

**Consignes.** Avant toute chose, prenez conscience que les corrections sont effectuées à l'aide d'un automate, par conséquent toute violation des règles conduira inévitablement à la délivrance d'une note nulle.

- Comment livrer mon code ?  
Si mon nom de famille est **yunès** et que je souhaite rendre le travail pour l'exercice 12 du tp 3 alors je dois construire une archive au format ZIP de nom **yunes-tp3-ex12.zip**, le nom ayant été converti en minuscules, sans accent et sans espace (*ie* : **de Gaulle** doit être traduit en **degaulle**). Si vous avez un homonyme rajoutez des initiales, par exemple **yunesjb-tp1-ex1**. Respectez votre nommage pour tout le semestre.
- Que mettre dans l'archive ?  
Le code source (et uniquement le code source) correspondant à l'exercice.

De plus, suivez bien les instructions complémentaires des livrables.

## 1 Exercice 1 (lecture de fichier)

Écrire un programme python qui :

- prend en argument un nom de fichier (contenant des lignes de texte) et affiche à l'écran une ligne sur deux,

Livable : l'archive contenant le programme `oneOverTwo.py`.

## 2 Exercice 2 (lecture/écriture)

Écrire un programme python qui :

- prend en argument deux noms de fichiers et copie le contenu du premier dans le second,
- différents messages d'erreurs doivent être affichés pour informer l'utilisateur de différents problèmes (ex. : fichiers inexistant, fichier déjà existant, problème ouverture, etc).

Livable : l'archive contenant le programme `copy.py`.

## 3 Exercice 3 (regex)

Écrire un programme qui :

1. permet de remplacer dans un fichier dont le nom est donné en argument toute suite d'espace, tabulation par un seul caractère d'espacement,
2. le programme générera en sortie un fichier de même nom additionné de l'extension `.clean`.
3. un fichier source d'exemple est fourni (`lafontaine.txt`)
4. dans un second temps on ajoutera :
  - on prendra soin d'enlever intégralement les espacements en début et fin de ligne.
  - on prendra soin de respecter les règles de ponctuations (un espace avant `?` et aucun avant `, ; .` et toujours un ensuite en sauf fin de ligne).

Livable : l'archive contenant le programme `clean.py`.

## Exercice 4 (regex)

Écrire un programme Python qui :

1. prend un fichier `.html` et supprime de celui-ci toutes les balises, le fichier d'entrée sera donné en argument et le fichier produit en sortie sera d'extension `.nohtml`.

Livable : l'archive contenant le programme `dehtml.py`.

## Exercice 5 (regex)

Écrire un programme Python qui :

1. saisie en entrée un mot de passe avec les caractéristiques suivantes :
  - entre 8 et 12 caractères,
  - contient au moins une majuscule,
  - contient au moins une minuscule,
  - contient au moins deux chiffres.

Livrable : l'archive contenant le programme `passcheck.py`.

## Exercice 6 (regex)

Transformer le programme de l'exercice précédent de sorte qu'en plus la vérification assure que le mot de passe ne contient aucun des mots contenus dans la liste des mots interdits et contenue dans le fichier `forbidden.txt` (une ligne = un mot interdit)

Livrable : l'archive contenant le programme `passcheck2.py`.

## Exercice 7 (regex)

Écrire un programme Python qui :

1. vérifie que la saisie de l'utilisateur est une URL valide :
  - commence par un protocole autorisé parmi (`http`, `https`, `ftp`) lequel doit être suivi de `://`
  - suivi par un nom de domaine valide (au moins deux «mots» séparés par des `.`)
  - peut-être suivi par la spécification d'un numéro de port : suivi par des chiffres
  - le reste étant constitué de chiffres, lettres, points, slashes, etc.

Livrable : l'archive contenant le programme `validurl.py`.