

# TP #5- Correction

## Expressions régulières et manipulations de fichiers

---

### I. Création d'un nouveau projet

Ouvrez l'IDE de votre choix (VSCode ou PyCharm), et initiez un projet vierge (tp5).

Créez un fichier de type python (tp5.py).

Récupérez le fichier index.html qui contient le code source d'une page web, et placez le dans le répertoire du projet.

### II. Récupération du contenu d'un fichier

#### Exercice 1

Récupérez le code source de la page web dans le fichier index.html, et affichez le dans la console.

```
with open('index.html', 'r', encoding='utf8') as fichier:
    contenu = fichier.read()
print(contenu)
```

```
<script type="text/plain" data-gdpr-purposes="analytics" data-gdpr-src="https://news.google.com/swg/js/v1/swg.js"
  async="1" subscriptions-control="manual"></script>
<script type="text/plain" data-gdpr-purposes="personalization"
  data-gdpr-src="//www.lemonde.fr/bucket/e2e4e01041f5d37fdb0a49c874e73850aae45869/js/batch_push.bundle.js"
  async="1"></script>
<script>var ADS_CONFIG = { "adUnit": "\128139881\LM_lemonde\a_la_une\a_la_une\hp", "targets": { "keywords":
<script type="text/plain" data-gdpr-purposes="ads" data-gdpr-src="https://wrapper.lemde.fr/v2/glmaw.js"
  data-gdpr-no-consent-src="https://wrapper.lemde.fr/v2/consentless.js" async="1"></script>
</body>
</html>
```

### III. Utilisation d'expressions régulières (RegEx)

#### Exercice 2

En utilisant une expression régulière, récupérez les liens du code source et affichez-les.

*Indice : Les liens sont contenus dans les attributs href des balises HTML.*

```
regex = 'href=\"(.+?)\"'
res = re.findall(regex, contenu)

for link in res :
    print(link)
```

```
https://moncompte.lemonde.fr/cgv
https://www.lemonde.fr/faq/
https://www.facebook.com/lemonde.fr
https://www.youtube.com/user/LeMonde
https://twitter.com/lemondefr
https://www.instagram.com/lemondefr?hl=fr
https://www.snapchat.com/discover/Le-Monde/8843708388
https://www.lemonde.fr/actualite-medias/article/2019/08/12/les-flux-rss-du-monde-fr_5498778_3236.html
```

### Exercice 3

Modifiez l'expression régulière de sorte à ne récupérer que les liens des articles.

*Indice : Les articles contiennent tous le pattern /article/, suivis d'une date, et se terminent par .html*

```
regex = 'href=\"(.+\/article\/.+?)\"'
```

```
https://www.lemonde.fr/guides-d-achat/article/2021/10/25/les-meilleurs-aspirateurs-robots_6099813_5306571.html
https://www.lemonde.fr/guides-d-achat/article/2018/06/30/les-meilleurs-antivols-pour-attacher-son-velo_5323670_5306571.html
https://www.lemonde.fr/qui-sommes-nous/article/2007/11/17/talents-un-site-d-emploi-coedite-par-le-monde-interactif-et-telerama_978404_3386.html
https://www.lemonde.fr/actualite-medias/article/2010/11/03/la-charte-d-ethique-et-de-deontologie-du-groupe-le-monde_1434737_3236.html
https://www.lemonde.fr/actualite-medias/article/2010/11/03/la-charte-d-ethique-et-de-deontologie-du-groupe-le-monde_1434737_3236.html
https://www.lemonde.fr/actualite-medias/article/2019/08/12/les-flux-rss-du-monde-fr_5498778_3236.html
```

### Exercice 4

Pour chaque lien, extrayez la date et affichez-la au format « yyyy-mm-dd ».

```
for link in res :
    regex = '\/([0-9]{4})\/([0-9]{2})\/([0-9]{2})\/'
    res = re.search(regex, link)
    if res :
        annee = res[1]
        mois = res[2]
        jour = res[3]

        date = annee + "-" + mois + "-" + jour
        print(date)
```

```
2022-10-07
2021-10-07
2021-10-25
2018-06-30
2007-11-17
2010-11-03
2010-11-03
2019-08-12
```

### Exercice 5

Instanciez un dictionnaire, qui contiendra les dates et le nombre d'articles publiés ce jour.

Pour chaque date, incrémentez le compteur.

```
dic = {}
for link in res :
```

```

regex = '(/[0-9]{4})/([0-9]{2})/([0-9]{2})/'
res = re.search(regex, link)
if res :
    annee = res[1]
    mois = res[2]
    jour = res[3]

    date = annee + "-" + mois + "-" + jour
    if date in dic :
        dic[date] = dic[date] + 1
    else :
        dic[date] = 1
print(dic)

```

```

{'2022-11-07': 9, '2022-11-08': 39, '2022-11-05': 1, '2022-11-06': 1, '2022-10-07': 2, '2022-11-03': 1, '2021-10-07': 1, '2021-10-25': 1, '2018-06-30': 1, '2007-11-17': 1, '2010-11-03': 2, '2019-08-12': 1}

```

#### IV. Ecriture d'un fichier

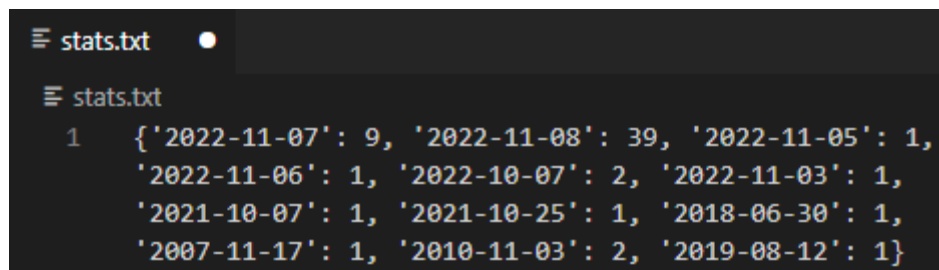
##### Exercice 6

Ecrivez ces statistiques dans un fichier.

```

fichier = open("stats.txt", "w")
fichier.write(str(dic))
fichier.close()

```



The screenshot shows a text editor window titled 'stats.txt'. The content of the file is a Python dictionary representing the statistics, formatted as a single line of code:

```

1 {'2022-11-07': 9, '2022-11-08': 39, '2022-11-05': 1, '2022-11-06': 1, '2022-10-07': 2, '2022-11-03': 1, '2021-10-07': 1, '2021-10-25': 1, '2018-06-30': 1, '2007-11-17': 1, '2010-11-03': 2, '2019-08-12': 1}

```