

Voici deux propositions de sujets TP au choix pratique. Chaque sujet doit être réalisé en groupe de 2 ou 3 étudiants et doit être développé en mode console en Python (sans utilisation de fonctions avancées ni imports externes, hormis les modules built-in). Les sujets intègrent des notions de structures de contrôle, boucles, collections et gestion des cas particuliers et vous pouvez persister les données avec les fichiers (texte, json ou csv).

Sujet 1 : Digitalisation de la Vente de Tickets et Cartes d'Abonnement pour SenTrans

Contexte :

SenTrans est une entreprise de transport urbain opérant dans la région de Dakar. Le territoire couvert est divisé en trois zones :

- **Zone 1**
 - *Keur Massar* : Ticket à 250 F
 - *Yumbeule* : Ticket à 300 F
 - *APIX* : Ticket à 350 F
 - **Abonnement mensuel : 10 000 F**

- **Zone 2**

- *Parcelle* : Ticket à 200 F
- *Dior* : Ticket à 150 F
- *Soprim* : Ticket à 250 F
- *Pate d'oie* : Ticket à 350 F
- *Gard yoff* : Ticket à 400 F
- **Abonnement mensuel : 13 000 F**

- **Zone 3**
 - *Ngor* : Ticket à 200 F
 - *Ouakam* : Ticket à 500 F
 - *Yoff* : Ticket à 250 F
 - *Foire* : Ticket à 300 F
 - *Coloban* : Ticket à 450 F
 - *UCAD* : Ticket à 400 F
 - **Abonnement mensuel : 16 000 F**

Travail demandé :

Développer un programme Python en mode console qui permet de digitaliser la vente des tickets et des cartes d'abonnement. Le programme devra :

1. **Proposer un menu interactif** demandant à l'utilisateur de choisir entre :

- Acheter un ticket
- Acheter une carte d'abonnement
- Quitter

2. Pour l'achat d'un ticket :

- Demander la destination (le nom d'une ligne).
- Afficher le prix du ticket correspondant à la destination.
- Demander à l'utilisateur de saisir le montant qu'il paie (le programme ne doit pas accepter un billet de 2000 F ou plus).
- Calculer et afficher la monnaie à rendre.
- Afficher un récapitulatif du ticket (destination, prix payé, monnaie rendue).

3. Pour l'achat d'une carte d'abonnement :

- Demander la zone (zone1, zone2 ou zone3).
- Afficher le prix de l'abonnement pour la zone choisie.
- Demander à l'utilisateur de saisir le montant versé.
- Calculer et afficher la monnaie à rendre.
- Afficher le récapitulatif de la carte d'abonnement (zone, prix, montant versé, monnaie rendue).

4. Gestion des cas particuliers :

- Si l'utilisateur saisit une destination ou une zone inconnue, afficher un message d'erreur.
- Si le montant saisi est insuffisant ou supérieur à 2000 F, afficher un message d'erreur.
- Répéter le menu tant que l'utilisateur ne choisit pas de quitter.

Objectifs pédagogiques :

- Utilisation de structures conditionnelles (if, else) et de boucles (while, for).
- Manipulation de listes et de dictionnaires pour stocker les données (lignes, zones, tarifs).
- Validation des entrées utilisateur et gestion des cas d'erreur.

Sujet 2 : Système de Gestion des Réservations de Salles pour l'ISI

Contexte :

L'Institut Supérieur d'Informatique (ISI) souhaite digitaliser la gestion de ses réservations de salles pour faciliter l'organisation des travaux de groupe, des réunions et des séminaires. La gestion sera divisée en plusieurs catégories de salles (par exemple, salle de cours, salle de réunion, amphithéâtre) avec des capacités et des équipements différents.

Travail demandé :

Développer un programme Python en mode console qui permet de gérer les réservations de salles. Le programme devra :

1. **Proposer un menu interactif** permettant à l'utilisateur de choisir parmi plusieurs actions :
 - Réserver une salle
 - Annuler une réservation
 - Afficher toutes les réservations
 - Consulter les disponibilités d'une salle
 - Quitter

2. Pour réserver une salle :

- Demander à l'utilisateur de saisir le type de salle (ex. "cours", "réunion", "amphithéâtre").
- Demander la date et l'heure de réservation.
- Vérifier que la salle est disponible pour le créneau demandé (vous pouvez simuler la disponibilité avec des données statiques ou des règles simples).
- Enregistrer la réservation dans une structure de données (par exemple, une liste ou un dictionnaire).
- Afficher une confirmation de la réservation avec un numéro de réservation généré automatiquement.

3. Pour annuler une réservation :

- Demander le numéro de réservation.
- Vérifier si la réservation existe.
- Supprimer la réservation et afficher un message de confirmation ou d'erreur.

4. Pour afficher toutes les réservations :

- Parcourir la liste des réservations et afficher pour chacune le type de salle, la date, l'heure et le numéro de réservation.

5. Pour consulter la disponibilité d'une salle :

- Demander le type de salle et la date.
- Afficher si la salle est disponible ou occupée pour ce créneau.

Objectifs pédagogiques :

- Utilisation de structures de contrôle et de boucles pour la navigation dans un menu interactif.
- Utilisation de listes et dictionnaires pour stocker et gérer les réservations.
- Génération de numéros uniques et gestion des erreurs (par exemple, réservation inexistante).

NB : Ces sujets de TP pratique visent à renforcer vos compétences en programmation Python tout en abordant des problématiques concrètes. Ils vous offriront une opportunité de travailler en collaboration et de mettre en œuvre vos connaissances sur les structures de contrôle, les collections et la persistance des données. Nous encourageons chaque groupe à donner le meilleur de soi et à innover dans les solutions proposées.

Date de retour des TP : le 7 mars 2025.

Nous avons hâte de découvrir vos travaux et soyez vous-même! 😊