

Docu SAE15

1) Présentation du projet

Dans le cadre de la première année de BUT réseaux et télécommunications à l'IUT de Valence, nous avons dû réaliser une SAE en groupe, sur de la programmation (Python, HTML, CSS...). Notre groupe est composé de 5 élèves : Bastien Soleihac (chef de groupe), Lily Lane, Fabien Roux, Yannis Anghour et Louis Hauersperger. Cette SAE est composée de 3 livrables que nous avons dû réaliser selon les dates suivantes :

-Livrable 1 : 15 décembre

-Livrable 2 : 10 Janvier

-Livrable 3 : 15 Janvier

Cette SAE en groupe a nécessité de la planification et de l'organisation pour répartir le travail. Pour le premier livrable, nous avons chacun réalisé un petit script qui extrait des informations de fichiers CSV. Ces petits scripts étaient exécutés depuis un menu affiché par le script « main », script réalisé par Bastien Soleihac. Pour le deuxième livrable le travail a été réparti un peu différemment. Nous devons réaliser un site web généré dynamiquement selon les informations extraites par nos scripts python. Nous avons chacun continué à travailler sur nos scripts en réalisant une deuxième partie créant des diagrammes statistiques ainsi qu'une partie qui modifie chacune de nos pages web pour afficher nos résultats. Bastien Soleihac s'est occupé de la gestion du site web : la sauvegarde de l'ancienne version ainsi que la génération des différentes pages pour chaque membre du groupe grâce au Template. Louis Hauersperger s'est occupé de la page « synthèse de l'évaluation Moodle » qui regroupe différentes données statistiques récupérés sur les résultats du groupe lors d'une évaluation Moodle de ce semestre.

Sujets choisis et fonctionnalités réalisés par chacun :

Bastien Soleihac :

Sujet choisi : sujet 17. Fonctionnalités réalisées :

-Lecture d'un fichier CSV et extraction des données

-Création d'un histogramme grâce à la bibliothèque Matplotlib

Script également réalisé : le « main ». Fonctionnalités réalisées :

-Sauvegarde du site

-Suppression du site

-Génération du site avec le template

-Rajout dynamique des pages perso selon le fichier « configuration.txt »

-Modification dynamique des pages perso (Ajout titre et nom)

-Dézippage des fichiers de l'opendata

-Appel de chacune des fonctions « MainNom » des membres du groupe.

-Suppression des fichiers dézippés de l'opendata

-Affichage du site

Lily Lane :

Sujet choisi : sujet 2. Fonctionnalités réalisées :

- Lecture d'un fichier CSV et extraction des données
- Création d'un histogramme grâce à la bibliothèque Matplotlib

Fabien Roux :

Sujet choisi : sujet 9. Fonctionnalités réalisées :

- Lecture d'un fichier CSV et extraction des données
- Création d'un histogramme grâce à la bibliothèque Matplotlib

Yannis Anghour :

Sujet choisi : sujet 15. Fonctionnalités réalisées :

- Lecture d'un fichier CSV et extraction des données
- Création d'un histogramme grâce à la bibliothèque Matplotlib

Louis Hauersperger:

Sujet choisi : sujet 18. Fonctionnalités réalisées :

- Lecture d'un fichier CSV et extraction des données
- Création d'un histogramme grâce à la bibliothèque Matplotlib

Script également réalisé : création de la page « Synthèse » de l'évaluation Moodle. Fonctionnalités réalisées :

- Lecture d'un fichier CSV et extraction des données
- Calculs statistiques (moyenne, écart type)
- Création d'un histogramme grâce à la bibliothèque Matplotlib

2) Manuel d'utilisation

Voici une rapide présentation de comment déployer le site correctement :

Premièrement placer les bons fichiers, zippés, dans le répertoire opendata.

Deuxièmement placer les bons fichiers, non zippés, dans le répertoire data

Troisièmement s'assurer de posséder la dernière version de python.

Quatrièmement installer la bibliothèque matplotlib si ce n'est pas déjà fait.

Cinquièmement s'assurer d'être dans le répertoire « groupe09_sae15_liv2 »

Dernièrement exécuter le script nommé « main » en renseignant les deux paramètres suivants :

1^{er} paramètre : le chemin vers le répertoire « data »

2^{ème} paramètre : le chemin vers le répertoire « opendata »

La commande par défaut à exécuter est **python3 ./source/main.py ./data ./opendata**

Et voilà ! Si vous suivez scrupuleusement ces étapes, le site devrait se déployer et s'afficher sur l'écran sans problème.

