

## Exercices simples sur les dates et les chaînes de caractères

Écrire une fonction qui prend une date de naissance au format YYYY-MM-DD en entrée et renvoie l'âge actuel de la personne.	
Demandez à l'utilisateur de saisir deux dates au format DD-MM-YYYY et calculez la différence en jours entre les deux dates.	
Écrire une fonction qui prend une date et un nombre de jours comme paramètres et renvoie la date après avoir ajouté le nombre de jours spécifié.	
Demander à l'utilisateur de saisir une date au format YYYY-MM-DD et afficher quel jour de la semaine correspond à cette date.	
Écrire un programme qui prend la date actuelle et calcule le nombre de jours restants jusqu'à la fin de l'année.	
Créer une fonction qui prend une date au format DD-MM-YYYY et la convertit en format YYYY/MM/DD.	
Écrire une fonction qui prend une chaîne de caractères et vérifie si elle correspond à un format de date valide (YYYY-MM-DD). Si la date est valide, affichez-la sous forme de date Python, sinon indiquez que la date est invalide.	
Écrire une fonction qui prend deux dates en entrée et calcule le nombre de jours ouvrés (lundi à vendredi) entre ces deux dates.	
Créer une fonction qui prend un mois et une année en entrée, puis affiche toutes les dates de ce mois.	
Écrire une fonction qui prend deux heures au format HH:MM et calcule la durée écoulée en heures et minutes entre ces deux heures.	
Écrire un programme qui génère une liste de 10 dates aléatoires dans une plage spécifiée (par exemple, entre 01-01-2000 et 31-12-2023).	
Créer une fonction qui prend un mois et une année et affiche toutes les dates de weekend (samedi et dimanche) de ce mois.	
Écrire un programme qui prend la date actuelle et renvoie le nombre de jours fériés restants dans l'année (en utilisant une liste fixe de jours fériés pour votre pays).	
Écrivez une fonction is_palindrome(s) qui prend une chaîne de caractères s et retourne True si la chaîne est un palindrome (c'est-à-dire qu'elle se lit de la même manière de gauche à droite et de droite à gauche, en ignorant les espaces, la casse et la ponctuation), et False sinon.	
Créez une fonction count_vowels_consonants(s) qui prend une chaîne s et renvoie un dictionnaire contenant le nombre de voyelles et de consonnes.	
Écrivez une fonction longest_word(sentence) qui prend une chaîne sentence et renvoie le mot le plus long dans la chaîne. En cas d'égalité, renvoyez le premier mot le plus long trouvé.	
Écrivez une fonction replace_word(sentence, old_word, new_word) qui remplace toutes les occurrences de old_word par new_word dans une phrase sentence, en respectant la casse.	
Créez une fonction reverse_words(sentence) qui prend une chaîne	

<p>sentence et renvoie la phrase avec l'ordre des mots inversé, tout en gardant chaque mot dans sa forme originale.</p>	
<p>Écrivez une fonction <code>word_frequency(sentence)</code> qui prend une phrase <code>sentence</code> et retourne un dictionnaire contenant la fréquence de chaque mot (insensible à la casse). Ignorez la ponctuation.</p>	
<p>Créez une fonction <code>find_anagrams(word, word_list)</code> qui prend un mot <code>word</code> et une liste de mots <code>word_list</code>, et retourne une liste de tous les mots de la liste qui sont des anagrammes de <code>word</code>.</p>	
<p>Écrivez une fonction <code>clean_string(s)</code> qui prend une chaîne <code>s</code> et retourne une nouvelle chaîne avec toutes les ponctuations enlevées et les espaces multiples remplacés par un seul espace.</p>	
<p>Écrivez une fonction <code>acronym(phrase)</code> qui prend une phrase et renvoie l'acronyme correspondant en majuscules.</p> <p>Exemple :</p> <p><code>acronym("As Soon As Possible") # "ASAP"</code></p> <p><code>acronym("National Aeronautics and Space Administration") # "NASA"</code></p>	
<p>Écrivez une fonction <code>compter_mots(phrase: str) -&gt; int</code> qui prend une phrase en entrée et retourne le nombre de mots qu'elle contient. Les mots sont séparés par des espaces.</p>	
<p>Écrivez une fonction <code>mot_le_plus_long(phrase: str) -&gt; str</code> qui prend une phrase en entrée et retourne le mot le plus long de cette phrase. S'il y a plusieurs mots avec la même longueur maximale, retournez le premier rencontré.</p>	
<p>Écrivez une fonction <code>est_palindrome(mot: str) -&gt; bool</code> qui vérifie si un mot est un palindrome (se lit de la même façon à l'envers).</p>	
<p>Écrivez une fonction <code>compter_voyelles_consonnes(mot: str) -&gt; dict</code> qui prend un mot et retourne un dictionnaire avec le nombre de voyelles et de consonnes.</p>	
<p>Écrivez une fonction <code>frequence_caracteres(chaine: str) -&gt; dict</code> qui prend une chaîne de caractères et retourne un dictionnaire contenant chaque caractère et sa fréquence d'apparition.</p>	
<p>Écrivez une fonction <code>remplacer_voyelles(chaine: str, symbole: str) -&gt; str</code> qui remplace toutes les voyelles d'une chaîne par un symbole donné.</p>	
<p>Écrivez une fonction <code>extraire_chiffres(chaine: str) -&gt; str</code> qui prend une chaîne de caractères et retourne une chaîne composée uniquement des chiffres présents dans la chaîne d'origine.</p>	

