

Projet Python



Logo Python (Source : <https://fr.wikipedia.org/wiki/Fichier:Python-logo-notext.svg>)

Table des matières

1	Introduction.....	3
1.1	Formation des groupes	3
1.2	Date Rendu	4
1.3	Format documentation	4
1.4	Evaluation	4
1.5	Conseils.....	5
1.6	Questions & autonomie	5
2	Projet 1 : Triage automatisé de fichiers reçu par FTP (Linux/Windows).....	6
2.1	Description	6
2.2	Prérequis	6
2.3	Rendu	6
3	Projet 2 : Système de log (Linux/Windows)	7
3.1	Description	7
3.2	Prérequis	7
3.3	Rendu	7
4	Projet 3 : Récolte et nettoyage de données automatisées (Linux/Windows)	8
4.1	Description	8
4.2	Prérequis	8
4.3	Rendu	8
5	Projet 4 : Monitoring PC (Linux/Windows)	9
5.1	Description	9
5.2	Prérequis	9
5.3	Rendu	9
6	Projet 5 : Informations utilisateur/groupe (Linux/Windows)	10
6.1	Description	10
6.2	Prérequis	10
6.3	Rendu	10
7	Projet 6 : Automatique unzip mail (Linux/Windows).....	11
7.1	Description	11
7.2	Prérequis	11
7.3	Rendu	11

1 Introduction

Ce document a pour but de décrire les sujets des différents projets Python pour le cours 63-12.

1.1 Formation des groupes

Vous formez les groupes et les retourner par e-mail **d'ici au 14 octobre 2021** à xavier.barmaz@hevs.ch et jeremie.vianin@hevs.ch

- Groupe 1 : 23 étudiants
- Groupe 2 : 21 étudiants

Un total de 44 étudiants est inscrit.

Introduction to information technology			
601_FR_GR1	Barmaz X. / Vianin J. / Le Calvé A.	601_FR_GR2	Barmaz X. / Vianin J. / Le Calvé A.
Andriulo	Tatiana	Délèze	Yann
Auchère	Arnaud	Delieutraz	Nohen
Barbe	Noah	Demir	Oktay
Berisha	Blendar	Destefani	Adrien
Blanch-Lanao	Julien	Favre	Julien
Celik	Turgay	Fournier	Benoît
Chiolini	Alessandro	Gallati	Eric
Compain	Milan	Gianadda	David
De Carvalho	Tony	Gillioz	Loïs
De Fino	Francesco	Hallén	Oscar
Grujicic	Damjan	Mayencourt	Valentin
Hebert	Patrik	Melchior	Nathan
Henriques Amaral	Guilherme Filipe	Mercuri	Chris
Herbelin	Dehlya	Moret	Léon
Le Goff	Benoît	Paz	Maria
Le Meillour	Anthony	Pereira	Adrian
Loureiro	Christophe	Perrin	Julien
Mabillard	Marie-Esther	Praz	Eliot
Martins	Diogo	Roch	Loris
Mbaye	Khady	Rodrigues	Tony
Mina	Ramy	Toplana	Erona
Salamin	Alexandre	Tscherrig	Léticia
Salamin	Guillaume	Uka	Zotrim
Savioz	Mathieu	Vuignier	Thaddée
Schneider	Bastien		
Free Auditor			

Les groupes à former sont de **4 étudiants** :

- Projet 1 : Triage automatisé de fichiers reçu par FTP (Windows) : 4 étudiants
- Projet 1 : Triage automatisé de fichiers reçu par FTP (Linux) : 4 étudiants
- Projet 2 : Système de log (Windows) : 4 étudiants
- Projet 2 : Système de log (Linux) : 4 étudiants
- Projet 3 : Récolte et nettoyage de données automatisées (Windows) : 4 étudiants
- Projet 3 : Récolte et nettoyage de données automatisées (Linux) : 4 étudiants
- Projet 4 : Monitoring PC (Windows) : 4 étudiants
- Projet 4 : Monitoring PC (Linux) : 4 étudiants
- Projet 5 : Informations utilisateur/groupe (Windows) : 4 étudiants
- Projet 5 : Informations utilisateur/groupe (Linux) : 4 étudiants

- Projet 6 : Automatique unzip mail (Windows) : 4 étudiants
- Projet 6 : Automatique unzip mail (Linux) : 4 étudiants

La réalisation du projet (Windows ou Linux) sera attribuée une fois les groupes reçus.