3) Documentation technique

Explication des différents scripts :

gestionConfig (méthode « recupDonneeEtudiant »)

Cette fonction prend en premier paramètre le chemin vers le fichier « configuration.txt » contenant les différentes informations sur les membres du groupe. En deuxième paramètre doit être renseigné un mot clé parmi : « nom », « login », « codePostaux », « communes », « sujet », « codeINSEE » et « prenom ». La fonction va ensuite lire le fichier « configuration.txt » et extraire les données spécifiées selon le mot clé (Par exemple « nom » rendra le nom de chacun des membres du groupe) avant de les rendre

gestionConfig (méthode « recupCheminData »)

Cette fonction lit le fichier configuration.txt pour récupérer les noms des sujets avant de les associer aux noms des fichiers CSV associés. La fonction prend en paramètre le chemin vers le répertoire data contenant le fichier configuration.txt. Tout est récupéré à partir du fichier configuration.txt avec « with open ».

gestionMoodle (méthode « evalMoodle »)

Cette fonction prend en paramètre le chemin vers le répertoire « data ». Elle va effectuer différents calculs depuis le fichier csv contenant les notes de tous les étudiants à l'évaluation Moodle effectuée au cours du semestre. Pour ce faire elle lit le fichier avec « with open » puis « reader » et recherche les noms des étudiants du groupe avec une boucle « if » elle récupère les différentes données dans des dictionnaires puis calcul et rend dans un tuple : La notes de tous les étudiants du groupe, l'écart-type du groupe, la moyenne du groupe ainsi que le temps moyen de réalisation des étudiants du groupe.

gestionMoodle(méthode « GraphMoodle »)

Cette fonction prend en paramètre le chemin vers le répertoire « data ». Elle va calculer le pourcentage de réussite des membres du groupe sur chacune des questions. Pour ce faire toujours pareil, elle lit le fichier csv contenant les notes à l'évaluation moodle ligne par ligne, tout en récupérant dans « dicomax » la note maximale atteignable sur chaque question. La fonction relit le fichier une deuxième fois en comparant les notes des membres du groupe aux notes maximales atteignable pour obtenir le pourcentage de réussite final dans « dicoresult ». Enfin elle va générer un histogramme (ptl.bar) à partir de matplotlib en indiquant en abscisse le numéro de la question et en ordonnée le pourcentage de réussite.

gestionOpenData (méthode « supprimerFichierOpenData »)

Cette procédure ne prend aucun paramètre ni ne renvoie rien. Elle va simplement supprimer tous les fichier « csv » et « pdf » du répertoire opendata.

gestionOpenData(méthode « extractsujet »)

Cette fonction prend en paramètre le chemin vers le fichier à extraire, appelle une autre fonction nommée « extractionZip » pour extraire le fichier csv puis rend le chemin d'accès vers ce fichier. Le chemin d'accès est légèrement modifié pour rajouter l'extension de fichier « .csv ».

gestionWeb(méthode « modifierWeb »)

Cette procédure prend trois paramètres : le premier, le chemin vers le fichier .html à modifier. Le deuxième, le repère ou le code doit être rajouté. Le troisième, le code à rajouter. Cette procédure permet d'éditer simplement et rapidement un fichier texte. Elle va ouvrir le fichier en écriture avec « with open » (sans oublier l'encodage « utf-8 » utilisé pour le fichier HTML), lire le fichier ligne par ligne jusqu'à tomber sur le repère, puis écrire code à rajouter après le repère.

gestionWeb(méthode « sauvegarderWeb »)

Cette procédure prend en paramètre le chemin vers le dossier « www » contenant la version courante du site web, ainsi que le chemin vers le dossier « sauvegarde ». Elle va archiver le répertoire www, grâce à « make_archive », avant de le placer dans le répertoire « sauvegarde » après avoir généré un nom comportant la date et l'heure.

gestionWeb(méthode « supprimerWeb »)

Cette procédure prend deux paramètres, le chemin vers le répertoire « www » et un booléen. Si le booléen est présent, alors ce sont les fichiers dans le répertoire qui sont supprimés, si le booléen n'est pas présent c'est le répertoire qui est supprimé (en effet la commande pour supprimer des fichier (rmtree) et celle pour supprimer un répertoire (remove) ne sont pas les mêmes, il faut donc une condition pour les distinguer.

gestionWeb(méthode « clonnerPageWeb »)

