

 créer un devoir dupliquer

Lien sujet + copie à fournir aux élèves pour un entraînement en autonomie

 www.codepuzzle.io/K31LGI

[PREMIERE_NSI_2026] BASES DU LANGAGE PYTHON - SUJET A

EXERCICE 1 - PLACES DE CINÉMA

Un cinéma propose plusieurs tarifs en fonction du nombre de places achetées. Voici la grille tarifaire :

- Pour 1 à 5 places achetées : **6 € par place**
- Pour 6 à 15 places achetées : **5 € par place**
- Pour plus de 15 places achetées : **4 € par place**

QUESTION 1

Compléter le programme afin d'afficher la somme totale à payer. Le nombre de places achetées est stocké dans la variable `nb_places`.

Le programme doit afficher un message de la forme :

```
La somme totale à payer est de 4 €
```

EXERCICE 2 - LISTE DES DIVISEURS

On s'intéresse ici à l'écriture d'un programme permettant d'obtenir la liste des diviseurs d'un nombre.

QUESTION 1

Écrire un programme qui affiche les diviseurs du nombre stocké dans la variable `nombre`.

- Il faudra utiliser une boucle `for`.
- Il faudra utiliser une structure conditionnelle `if`.

Pour `nombre = 50`, le programme doit afficher le message suivant :

```
Liste des diviseurs de 50 entre 1 et 50 :  
1 2 5 10 25 50
```

Pour afficher les diviseurs côte à côte, sur la même ligne, on peut utiliser l'instruction `print(valeur, end=" ")`.

EXERCICE 3 - ÉTAT DE L'EAU

La fonction `etat_eau` permet de déterminer l'état de l'eau en fonction de sa température. Elle prend en paramètre un nombre entier `temperature`, qui représente la température de l'eau en °C, et renvoie une chaîne de caractères.

QUESTION 1

Compléter la fonction `etat_eau` afin qu'elle renvoie :

- `"solide"` si la température est négative ;
- `"liquide"` si la température est comprise entre 0 °C et 100 °C inclus ;
- `"gazeux"` si la température est supérieure à 100 °C.

QUESTION 2

Ajouter les instructions permettant d'afficher l'état de l'eau pour les températures **-8 °C, 17 °C et 103 °C**.

EXERCICE 4 - NOMBRE IMPAIR

La fonction `est_impair` permet de déterminer si un nombre est impair. Elle prend en paramètre un nombre entier `nombre` et renvoie `True` si le nombre est impair et `False` sinon.

QUESTION 1

Compléter la fonction `est_impair`. On rappelle qu'un nombre est impair si le reste de sa division par 2 est égal à 1.

QUESTION 2

Ajouter les instructions permettant d'afficher si les nombres **4, 7 et 9** sont impairs.

EXERCICE 5 - CALCULS AUTOUR D'UN RECTANGLE

La fonction `calcul_rectangle` permet de calculer l'aire ou le périmètre d'un rectangle à partir de la longueur de ses côtés. Elle prend en paramètres deux nombres positifs `a` et `b`, ainsi qu'une chaîne de caractères `operation`. Elle renvoie l'aire ou le périmètre du rectangle en fonction de l'opération demandée.

QUESTION 1

Compléter la fonction `calcul_rectangle` afin qu'elle permette de calculer l'aire ou le périmètre du rectangle.

QUESTION 2

Compléter la fonction afin d'ajouter une assertion permettant de contrôler le paramètre `operation`. Celui-ci ne peut prendre que deux valeurs :

- `"aire"` → la fonction renvoie l'aire du rectangle ;
- `"perimetre"` → la fonction renvoie le périmètre du rectangle.

QUESTION 3

Ajouter les instructions permettant d'afficher l'aire et le périmètre d'un rectangle de **longueur 4** et de **largeur 6**.

EXERCICE 6 - ANALYSE D'UNE CHAÎNE DE CARACTÈRES

Dans cet exercice, on considère la chaîne de caractères suivante :

```
texte = "Je Vais Réussir l'Évaluation en NSI."
```

QUESTION 1

Écrire la fonction `nombre_caracteres_total`. Elle prend en paramètre une chaîne de caractères `chaîne` et renvoie le nombre total de caractères de cette chaîne.

- On vérifiera que le type du paramètre `chaîne` est bien une chaîne de caractères.

QUESTION 2

Écrire la fonction `nombre_majuscules`. Elle prend en paramètre une chaîne de caractères `chaîne` et renvoie le nombre de lettres majuscules présentes dans cette chaîne.

- On vérifiera que le type du paramètre `chaîne` est bien une chaîne de caractères.
- On utilisera une boucle pour parcourir chaque caractère de la chaîne.
- On utilisera une condition pour vérifier si le caractère est une majuscule.
- Pour savoir si un caractère est une majuscule, on peut utiliser la méthode `isupper()`.

Voici un exemple :

```
>>> a = 'B'
>>> a.isupper()
True
>>> b = 'j'
>>> b.isupper()
False
```

QUESTION 3

Tester vos fonction sur la variable `texte`.

PORTÉE D'EXÉCUTION DU CODE : exécution par cellule (chaque cellule s'exécute indépendamment des autres)

PROGRAMME 1

```
CODE ÉLÈVE

1  #===== EXERCICE 1 =====#
2  nb_places = 8
3
4  #===== FIN EXERCICE 1 =====#

CODE ENSEIGNANT

1
```

SOLUTION POSSIBLE

1

PROGRAMME 2

CODE ÉLÈVE

```
1  #===== EXERCICE 2 =====#
2  nombre = 50
3
4  #===== FIN EXERCICE 2 =====#
```

CODE ENSEIGNANT

1

SOLUTION POSSIBLE

1

PROGRAMME 3

CODE ÉLÈVE

```
1  #===== EXERCICE 3 =====#
2  # Définition de la fonction ici
3  def etat_eau(temperature:float) -> str :
4      '''
5      Renvoie une chaine de caractères en fonction de la valeur du paramètre temperature
6      '''
7      # À compléter
8
9
10 # Utilisation et test de la fonction ici
11
12
13
14 #===== FIN EXERCICE 3 =====#
```

CODE ENSEIGNANT

1

SOLUTION POSSIBLE

1

PROGRAMME 4

CODE ÉLÈVE

```
1  #===== EXERCICE 4 =====#
2  # Définition de la fonction ici
3  def est_impair(nombre:int) -> bool:
4      # À compléter
5
6
7  # Utilisation et test de la fonction ici
8
9
10 #===== FIN EXERCICE 4 =====#
```

CODE ENSEIGNANT

1

SOLUTION POSSIBLE

1

PROGRAMME 5

CODE ÉLÈVE

```
1  #===== EXERCICE 5 =====#
2  # Définition de la fonction ici
3  def calcul_rectangle(a:int, b:int, operation:str) -> float:
4      # À compléter
5
6
7  # Utilisation et test de la fonction ici
8
9
10 #===== FIN EXERCICE 5 =====#
```

CODE ENSEIGNANT

1

SOLUTION POSSIBLE

1

PROGRAMME 6

CODE ÉLÈVE

```
1  #===== EXERCICE 6 =====#
2  # Définition des fonctions ici
3
4
5  # Utilisation et test de la fonction ici
6
7
8  #===== FIN EXERCICE 6 =====#
```

CODE ENSEIGNANT

1

SOLUTION POSSIBLE

1