L'objectif de ce TP est de prendre en main la bibliothèque `BeatifulSoup`, qui permet de faire du web-scraping. La documentation et les instructions d'installation de cette bibliothèque se trouvent à l'adresse suivante.

1 Travail sur un fichier local

Dans un premier temps, on va travailler sur le fichier `King's_Landing.html`, à télécharger sur Eprel.

Ajoutez le code suivant en début de fichier :

```
from bs4 import BeautifulSoup

kl = open("King's_Landing.html",'r')
klsoup = BeautifulSoup(kl, "html.parser")
```

Ainsi, l'objet de type soup décrivant le fichier HTML sera stocké dans la variable `klsoup`.

Question 1 Écrire une fonction `titre(soup)` qui renvoie le titre de la page représentée par `soup`. On fera bien attention à ne pas confondre *titre* et *h1*...
Résultat attendu sur l'exemple :

```
King's Landing | A Song of Ice and Fire
```

Question 2 Écrire une fonction `afficher_h2(soup)` qui affiche le nom de tous les *h2* de la page représentée par `soup`, à raison de un par ligne.
Résultat attendu sur l'exemple :

```
Layout
Population
Military Forces
History
```

Question 3 Écrire une fonction `nb_par_avec_lien(soup)` qui renvoie le nombre de paragraphes (i.e. de balises *p*) contenant au moins un lien.
Résultat attendu sur l'exemple :

```
10
```

2 Travail sur un fichier distant

On va maintenant travailler sur une page html qu'on téléchargera sur un serveur distant.

On commencera par se rendre sur la page `https://iceandfire.fandom.com/wiki/Petyr_Baelish`. L'objectif de cette partie est de lister les pages du wiki vers lesquelles renvoie la page de `Petyr Baelish`. On est bien évidemment intéressé uniquement par les liens situés dans le corps de la page (c'est-à-dire, ceux en rapport avec le personnage de Petyr Baelish), et pas ceux situés à la périphérie (qui sont des liens globaux, des publicités, etc.)

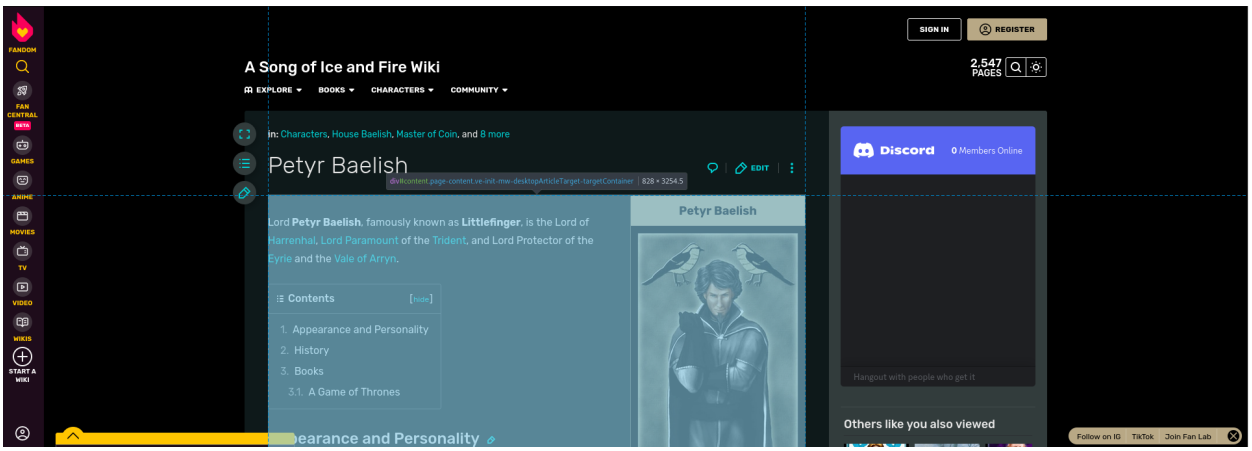

FIGURE 1 – En bleu, le contenu propre à la page de Petyr Baelish.

Question 4 En étudiant la structure html de la page `Petyr Baelish` du wiki via l'inspecteur de son navigateur favori, déterminer quelle balise html correspond au contenu de la page.

Le contenu de la page est représenté en bleu dans la Figure 1.

Question 5 Écrire une fonction `liens()` qui attend en argument l'adresse d'une page web `page` du wiki `https://iceandfire.fandom.com/wiki/`, et qui renvoie l'ensemble (sous forme de `set`) des pages du wiki vers lesquelles pointe `page`. On ne prendra évidemment en compte que les liens contenus dans le corps de `page`, comme identifié à la question précédente.

Pour récupérer le fichier html à l'adresse `adresse`, on fera appel à `requests.get(adresse).text`, après avoir installé et importé la bibliothèque `requests`.

On testera cette question sur la page de Petyr Baelish.

Question 6 Écrire une fonction `liens_distance()` qui attend en paramètre l'adresse d'une page `page` du wiki ainsi qu'un entier `d`. Cette focntion renverra l'ensemble des pages du wiki situés à distance au plus `d` de `page` dans le wiki, où la distance est calculée comme le plus petit nombre de liens à suivre pour passer de la première page à la seconde.

En particulier, `liens_distance(page,1)` renverra le même ensemble que `liens(page)`.