Cette procédure permet de cloner une page web en modifiant les titres selon l'étudiants à qui sont attribués les pages web. Elle prend en paramètre le chemin vers le fichier à copier (qui sera le Template du site), le nom que le nouveau fichier aura ainsi que nom et le prénom de l'étudiant à qui sera attribuée la page. Le fichier HTML du Template va donc être copié avec « copy » puis il sera modifié grâce à la fonction « modifierWeb » expliquée ci-dessus. Le nom de l'auteur, le titre, et le pied de page seront modifier pour afficher le nom de l'étudiant à qui est attribuée la page. La barre de navigation est aussi modifiée en conséquence pour intégrer la nouvelle page.

gestionZip (méthode « extractionZip »)

Cette procédure dézippe tout les fichiers zip contenus dans le répertoire appelé en paramètre grâce à la méthode « extractall ». Le paramètre par défaut est le chemin vers le Template html, la procédure prend aussi comme paramètre un booléen, s'il est présent, le répertoire ou seront dézippés les fichiers est le répertoire open data.

gestionZip (méthode « renommerFichier »)

Cette méthode utilise simplement la méthode « rename » qui renome le fichier en prenant le premier paramètre l'ancien nom et en deuxième paramètre, le nouveau nom. Cette procédure permet simplement de ne pas faire planter le code, en cas d'erreur, l'erreur est simplement affichée et n'arrête pas le fonctionnement du code.

diagrammeAnghour (méthode « sujet15Graph»)

Cette méthode permet d'obtenir le pourcentage de cours d'eau à sec pour chaque département de naissance des membres du groupe. Elle prend pour paramètre le chemin vers le fichier CSV contenant les données ainsi qu'un tuple contenant les listes renvoyées par la fonction « mainsujet15 ». La méthode va diviser le nombre de cours d'eau à sec (rendu par la fonction mainsujet15) par le nombre de cours d'eau total (rendu par la fonction mainsujet15) et le multiplier 100 pour obtenir un pourcentage. Un diagramme va ensuite être créé en prenant en abscisse le numéro de département suivis du numéro du mois et en ordonné, le pourcentage de cours d'eau à sec.

mainAnghour (méthode « mainAnghour»)

Cette méthode va venir exécuter les autres fonctions « anghour », récupérer les données, et les intégrer à la page web « anghour.html ». Elle prend en paramètre deux chemins, le premier vers le fichier CSV contenant les informations à extraire, le deuxième, le chemin vers le répertoire « opendata ». La méthode va appeler la procédure « modifierWeb » pour venir modifier la page web « anghour.html ».

sujet15 (méthode « mainsujet15 »)

Cette fonction permet d'extraire d'un fichier csv, le nom des cours d'eau des communes de naissance des membres du groupe, s'il n'y a aucun cours d'eau dans le relevé, la fonction prend le code postal immédiatement supérieur contenant un relevé, ces informations sont ensuite directement ajoutées à la page web « anghour.html ». Elle prend en paramètre le chemin vers le fichier CSV où extraire les données ainsi que le chemin vers le fichier opendata. La fonction va tout d'abord récupérer les codes Postaux des communes de naissance de chacun des membres du groupe grâce à la fonction « recuDonnéeEtudiant » (décrite ci-dessus) puis va lire le fichier csv grâce à « with open » et « csv.reader » en relevant les cours d'eau présent dans chaque commune dans un dictionnaire. Les données sont stockées dans un dictionnaire qui est ensuite utilisé pour modifier la page web « anghour.html ». Dans le même temps, la fonction récupère également le nombres de cours d'eau totaux, et à sec, dans chaque département de naissance de membres du groupe, pour chaque mois. Ces informations sont stockées dans des dictionnaires qui sont ensuite rendus sous forme de tuple.

diagrammeLane (méthode « sujet2Graph »)

La fonction « sujet2Graph » récupère les logins des étudiants à partir d'un fichier spécifié (configuration.txt) en utilisant la fonction « recupDonneeEtudiant » depuis le fichier « gestionConfig ».

Elle utilise ensuite la fonction « mainsujet2 » du fichier « sujet2 » pour obtenir des données sur les interactions des étudiants qu'elle stocke dans la variable « graphdata ».

Après cela, elle organise ces données pour créer des listes d'interactions par tranche horaire pour chaque étudiant.

En utilisant « matplotlib.pyplot », elle crée un histogramme où chaque barre représente le nombre d'interactions pour chaque tranche horaire, pour chaque étudiant du groupe.

Pour finir, elle sauvegarde cet histogramme dans un fichier image « GraphLane.png ».

Si une erreur survient lors de ce processus, elle capture et affiche un message d'erreur.

mainLane (méthode « mainLane »)

La fonction « mainLane » récupère les logins des étudiants à partir d'un fichier spécifié (configuration.txt) en utilisant la fonction « recupDonneeEtudiant » depuis le fichier « gestionConfig ».

