Objectif



L'objectif de ce TP est de vous initier à la génération automatique de documents numériques (ici des pages HTML) à l'aide du moteur de templates Jinja.

À retenir

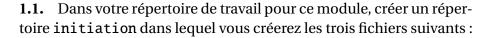
Avec un moteur de templates, la génération d'une page HTML se fait en trois parties :

- Une partie *modèle* contenant les *données* ainsi que tous les algorithmes qui permettent de traiter ces données (filtres, tris).
- Une partie *vue* contenant du code python et permettant de construire les visuels, c'est à dire le ou les fichiers html.



• Une partie *mise en forme* qui est un *template* contenant du html et un peu de code ressemblant à python.

Exercice 1. Les courses de la Famille Adams





Documents numériques P2 (FEUILLE n°1)

```
____ initiation_template.html __
<!DOCTYPE html>
<html lang="fr">
<head>
        <meta charset="utf-8"/>
        <title>TP7</title>
</head>
<body>
        <header>
                <h1> Les courses de \{\{nom\}\}\ </h1>
        </header>
    <h2>Produits achetés :</h2>
        {% for produit in courses %}
        <\!\!\mathbf{li}\!\!>\!\!\{\{\mathtt{produit}\}\}<\!\!/\mathbf{li}\!\!>
        {% endfor %}
        <h2>Facture :</h2>
        <u1>
        {% for i in range(prix|length) %}
        {{prix[i]}} euros
        {% endfor %}
        Total : {{total}}
</body>
</html>
                                            __ initiation_vue.py __
#!/usr/bin/python3
# =========
# NOM :
# -----
from jinja2 import Environment, FileSystemLoader # ne pas modifier
# Ajouter ici les éléments du modèles dont on a besoin
from initiation_modele import *
# ne modifiez pas cette fonction !
def creer_html(fichier_template, fichier_sortie,**infos):
    Cette fonction génère automatiquement un fichier à partir d'un
    template et d'informations
    paramètres :
        fichier_template : un fichier template (template HTML par exemple)
        fichier_sortie : le nom du fichier généré
        **infos : un nombre indéfini de paramètres qu'il suffit de nommer
    return : -
    env=Environment(loader=FileSystemLoader('.'),trim_blocks=True)
    template=env.get_template(fichier_template)
   html=template.render(infos)
    f = open(fichier_sortie, 'w')
    f.write(html)
    f.close()
# Ajouter ici les appels à la fonction creer_html
creer_html("initiation_template.html", "initiation.html",
            nom=morticia.
            courses=courses_morticia,
            prix=facture_morticia,
            total=somme(facture_morticia),
            )
```

Documents numériques P2 (FEUILLE n°1)

- **1.2.** Maintenant, exécutez le script qui permet de générer le visuel.
- **1.3.** Dans la *vue*, ajoutez un appel à la fonction creer_html de façon à générer une page initiation2.html pour les courses de Gomez.
- **1.4.** Modifiez le *modèle* de façon à obtenir un total correct. Générez les visuels à nouveau.

Attention

Dans chaque exercice, vous devrez donc:

- Définir les données dans un (ou plusieurs) fichier exercice_modele.py.
- Créer un fichier exercice_vue.py contenant au minimum la fonction creer_html donnée dans l'exemple ainsi qu'un appel à cette fonction.
- Créer un (ou plusieurs) fichier exercice_template.html.



Pour chercher des informations sur Jinja, la documentation officielle se trouve là :

http://jinja.pocoo.org/docs/2.9/ (adresse à mettre dans vos favoris)

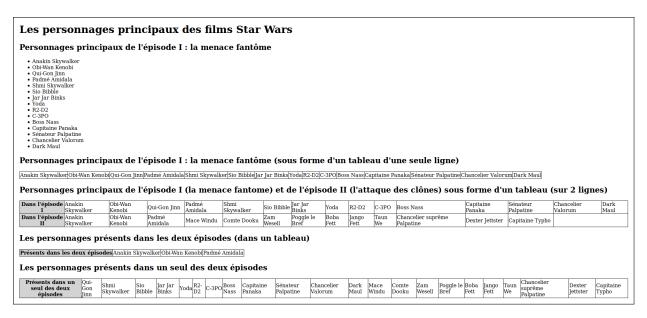
Exercice 2. Personnages principaux des films Star Wars

