

TP 2 - Go

Les TP de ce cours peuvent être effectués sous linux/windows/macOS avec l'IDE que vous souhaitez.

Exercice 1 : Communes

Ci-dessous vous sont données 10 communes Françaises ainsi que leur nombre d'habitants.

L'Abergement-Clémenciat: 832
Les Aires: 608
Marseille-en-Bauvaisis: 1443
Martincourt: 135
Saint-Laurent-Bretagne: 445
Alès: 43892
Affoux: 395
Aigueperse: 243
Aguts: 236
Aguts: 236
Aiguefonde: 2510

1. Créez un programme qui contiendra une **map** qui associera son nombre d'habitants à chaque commune et permettra à l'utilisateur de demander le nombre d'habitants d'une commune.
2. Implémentez la possibilité pour un utilisateur d'insérer une commune de son choix.
3. Ajoutez la possibilité pour un utilisateur de supprimer une commune.
4. Modifiez votre programme afin qu'il permette à l'utilisateur de renseigner le nom de deux communes et affiche laquelle est la plus peuplée.

Exercice 2 : Régions

Les communes ci-dessus sont dans les régions suivantes (4 régions):

L'Abergement-Clémenciat: Auvergne-Rhône-Alpes
Les Aires: Occitanie
Marseille-en-Bauvaisis: Hauts-de-France
Martincourt: Hauts-de-France
Saint-Laurent-Bretagne: Nouvelle-Aquitaine
Alès: Occitanie
Affoux: Auvergne-Rhône-Alpes
Aigueperse: Auvergne-Rhône-Alpes
Aguts: Occitanie
Aguts: Occitanie
Aiguefonde: Occitanie

Gardez la **map** de l'exercice précédent et créez une autre **map** qui associera à chaque région une liste de communes.

1. Modifiez votre programme afin qu'il permette à l'utilisateur de sélectionner une région et affiche le nombre de communes dans cette région.
2. Affichez aussi le nom de la commune la plus peuplée de la région.
3. Affichez aussi le nombre total d'habitants dans la région.

Exercice 3 : Csv

Il vous est donné un fichier `communes.csv` contenant des informations pour les communes précédentes ainsi qu'un fichier `lire_communes.go` qui charge ce fichier dans une chaîne de caractères.

Chaque ligne du fichier csv contient des cases, séparées par des virgules.

1. Créez une fonction qui accepte en argument un numéro de colonne ainsi qu'un numéro de ligne et affiche le contenu de la case correspondante.

Vérifiez que vous trouvez bien "nom_standard" pour ligne 0 et colonne 3.

2. Créez une fonction qui accepte une colonne et une ligne et affiche la valeur de la case à la ligne 0 ainsi qu'à la ligne demandée.

Par exemple si on demande la colonne 11 et la ligne 2 on affichera "reg_nom" (colonne 11 ligne 0) et "Occitanie" (colonne 11 ligne 2).

3. Modifiez votre programme afin qu'il appelle la fonction précédente, pour les lignes 1 à 11 et les colonnes 2, 11 et 18.

Vous devriez retrouver les informations des exercices précédents.

Exercice 4 : Tous ensembles

1. Reprenez le code des exercices précédents et combinez les afin de remplir les `map` des régions et des communes automatiquement.
2. Téléchargez le fichier suivant (<https://www.data.gouv.fr/api/1/datasets/r/f5df602b-3800-44d7-b2df-fa40a0350325>) et essayez votre code avec ce fichier.