En utilisant ensuite la fonction « mainsujet2 » du fichier « sujet2 », elle obtient des observations sur les hits des étudiants à partir du fichier spécifié par le paramètre « src ».

Après cela, elle appelle la fonction « sujet2Graph », générant ainsi un histogramme des hits des étudiants par tranche horaire et l'enregistrant dans « GraphLane.png ».

Pour finir, modifie le contenu de la page web « lane.html » en insérant des titres, des observations sous forme de paragraphes pour chaque étudiant, et l'histogramme.

Pendant le processus, elle affiche les observations des hits pour chaque étudiant dans la console.

Si une erreur survient lors de ce processus, elle capture et affiche un message d'erreur.

sujet2 (méthode « mainsujet2 »)

La fonction « mainsujet2 » prend en paramètres le chemin (src) vers un fichier CSV, une liste de logins (logins), et une chaîne de caractères (change) pour spécifier le type de traitement.

Elle initialise trois dictionnaires (result1, result2, result3) avec les logins fournis, servant à stocker les résultats du calcul des hits.

Elle essaie ensuite d'ouvrir le fichier CSV spécifié et de le lire ligne par ligne en utilisant « csv.reader() ». Les données pertinentes sont extraites en fonction de certaines conditions (catégorie, heures, etc.) et des opérations sont effectuées sur les hits en fonction de ces conditions et du paramètre « change ».

Après cela, les hits sont calculés et stockés dans les dictionnaires (result1, result2, result3) en fonction du cas spécifié par le paramètre « change »

Enfin, selon le cas, la fonction retourne le résultat final calculé sous forme d'une liste ou d'un dictionnaire et celui change aussi selon le cas.

Si une erreur survient lors du processus, elle capture et affiche un message d'erreur.

diagrammeRoux (méthode « get »)

La fonction get prend deux arguments de type string en paramètres :

« cheminData » qui dirige vers un répertoire qui contient les fichiers de configuration

« src » qui dirige vers un répertoire qui contient les fichiers .csv

La fonction teste si elle peut ouvrir le fichier csv et stocker dans un dictionnaire la consommation industrielle de gaz en fonction des communes correspondantes à cheminData. Si aucune erreur, est rencontrée, alors seulement elle exécute le code.

diagrammeRoux (méthode « histo9 »)

La procédure histo9 prend deux arguments de type string en paramètres :

« cheminData » qui dirige vers un répertoire qui contient les fichiers de configuration

« src » qui dirige vers un répertoire qui contient les fichiers .csv

A partir des données récoltées de la fonction précédente get, la procédure teste si elle peut construire un histogramme avec la librairie matplotlib et le sauvegarder dans le répertoire medias. Si elle peut, elle exécute le code.

mainRoux (méthode « mainRoux »)

La procédure mainRoux prend deux arguments de type string en paramètres :

« cheminData » qui dirige vers un répertoire qui contient les fichiers de configuration

« src » qui dirige vers un répertoire qui contient les fichiers .csv

Le mainRoux modifie la page roux.html, en ajoutant les données récupérées concernant la consommation résidentielle d'électricité, et d'autre part les données concernant la consommation de gaz sous forme d'histogramme précédemment sauvegardé dans médias.

sujet9 (méthode « mainsujet9 »)

La fonction mainsujet9 prend deux arguments de type string en paramètres

cheminData qui dirige vers un répertoire qui contient les fichiers de configuration

src qui dirige vers un répertoire qui contient les fichiers .csv

La fonction test si elle peut ouvrir le fichier csv et stocker dans un dictionnaire la consommation résidentielle d'électricité en fonction des communes correspondantes à cheminData. Si aucune erreur, est rencontrée, alors seulement elle exécute le code.

diagrammeHauersperger (méthode « sujet18Part2 »)

Cette méthode permet d'extraire l'indicateur de loyer en fonction du nombre de commune pour chaque commune de naissance des étudiants du groupe. Elle prend pour paramètre le chemin vers le fichier CSV contenant les données ainsi que le chemin vers le répertoire « data ». La fonction va tout

d'abord récupérer les codes postaux des communes de naissance de chacun des membres du groupe grâce à la fonction « `recuDonnéeEtudiant` » (décrite ci-dessus) puis va lire le fichier csv grâce à « `with open` » et « `csv.reader` » en relevant l'indicateur de loyer pour chaque commune commençant par le même code départemental de chaque commune de naissance des membres du groupe. Les données sont stockées dans un dictionnaire qui est trié avant d'être rendu.

