Validation et construction du corpus depuis des URL

Mot d'introduction

Le but de cette feuille est d'obtenir un script capable de générer les premiers tableaux de données concernant vos URL. Ce script va utiliser différentes commandes que nous avons pu voir jusqu'à présent, pensez à bien regarder les slides (et la doc !) en cas de soucis. Nous allons procéder ici en plusieurs étapes :

1. nous travaillerons d'abord sur un unique fichier d'URL
2. nous récupérerons d'abord les informations intéressantes sur les URL
3. nous construirons ensuite un tableau HTML avec des informations minimales
4. nous ferons alors d'autres traitements qui nous permettront d'enrichir le tableau
5. nous étendrons le script à un ensemble de fichiers URL.

Nous avons pour cible d'avoir les données disponibles actuellement sur : `https://yoanndupont.github.io/PPE/tableaux.html`

Note 1 on peut stocker le résultat d'une commande dans une variable.
Exemple : `n_lines=$(wc -l fichier.txt)`

Note 2 on peut très bien stocker le résultat de chaînes de commandes.
Exemple : `n_loc=$(grep Location fichier.ann | wc -l)`

Note 3 lorsqu'on est encore à la recherche du résultat, ne pas hésiter à simplement afficher l'état du contenu au fur et à mesure du traitement. Vous pourrez toujours commenter ces lignes plus tard.

Exercice 1 Télécharger le script de base

Télécharger le script `traitement_url_base.sh` et lancez-le sur un de vos fichiers d'URL afin de vous assurer qu'il fonctionne bien.

Exercice 2 Récupérer des informations de la requête

Dans le cours précédent, nous vous avons fait tester si une URL démarrait pas http pour savoir si une URL était *a priori* valide. Nous allons tester ici une méthode plus directe.

Nous pouvons tester la validité d'une URL en requêtant directement le serveur :

1. utiliser `curl` pour récupérer uniquement l'entête de la réponse du serveur. Quelle(s) option(s) utiliser pour avoir uniquement l'entête de la réponse ? -I
2. Où se trouve le code de réponse ? Comment peut-on le récupérer ? 200
   curl -I https://melimeloculinaire.wordpress.com/ | head -1
3. Mettre le code dans une variable
   n_code=$(curl -I https://melimeloculinaire.wordpress.com/ | head -1)

À propos de l'entête l'entête sera utilisé pour récupérer plusieurs informations, pensez à le stocker dans une variable, par exemple `header`.

Indice idéalement, pour éviter de polluer les résultats avec des informations qui ne nous intéressent pas, on peut passer `curl` en mode silencieux.

Attention si vous suivez les redirections, il peut y avoir plusieurs codes, pensez bien à prendre le dernier.

Exercice 3 Écrire une première version du tableau

Écrivez une page HTML qui contiendra un tableau à trois colonnes, ordonnées dans l'ordre suivant :
— le numéro de ligne dans le fichier
— le code de retour
— l'URL

Un exemple (formatté) de sortie de fichier HTML de tableau à ce niveau :

```
<html>
  <header>
    <meta  charset="UTF-8" />
  </header>
  <body>
    <table>
      <tr><th>ligne</th><th>code</th><th>URL</th></tr>
      <tr><td>1</td><td>200</td><td>www.perdu.com</td></tr>
    </table>
  </body>
</html>
```

Une fois ce tableau obtenu :
- écrivez-le dans un fichier `tableau.html`
- Créer un lien vers `tableau.html` dans `index.html`
- poussez les deux fichiers HTML et observez que le tout s'affiche bien sur votre site github.

Exercice 4 Récupérer l'encodage de la page

La requête `curl` que vous avez faite permet également de récupérer l'encodage de la page :

1. où trouver cette information ? content-type: text/html; charset=UTF-8
2. quelles commandes enchaîner ? curl -I https://melimeloculinaire.wordpress.com/ | sed -n '4,4p'
3. Parfois, aucun encodage n'est renvoyé : dans ce cas, on fera la supposition que la page est en UTF-8.

Une fois l'encodage récupéré, ajoutez-le à la fin du tableau. Votre tableau doit donc avoir quatre colonnes.

Exercice 5 Récupérer l'encodage de la page

Lynx vous permet de récupérer le contenu textuel d'une page :

1. Quelles options faut-il utiliser pour récupérer le contenu textuel et ne pas avoir la liste des liens en fin de page ? lynx -dump -nolist http://www.kozlika.org/kozeries/post/2006/01/13/416-meli-melo
2. Quel traitement effectuer sur votre contenu si votre site n'est pas encodé UTF-8 ?

Pensez à bien stocker ce résultat dans une variable. lynx -dump --display_charset=utf-8 http://www.kozlika.org/kozeries/post/2006/01/13/416-meli-melo

Exercice 6 Compter les occurrences du mot cible

Comptez à présent le nombre d'occurrences du mot cible dans le contenu textuel récupéré. Une fois ce compte récupéré, ajoutez-le à la fin du tableau. Votre tableau doit donc avoir cinq colonnes.

Exercice 7 Généraliser à un ensemble de fichier URL

Généralisez votre script afin qu'il fonctionne sur un dossier contenant une liste de fichiers d'URL.

Exercice 8 Commencez à rendre votre site tout beau

Une fois ce résultat obtenu, vous pouvez regarder comment modifier son CSS afin d'ajouter de la couleur, de l'agencement, etc.