Vous choisissez le projet, les profs choisissent l'OS

1.2 Date Rendu

16.12.2021 à 18h.00

Le projet est rendu sur «<https://filesender.switch.ch/>» ou autre dans un seul **ZIP** avec lien à transmettre à xavier.barmaz@hevs.ch et jeremie.vianin@hevs.ch.

- «NomProjet_Nom1-Nom2-Nom3-Nom4-NomX.zip»
 - «NomProjet_Nom1-Nom2-Nom3-Nom4-NomX.docx»
 - «NomScriptXXX_Nom1-Nom2-Nom3-Nom4-NomX.py»

1.3 Format documentation

Le rendu se fait au nombre de 2 documents :

- Documentation
- Script(s) Python

La **documentation** (docx) doit être au format APA avec tout le formalisme académique exigé :

- table des matières
- tables des illustrations
- tables des abréviations
- résumé exécutif
- introduction
- développement
- pistes d'améliorations
- conclusion générale
- conclusion personnelle
- références
- etc...

Voir chaque projet pour les points importants à réaliser dans la documentation.

1.4 Evaluation

Voir le fichier annexe « 02. GrilleEvaluation.pdf »

100% de la note du CC → note individuelle

Il faut indiquer qui a réalisé quel point (documentation, scripts)

Généralités :

- Être expert du domaine → concepts avancés, illustrations, démos, fond, forme
- Chaque étudiant s'investit dans une partie de matière équivalente
- Reproductibilité – Réutilisabilité pour le correcteur et les autres groupes

Documentation + Pratique (format docx) :

- Guide: Qualité et clarté des explications (forme APA)
- Pratique: Qualité de la pratique, reproductibilité pour les étudiants

Division du travail au sein du groupe mais connaissances identiques de tous

Matière d'examen

1.5 Conseils

Débuter vers début novembre après avoir traité les matières de cours, notamment :

- Norme APA et rédaction d'un document
- Base de la programmation (entier, float, chaînes de caractères, boucles, fonctions)

Vous pouvez dès à présent débiter les recherches sur Python, mais les autres cours vont vous aider à réaliser ce qui est demandé dans les projets.

1.6 Questions & autonomie

En cas de questions durant la réalisation du projet, merci de vous adresser à jeremie.vianin@hevs.ch.

Les groupes sollicitant de manière trop fréquente l'aide de Jérémie ou Xavier, seront pénalisés dans l'évaluation du projet.

2 Projet 1 : Triage automatisé de fichiers reçu par FTP (Linux/Windows)

2.1 Description

L'utilisateur reçoit des fichiers divers sur un FTP (mis à disposition de l'étudiant).

L'objectif de ce projet est de créer un script Python permettant la récupération et le tri automatisé des fichiers arrivant dans le FTP :

- si le fichier est un .png, .jpeg, .jpg, il sera déplacé du FTP à un dossier « image ».
- si le fichier est un .docx, .xlsx, .pptx, il ira dans un dossier « document ».
- lorsqu'une extension n'est pas répertoriée, le fichier est stocké dans un répertoire « divers ».

Lorsque le dernier cas survient, un mail doit être envoyé à l'utilisateur pour lui indiquer que l'extension en question n'est pas encore enregistrée et que son fichier est stocké dans le dossier « divers ».