diagrammeHauersperger (méthode « `sujet18Graph` »)

Cette méthode se contente d'appeler la méthode « `sujet18Part2` » avec les bons chemins pour récupérer les informations avant d'en faire plusieurs graphiques grâce à `matplotlib`.

mainHauersperger (méthode « `mainHauersperger` »)

Cette méthode va venir exécuter les autres fonctions « `hauersperger` », récupérer les données, et les intégrer à la page web « `hauersperger.html` ». Elle prend en paramètre deux chemins, le premier vers le fichier CSV contenant les informations à extraire, le deuxième, le chemin vers le répertoire « `opendata` ». La méthode va appeler la procédure « `modifierWeb` » pour venir modifier la page web « `hauersperger.html` ».

sujet18 (méthode « `main sujet18` »)

Cette fonction permet d'extraire d'un fichier csv, le nombre d'observations de loyers en 2022 dans les communes de naissance des membres du groupe. Elle prend en paramètre le chemin vers le fichier CSV où extraire les données ainsi que le chemin vers le fichier `opendata`. La fonction va tout d'abord récupérer les codes INSEE et les noms des communes de naissance de chacun des membres du groupe grâce à la fonction « `recuDonnéeEtudiant` » (décrite ci-dessus) puis va lire le fichier csv grâce à « `with open` » et « `csv.reader` » en relevant le nombre d'observations de loyers dans les communes de naissance des étudiants. Les données sont stockées dans un dictionnaire qui est ensuite rendu.

diagrammeSoleilhac (méthode « `diagramme` »)

Cette procédure permet d'extraire d'un fichier csv, le nombre total d'élève par niveau (CP, CE1, CE2...) pour chaque commune de naissance des étudiants du groupe. Elle va appeler la fonction « `main sujet17` » (que j'expliquerai plus tard) avec les noms de chaque niveau, ce qui va permettre d'avoir le nombre d'élève par niveau. Les résultats sont stockés dans des listes. La procédure attribue ensuite une couleur par niveau, créer une légende (`plt.legend`), attribue les noms de niveaux aux abscisses et les nombres d'élèves en ordonnées. Le diagramme est enfin sauvegardé dans le répertoire médias de `www`.

mainSoleilhac (méthode « `mainSoleilhac` »)

Cette méthode va venir exécuter les autres fonctions « `soleilhac` », récupérer les données, et les intégrer à la page web « `soleilhac.html` ». Elle prend en paramètre deux chemins, le premier vers le fichier CSV contenant les informations à extraire, le deuxième, le chemin vers le répertoire « `opendata` ». La méthode va appeler la procédure « `modifierWeb` » pour venir modifier la page web « `soleilhac.html` ».

sujet17 (méthode « `mainSoleilhac` »)

Cette fonction permet d'extraire du fichier csv le nombre d'élève dans le niveau spécifié, dans les communes de naissance des étudiants du groupe. Elle prend en paramètre le chemin vers le fichier CSV contenant les informations ainsi que la liste des communes de naissance et le niveau sur lequel

effectuer l'étude. Selon le niveau spécifié le code va chercher le nom du niveau ainsi que la liste des communes dans le fichier CSV. Les résultats sont stockés dans un dictionnaire qui est ensuite rendu.

Main

Ce script ne contient pas de fonctions, il prend en paramètres le chemin vers le fichier data et le fichier open data. Le « main » va d'abord importer toutes les méthodes depuis les différents fichiers. Il va ensuite récupérer la liste des noms, des prénoms, des sujets, et les chemins vers les fichiers CSV associés aux sujets grâce aux différentes fonctions décrites ci-dessus utilisant le fichier de configuration.txt. La fonction exécute ensuite dynamiquement, selon les listes récoltées, les fonctions « clonnerPageWeb », « supprimerWeb », « pageMoodle », « supprimerFichierOpenData » ainsi que chacun des « mainNom ». Ceci va générer les pages web de chacun des étudiants et les modifier. Les fichiers ZIP contenant les fichier CSV sont dézippé, utilisés et supprimés. Au passage la version courante du site est sauvegardée et une nouvelle version est mise à jour.

mainMoodle (méthode « pageMoodle »)

Cette méthode va venir exécuter les autres fonctions « moodle », récupérer les données, et les intégrer à la page web « index.html ». Elle prend en paramètre deux chemins, le premier, le chemin vers le fichier CSV contenant les informations à extraire, le deuxième, le chemin vers le répertoire « opendata ». La méthode va appeler la procédure « modifierWeb » pour venir modifier la page web « index.html ».