Traitement de documents HTML - TD 7

DESS TEXTE

Exercice 1 : étude la classe `HTML : :LinkExtor` (suite)

Dans cet exercice, nous allons voir comment extraire les différents types de liens d'un document HTML.

a) Ecrire un programme qui analyse le document HTML passé en argument et qui affiche l'ensemble des liens sous leur forme absolue.

b) Ecrire un programme qui analyse le document HTML passé en argument et qui affiche l'ensemble des liens de type image (couples attribut-valeur apparaissant derrière la balise *img*).

c) Ecrire un programme qui analyse le document HTML passé en argument et qui imprime dans un fichier de sortie "images.txt" l'ensemble des codes URL correspondant à la balise *img* sous leur forme absolue.

d) Ecrire un programme qui analyse le document HTML passé en argument et qui affiche l'ensemble des liens hypertextes ainsi que le nombre d'apparitions pour chaque balise.

e) Ecrire un programme qui analyse le document HTML passé en argument et qui affiche l'ensemble des liens de type html sous leur forme absolue.

Exercice 2 : étude de la classe `HTML : :TokeParser`

Dans cet exercice, nous allons apprendre à extraire d'un document HTML l'ensemble des attributs d'une balise donnée.

a) Quelles sont les méthodes de la classe `HTML : :TokeParser`, que prennent-elles en argument et que renvoient-elles ?

b) Ecrire un programme qui analyse un document HTML passé en argument et qui retourne le titre du document, ainsi que l'ensemble des liens (balises <a href>) associés au texte servant de pointeur.

c) Ecrire un programme qui analyse un document HTML passé en argument au moyen de la méthode `get_token` et extraire le texte avec `get_trimmed_text`.

d) Ecrire un programme qui analyse un document HTML passé en argument et qui affiche comme résultat la liste des types rencontrés dans le format :

```
["S", $tag, $attr, $attrseq, $text]
["E", $tag, $text]
```

```
(...)
```

e) Ecrire un programme qui analyse un document HTML passé en argument et qui affiche l'ensemble des textes qui sont codés en italique (balise et).

Exercice 3 : étude de la classe `URI : :URL`

Dans cette section, nous allons étudier les différents composants d'une URL, spécialement lorsque nous interrogeons un moteur de recherche.

a) Quelles sont les méthodes de la classe `URI : :URL`, que prennent-elles en argument et que renvoient-elles ?

b) Via votre navigateur web, effectuez une requête sur un moteur de recherche, recopiez alors la requête telle que traduite par votre navigateur (champ "adresse"), et analysez la au moyen des méthodes de la classe `URI : :URL`.