2.2 Prérequis

Les groupes reçoivent, en début de projet, les éléments suivants :

- Les accès au FTP (une semaine après la formation des groupes)

Les groupes génèrent par eux-mêmes les fichiers avec extensions :

- .png, .jpeg, .jpg
- .docx, .xlsx, .pptx

2.3 Rendu

La **documentation** doit inclure les points importants du développement suivants :

- explication de la mise en place de l'environnement
- description des principales fonctions du code
- visuel + explications du cheminement des fichiers (automatisation du processus visio ou autre)

Le **script Python**, remis comme code de preuve de la réalisation du projet doit :

- avoir des commentaires expliquant les fonctions
- être au format standard de présentation/découpage/fonction du monde professionnel
- pouvoir être exécuté par les correcteurs

3 Projet 2 : Système de log (Linux/Windows)

3.1 Description

Un script Python est mis en place pour analyser les logs du système (Windows ou Linux).

Le script Python est exécuté toutes les nuits à 01:00 am et doit fournir :

- un premier fichier .txt ou .csv (ou .json pour les plus avancés) contenant l'ensemble des logs analysés
- un second fichier contenant un récapitulatif (1 critique, 32 warning, 30 info, etc...) de la journée précédente

Lorsqu'un log est en statut « Critique », le script automatisé doit en informer au plus vite l'utilisateur par e-mail avec le log en statut « critique » et les informations jugées pertinentes à la compréhension rapide de la situation.

3.2 Prérequis

Il n'y a pas de prérequis dans ce projet.

3.3 Rendu

La **documentation** doit inclure les points importants du développement suivants :

- explication de la mise en place de l'environnement
- description des principales fonctions du code
- visuel + explications du cheminement des fichiers (automatisation du processus visio ou autre)

Le **script Python**, remis comme code de preuve de la réalisation du projet doit :

- avoir des commentaires expliquant les fonctions
- être au format standard de présentation/découpage/fonction du monde professionnel
- pouvoir être exécuté par les correcteurs

4 Projet 3 : Récolte et nettoyage de données automatisées (Linux/Windows)

4.1 Description

Le script doit s'activer automatiquement lorsque l'un des fichier de test (valide ou non-valide) se trouve sur le FTP.

Dans un premier temps, le script doit vérifier que le fichier reçu contient toutes les colonnes du fichier de test valide. Si ce n'est pas le cas, le script Python annule le processus, place le fichier dans un dossier spécifique du FTP et avertit l'utilisateur par email.

Dans un second temps si le fichier est valide, le script vérifie que les données (les lignes) contenues dans le fichier test (valide) sont corrects. Certaines lignes du fichier de test (valide) vont avoir des erreurs (un string à la place d'un double, des données manquantes, etc...).

- Le script doit, une fois le fichier récupéré, réaliser un nettoyage automatisé des lignes erronées (suppression de la ligne en question).
- Une fois le nettoyage réalisé, le script doit envoyer sur un second FTP le nouveau fichier nettoyé.

Un email de confirmation contenant le nombre de lignes effacées doit être envoyé à l'utilisateur en cas de succès de la procédure.

4.2 Prérequis

Les groupes reçoivent, en début de projet, les éléments suivants :

- Les accès aux FTP (une semaine après la formation des groupes)
- Un fichier de test valide et non-valide contenant des données anonymisées (une semaine après la formation des groupes)

4.3 Rendu

La **documentation** doit inclure les points importants du développement suivants :

- explication de la mise en place de l'environnement
- description des principales fonctions du code
- visuel + explications du cheminement des fichiers (automatisation du processus visio ou autre)

Le **script Python**, remis comme code de preuve de la réalisation du projet doit :

- avoir des commentaires expliquant les fonctions
- être au format standard de présentation/découpage/fonction du monde professionnel
- pouvoir être exécuté par les correcteurs

5 Projet 4 : Monitoring PC (Linux/Windows)

5.1 Description

Le script Python vérifie que les performances de l'ordinateur sur lequel il est lancé sont bonnes. Il doit s'assurer que :

