Programmation et projet encadré - L7TI005

mini-projet : préparation au travail en groupe

Yoann Dupont yoann.dupont@sorbonne-nouvelle.fr
Pierre Magistry pierre.magistry@inalco.fr

2024-2025

Université Sorbonne-Nouvelle
INALCO
Université Paris-Nanterre

Les enjeux de cette portion de cours :

- débuter l'écriture de la chaîne de traitement
  - étape 1 : faire des aspirations de pages web à partir d'une liste d'URL et en extraire le contenu textuel
- écrire une sortie en tableau
  - d'abord au format "texte" tabulaire
  - puis au format "web" HTML

Début du mini-projet : commencer la récolte sur des données préparées

Mini-projet

Le mini-projet va permettre de voir (au moins en partie) les différentes étapes du projet de groupe, mais individuellement.

Cela a plusieurs intérêts :

- tout le monde fait le travail essentiel à chaque projet
- permet de vous lancer en douceur sur le travail en groupe

Mini-projet

Le mini-projet va permettre de voir (au moins en partie) les différentes étapes du projet de groupe, mais individuellement.

Cela a plusieurs intérêts :

- tout le monde fait le travail essentiel à chaque projet
- permet de vous lancer en douceur sur le travail en groupe

On démarrera d'une liste d'URL déjà faite, à récupérer sur le dépôt git / icampus du cours.

L'allure générale du travail de récolte

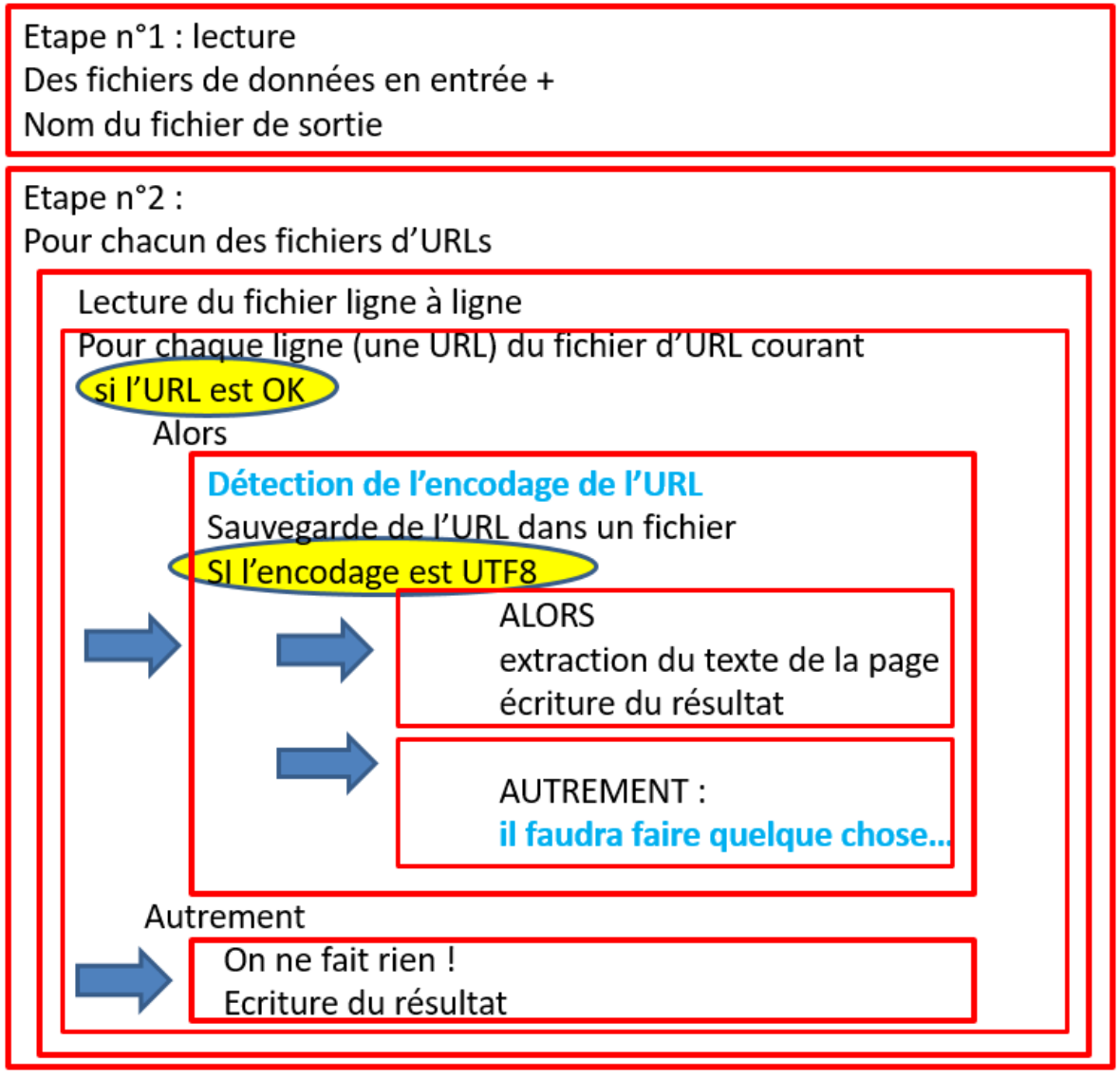

Figure 1: crédits : Serge Fleury

Construction du script

Votre script devra faire plusieurs choses :

- récupérer les URL contenues dans un fichier texte
- écrire sur le terminal des informations séparées par des tabulations
    - on transformera le tout plus tard en page web

On a donc un argument : le fichier d'URL.

Travail à faire

Exercice 1 : lire les lignes d'un fichier en bash

Dans le dépôt vous trouverez le dossier "mini-projet" avec le code suivant :

```
while read -r line;
do
    echo ${line};
done < "urls/fr.txt";
```

Questions :

1. Pourquoi ne pas utiliser cat ?
2. Comment transformer "urls/fr.txt" en paramètre du script ?
   2.1 Valider l'argument : ajouter le code nécessaire pour s'assurer qu'on donne bien un argument au script, sinon on s'arrête
3. Comment afficher le numéro de ligne avant chaque URL (sur la même ligne) ?
   - Bien séparer les valeurs par des tabulations

Exercice 2 : récupérer les métadonnées de collecte

Après l'exercice 1 fait, on va rajouter des informations à chaque ligne, toujours séparées par des tabulations :

1. le code HTTP de réponse à la requête
   1.1 les erreurs peuvent être corrigées
2. l'encodage de la page, s'il est présent
3. le nombre de mots dans la page

Structure attendue pour le mini-projet

En supposant que votre dépôt s'appelle **PPE1-2024**, l'arborescence de vos fichiers devra ressembler à cela une fois tous les exercices de cette semaine finis :

```
PPE1-2024
├── journal.md
├── exercices
│   └── ...
└── miniprojet
    ├── programmes
    │   └── miniprojet.sh
    ├── urls
    │   └── fr.txt
    └── tableaux
        └── tableau-fr.tsv (cette semaine uniquement)
```

Consignes de rendu

- faire les exercices du miniprojet sur son dépôt individuel ;
- créer le tag **miniprojet-1** à la fin du travail et le push sur Github ;
- créer un fichier texte avec le lien github vers le tag **miniprojet-1** ;
- le déposer jusqu'au lundi 11/11 à 23h59 (icampus/mail) ;
- réfléchir au mot à étudier et nous contacter si vous avez des idées.