppose que l'extension *Flask-SQLAlchemy* a été correctement installée et intégrée au projet. C'est bien sûr un « wrapper » pour SQLAlchemy. On admet également qu'un « engine » SQLITE est fonctionnel, que la base de données a été créée, remplie, interfacée... et que toutes les commandes d'administration comme

```
./db_create.py
./db_migrate.py
```

ont été exécutées avec succés. Ces commandes sont évidemment les équivalents des commandes

```
python manage.py migrate
python manage.py makemigration
python manage.py sqlmigrate
```

que vous avez utilisé dans le cadre de votre projet en CC.

On voudrait maintenant tester en shell (comme avec `python manage.py shell`) du code Python pour tester nos idées sur la suite à donner à l'application. Ces commandes sont donc celles qui seraient exécutées juste après le prompt `>>>>`. Indiquer quel code Python permettrait :

2.1 de compter le nombre de services disponibles dans la base,

2.2 de compter le nombre de personnes d'au moins 50 ans dans la base,

2.3 de lister les services dont le nom contient le mot "Déplacement" (initiale majuscule ou non) avec le nombre de personnes susceptibles d'être intéressées par ce service si le critère d'age est vérifié. On admettra que le service est proposable à une personne si son age est compris entre l'age minimal et l'age maximal définis pour le service, bornes incluses.

3. Vive la ligne de commande !

Pour comprendre et dialoguer en HTTP, rien ne vaut la ligne de commande et le fameux utilitaire httpie dont la documentation est disponible à l'adresse `https://httpie.org/doc`.

Il n'est pas nécessaire que httpie soit installé pour pouvoir répondre à ces questions.

Afin de montrer que vous savez lire une documentation et après avoir lu la section `https://httpie.org/doc#examples`, indiquez en ligne de commande

3.1 comment afficher toute le contenu de la page `ndjpm.php` utilisée dans mes tuteurs Python et Ruby pour le paramètre m avec la valeur 5 ;

3.2 comment compléter cette commande en « pipe » avec le symbole | pour n'afficher que la ligne avec le mot "comporte" ;

3.3 en admettant que cette ligne est mise dans le champ *Your test string* du site Rubular, quelle expression régulière écrite dans le champ *Your regular expression* permet d'obtenir dans la zone *Match groups* juste les deux nombres entiers de la phrase, à savoir ici 5 et 31 ?

4. Discussion sur l'ORM

Le MRO (Mapping Relationnel Objet), nommé ORM (Object Relational Mapping) en anglais est-il un concept ou un paradigme ?

Vous essaierez de construire une réponse structurée et bien rédigée à cette questions, si possible à l'aide d'exemples concrets centrés sur Ruby et Python. Il est conseillé d'utiliser au moins trois mots de trois syllabes ou plus pour « transmettre un contenu rédactionnel fort ».

Une dizaine de lignes parait être une rédaction minimale.