

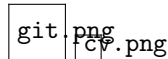
TP4

Ibrahim ALAME

24/10/2022

1 HTML-CSS

Nous commençons par afficher dans un navigateur la page web <https://github.com/ialame/htmlcssCV>. Il s'agit d'un dépôt Git d'un CV au format HTML-CSS.

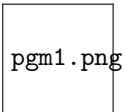


1. Télécharger le dossier **htmlcssCV** sous format zippé en cliquant sur **code** puis **Download ZIP**.
2. Modifier les données personnelles dans le fichier **cv.html** pour que cette page devient votre propre CV.

2 Le parseur HTML : BeautifulSoup

BeautifulSoup est une bibliothèque Python permettant de parser du HTML de manière très simple et de façon tolérante aux erreurs.

Tester BeautifulSoup avec l'exemple suivant :



On construit l'objet HTML parsé **soup** à partir d'un filehandle **html_doc** par l'instruction :

```
soup = BeautifulSoup(html_doc, "html.parser")
```

où **"html.parser"** est une option indiquant au constructeur BeautifulSoup le format HTML et **select("selecteur css")** est une méthode de la classe BeautifulSoup qui retourne une liste d'objet balise correspondant au sélecteur css spécifié.

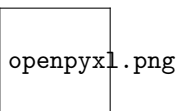
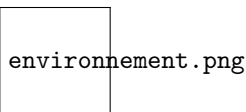
Sur la page : <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> on trouve les méthodes de la classe BeautifulSoup les plus utiles.

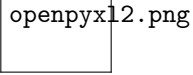
A l'aide d'un programme python :

1. Lister les catégories du cv (les balises **dt**).
2. Lister les sous titres H2 du cv.
3. Extraire du cv les activités de sa propriétaire à l'association "La Coco".
4. Récupérer les coordonnées de Claire DELALUNE : Le nom, l'adresse, le téléphone, le mail et les langues vivantes.

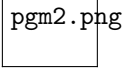
3 Manipulation d'Excel en Python : openpyxl

Installer, si nécessaire, openpyxl avec les commandes suivantes :



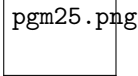


1. Ouvrir le fichier `cartes.xlsx` et afficher sur la console son contenu à l'aide du code suivant :



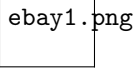
La méthode `load_workbook` de la classe `openpyxl` permet d'ouvrir le fichier excel `cartes.xlsx`. Le `workbook` obtenu contient ses feuilles de calcul dans le tableau `ws=wb.sheetnames`. On extrait la première feuille de calcul en faisant `ws = wb[sheets[0]]`. La méthode `iter_rows` retourne un itérateur que l'on peut utiliser dans une boucle `for` pour afficher les éléments d'une ligne courante.

2. Exécuter le programme suivant :

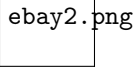


4 Importation des données à partir d'un site web

On commence par afficher le site web : <https://www.ebay.fr>
chercher le produit `Pikachu 58/102`



dans la colonne de gauche du site, filtrer la recherche en choisissant dans la rubrique `Afficher uniquement` l'option `Ventes réussies`

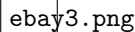


Vérifier bien que l'url de la page web courante du site ressemble à ceci :

`https://www.ebay.fr/sch/i.html?_from=R40&_sacat=0&_nkw=Pikachu%2058/102&LH_Complete=1&LH_Sold=1&rt=nc`

Remarquer que l'espace dans la chaîne de recherche `Pikachu 58/102` a été remplacé par `%20`.

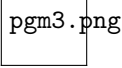
A présent, nous allons repérer le résultat de la recherche dans le fichier source HTML en cliquant sur le bouton droite de la souris et en choisissant `Examiner l'élément` dans le menu contextuelle.



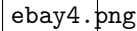
La liste des produits vendus `<ul class="srp-results srp-list clearfix">` est caractérisée par les trois classes `srp-results`, `srp-list` et `clearfix`. Les éléments de cette liste `` sont caractérisés par la classe `class="s-item"`. Donc, en python :

- On télécharge le document de la page web à l'aide de la méthode `urllib.request.urlopen(url)`
- A l'aide de la classe `BeautifulSoup` on parse le document HTML : `soup = BeautifulSoup(html_doc, "html.parser")`
- On cherche dans le document parsé l'ensemble de produits vendus par `soup.select("li.s-item")[1:]` puis on affiche la liste avec une simple boucle `for`.

D'où le code python suivant :

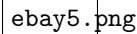


On cherche à extraire pour chaque produit vendu :



- Le titre, exemple : **Spanish 1st Edition Pikachu Base Set 58102 Pokemon Card**,
- Le prix, exemple **43,50 EUR**,
- La date, exemple **oct.-25 17:18**
- La Provenance, exemple **Espagne**

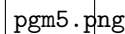
Il suffit pour cela de repérer la donnée dans le fichier source HTML et d'appliquer une méthode de sélection convenable :



Par exemple le prix est un `` de classe `POSITIVE` dans un `` de classe `s-item__price` . d'où l'instruction python :

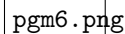
```
prix=k.select(".s-item__price.POSITIVE")
prix=prix[0].text
```

il en est de même pour les autres données :

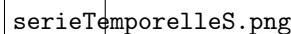


Attention : à rectifier dans ce code : `div` à la place de `h3` à la ligne 9.

1. Ce programme récupère les données de la page web concernant les ventes de l'article **Pikachu 58/102**. Compléter le code pour sauvegarder le résultat dans un fichier excel **ventes1.xlsx**.
2. Écrire une fonction **parseDate** permettant de transformer une chaîne de caractères de type : "Vendu 25 sept. 2021" en une date de type `datetime64`.



3. Écrire une fonction **getPrix** permettant de transformer une chaîne de caractères de type "41,82 EUR" en un nombre réel 41.82
4. Construire une série temporelle de type `pandas.Series` indexant par date le prix de vente des cartes Pikachu 58/102 :



5. Afficher les caractéristiques statistiques des ventes à l'aide la méthode `describe()` puis représenter graphiquement l'évolution du prix en fonction du temps.