### Exercice 1 : Créer une fonction qui analyse un texte donné.

- **Question** : Écrivez une fonction `analyze_text(text)` qui prend en paramètre une chaîne de caractères `text`. La fonction doit retourner un dictionnaire contenant :
  - Le nombre total de mots dans le texte.
  - Le nombre de mots uniques.
  - Le mot le plus fréquent et son nombre d'occurrences.
  - Le mot le plus long et sa longueur.

### Exercice 2 : Comparer deux dates et afficher la différence en années, mois et jours.

- **Question** : Écrivez une fonction `date_difference(date1, date2)` qui prend en paramètre deux chaînes de caractères `date1` et `date2` représentant des dates au format "YYYY-MM-DD". La fonction doit retourner une chaîne de caractères indiquant la différence en années, mois et jours.
- **Indice** : Utilisez le module `datetime` pour convertir les chaînes en objets `date` et calculez la différence.

### Exercice 3 : Manipuler les dates et les chaînes de caractères pour vérifier si une date est un palindrome.

- **Question** : Écrivez une fonction `is_date_palindrome(date)` qui prend une date au format "DD-MM-YYYY" en paramètre et vérifie si la date est un palindrome (c'est-à-dire si elle se lit de la même manière à l'envers, sans tenir compte des tirets). Retournez `True` ou `False`.
- **Indice** : Utilisez la méthode `.replace()` pour supprimer les tirets et `.reverse()` pour vérifier si la date est un palindrome.

### Exercice 4 : Chiffrement de César

Implémentez un **chiffrement de César**. Il s'agit d'un simple chiffrement par substitution où chaque lettre d'un texte est décalée d'un certain nombre de positions dans l'alphabet.

#### Instructions :

1. Écrivez une fonction `encrypt_caesar(text, shift)` qui prend un texte `text` et un entier `shift` pour décaler chaque lettre.
2. Écrivez la fonction inverse `decrypt_caesar(cipher_text, shift)` pour déchiffrer le texte.
3. Gérez les majuscules et minuscules, mais ignorez les caractères non alphabétiques.

### Exercice 5 : Chiffrement par substitution monoalphabétique

Implémentez un **chiffrement par substitution monoalphabétique**. Dans ce chiffrement, chaque lettre de l'alphabet est remplacée par une autre lettre. Vous pouvez générer une clé aléatoire pour la substitution.

#### Instructions :

1. Créez une fonction `generate_key()` pour générer une clé de substitution aléatoire.
2. Implémentez une fonction `encrypt_substitution(text, key)` pour chiffrer le texte en utilisant la clé de substitution.
3. Implémentez une fonction `decrypt_substitution(cipher_text, key)` pour déchiffrer le texte.

### Exercice 6 : Chiffrement de Vigenère

Implémentez un **chiffrement de Vigenère**, une méthode de chiffrement polyalphabétique.

#### Instructions :

1. Écrivez une fonction `encrypt_vigenere(text, keyword)` où chaque lettre du texte est décalée selon la lettre correspondante dans le mot-clé `keyword`.
2. Écrivez une fonction `decrypt_vigenere(cipher_text, keyword)` pour déchiffrer le texte.

### Exercice 7 : Chiffrement symétrique avec Fernet (cryptography)

Utilisez la bibliothèque `cryptography` pour implémenter un **chiffrement symétrique** avec l'algorithme **Fernet**.

#### Instructions :

1. Installez la bibliothèque `cryptography` (`pip install cryptography`).
2. Implémentez une fonction `encrypt_message(message, key)` pour chiffrer un message.
3. Implémentez une fonction `decrypt_message(cipher_text, key)` pour déchiffrer le message.