Dans cet exercice, le modele est le suivant :

```
#!/usr/bin/python3
# =======
# Les données
# =======
l_attaque_des_clones=["Anakin_Skywalker", "Obi-Wan_Kenobi", "Padmé_Amidala", "Mace_Windu", "Comte_Dooku", "Zam_Wesell", "Poggle_le_Bref", "Boba_Fett", "Jango_Fett", "Taun_We",
   "Chancelier_suprême_Palpatine", "Dexter_Jettster", "Capitaine_Typho"
la_revanche_des_sith=["Anakin_Skywalker", "Dark_Vador", "Padmé_Amidala", "Palpatine"
    "Obi-Wan_Kenobi", "Mace_Windu", "Comte_Dooku", "C-3PO", "R2-D2", "Yoda", "Général_Grievous",
   "Chewbacca", "Tarfful", "Bail_Organa"
un_nouvel_espoir=["Luke_Skywalker", "Han_Solo", "Princesse_Leia_Organa", "Le_Grand_Moff_Tarkin",
   "Obi-Wan_Kenobi", "C-3PO", "R2-D2", "Chewbacca", "Dark_Vador", "Owen_Lars", "Beru_Lars"
"Boba_Fett"
le_retour_du_jedi=["Luke_Skywalker", "han_Solo", "Princesse_Leia_Organa", "Lando_Calrissian", "C-3PO", "R2-D2","Chewbacca", "Dark_Vador", "Empereur_Palpatine", "Yoda", "Obi-Wan_Kenobi"
# Outils de traitement des données
```

Documents numériques P2 (FEUILLE n°1)

- 2.1. Dans un dossier dédié à cet exercice, créez un fichier personnagesSW_vue.py et un fichier personnagesSW_template.html CONTENANT UN SQUELETTE MINIMAL : une fois le script personnagesSW_vue.py exécuté, vous devez obtenir un fichier personnagesSW_episodes_I_et_II.html avec le titre "Personnages principaux des films Star Wars (épisodes I et II)" et SANS RIEN D'AUTRE.
- **2.2.** Modifiez la vue et le template de manière à mettre afficher dans le html généré le nom des personnages de l'épisode I (la menace fantôme) sous forme d'une liste.
- **2.3.** Modifiez le template pour afficher (en dessous de la liste précédente) les personnages de l'épisode I dans un tableau d'une seule ligne.
- **2.4.** Modifiez vue et template afin d'afficher dans un même tableau : sur la première ligne, les personnages de l'épisode I, et sur la seconde, les personnages de l'épisode II (l'attaque des clônes).
- **2.5.** Comment faire pour afficher les noms des personnages présents à la fois dans l'épisode I et à la fois dans l'épisode II? Ajoutez ce qu'il faut à votre code.
- **2.6.** Créez une section avec les personnages qui ne sont présent que dans un seul des deux épisodes.
- **2.7.** Ajoutez un CSS pour que votre page ressemble à la copie d'écran suivante :



2.8. Modifiez votre code pour pouvoir générer une page personnages SW_episodes_IV_et_V.html qui fait la même chose que précedemment, mais avec les personnages des épisodes IV (Un nouvel espoir) et V (L'empire contre attaque)

Documents numériques P2 (FEUILLE $n^{\circ}1$)

2.9. Défi

Dans le dernier tableau, mettez en bleu les personnages présents dans l'épisode I, et en vert les personnages présents dans l'épisode II.

Indication



Jinja possède une structure de contrôle if ... else ... (voir la doc) Mais attention à l'utiliser que lorsqu'il s'agit de mise en forme et non quand il s'agit de filtrer les données!!