- la RAM est inférieure à 90% de son maximum
- le processeur n'atteint pas le 97% de sa disponibilité
- les disques/volumes/partitions restent en dessous de 90% d'utilisation et qu'il ne soit pas rempli à plus de 75% de sa capacité

Si une de ces conditions n'est plus respectée, le problème est remonté dans un fichier de log. Une nouvelle ligne y est créée contenant la date, l'heure, l'élément problématique (Ram, Disque, etc...), et le pourcentage lors du dépassement. Ce fichier de log est journalier et est transmis à minuit à l'utilisateur via email.

5.2 Prérequis

Il n'y a pas de prérequis dans ce projet.

5.3 Rendu

La **documentation** doit inclure les points importants du développement suivants :

- explication de la mise en place de l'environnement
- description des principales fonctions du code
- visuel + explications du cheminement des fichiers (automatisation du processus visio ou autre)

Le **script Python**, remis comme code de preuve de la réalisation du projet doit :

- avoir des commentaires expliquant les fonctions
- être au format standard de présentation/découpage/fonction du monde professionnel
- pouvoir être exécuté par les correcteurs

6 Projet 5 : Informations utilisateur/groupe (Linux/Windows)

6.1 Description

Audit

Le script Python crée un fichier de log avec tous les utilisateurs/groupes actuels du système et le met à jour automatiquement lors d'ajout, de suppression ou de modification.

Création/Suppression/Modification

Pour la réalisation du test, le script doit aussi pouvoir créer/supprimer/modifier des utilisateurs/groupes.

Mail

Lorsqu'un utilisateur est ajouté/modifié/supprimé, un mail d'information est envoyé à l'utilisateur et cette activité est retranscrite dans un second fichier de log. Une nouvelle ligne est ajoutée contenant la date, l'heure, l'utilisateur (ajouté ou supprimé).

6.2 Prérequis

Il n'y a pas de prérequis dans ce projet.

6.3 Rendu

La **documentation** doit inclure les points importants du développement suivants :

- explication de la mise en place de l'environnement
- description des principales fonctions du code
- visuel + explications du cheminement des fichiers (automatisation du processus visio ou autre)

Le **script Python**, remis comme code de preuve de la réalisation du projet doit :

- avoir des commentaires expliquant les fonctions
- être au format standard de présentation/découpage/fonction du monde professionnel
- pouvoir être exécuté par les correcteurs

7 Projet 6 : Automatique unzip mail (Linux/Windows)

7.1 Description

L'utilisateur reçoit sur un FTP un fichier ZIP contenant :

- un fichier .txt contenant uniquement un email. Ce mail est défini par le groupe.
- minimum deux autres fichiers. Le groupe choisit par eux-mêmes ces fichiers.

Dans un premier temps, le script python doit dézipper le ZIP et doit vérifier que le mail est correctement construit (text@text.text). En cas d'erreur, le fichier ZIP sera déplacé dans un dossier nommé « Echec » sur le même FTP.

Dans un second temps, le script doit envoyer un mail contenant les fichiers du ZIP (sauf le .txt) à l'email inscrit dans le .txt. Une fois la procédure réalisée, le script Python déplace le fichier ZIP dans le FTP dans un dossier nommé « Succes ».

7.2 Prérequis

Les groupes reçoivent, en début de projet, les éléments suivants :

- Les accès au FTP (une semaine après la formation des groupes)

7.3 Rendu

La **documentation** doit inclure les points importants du développement suivants :

- explication de la mise en place de l'environnement
- description des principales fonctions du code
- visuel + explications du cheminement des fichiers (automatisation du processus visio ou autre)

Le **script Python**, remis comme code de preuve de la réalisation du projet doit :

- avoir des commentaires expliquant les fonctions
- être au format standard de présentation/découpage/fonction du monde professionnel
- pouvoir être exécuté par les correcteurs