

Python: Les fonctions (pratique autonome)

En Python, comme dans d'autres langages de programmation, une fonction est une suite d'instructions que l'on peut appeler avec un nom. C'est un moyen de regrouper des opérations pour les réutiliser efficacement. Cela permet de rendre ton code plus clair, plus organisé et plus facile à maintenir.

5GMS

 Exploration

Envoyez ces exercices dans le dossier [Python / Exercices / Fonctions / 0 - Pratique autonome](#).

Exercice 11: Conversion de miles en kilomètres

Objectif: Pratiquer la conversion simple.

Énoncé: Crée une fonction `miles_en_kilometres(miles)` qui convertit les miles en kilomètres en utilisant la formule 1 mile = 1.60934 kilomètres.

Exemple:

```
1 | print(miles_en_kilometres(5))
```

Sortie attendue:

8.0467

Nom du fichier: `fonctions-11-miles-[prenom].py`

Exercice 12: Température de surface d'une étoile

Objectif: Utiliser la loi de Wien pour des calculs astronomiques.

Énoncé: La loi de Wien relie la température d'une étoile à la longueur d'onde à laquelle elle brille le plus. Écris une fonction `temperature_etoile(longueur_max)` qui calcule la température de surface approximative de l'étoile en Kelvin. Utilise la formule $T = \frac{2.898 * 10^6}{longueur_{max}}$ où `longueur_max` est en micromètres

Exemple:

```
1 | print(temperature_etoile(0.5))
```

Sortie attendue:

5796.0 K

Nom du fichier: fonctions-12-etoile-[prenom].py

Exercice 13: Conversion de radians en degrés

Objectif: Travailler avec des conversions d'unités angulaires.

Énoncé: Définis une fonction `radians_en_degrees(radians)` qui convertit une mesure en radians en degrés en utilisant la formule $degrés = radians * \frac{180}{\pi}$.

Exemple:

```
1 | import math
2 | print(radians_en_degrees(math.pi/4))
```

Sortie attendue:

45.0

Nom du fichier: fonctions-13-radians-[prenom].py

Exercice 14: Calcul de la force gravitationnelle

Objectif: Appliquer la loi de la gravitation universelle de Newton.

Énoncé: Écris une fonction `force_gravitationnelle(m1, m2, r)` qui calcule la force gravitationnelle entre deux masses (en kg) à une certaine distance (en mètres). Utilise la formule $F = G \frac{m1 * m2}{r^2}$, où **G** est la constante gravitationnelle $6.674 * 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$.

Exemple:

```
1 | print(force_gravitationnelle(5.98e24, 1.99e30, 1.50e11)) # Terre - Soleil
```

Sortie attendue:

3.5277481777777787e+22

Nom du fichier: fonctions-14-radians-[prenom].py

Exercice 15: Conversion de pression en Pascal

Objectif: Pratiquer les conversions d'unités de pression.

Énoncé: Crée une fonction `atm_en_pascal(atm)` qui convertit la pression atmosphérique de atmosphères en Pascal. Utilise la formule $1 \text{ atm} = 101325 \text{ Pa}$.

Exemple:

```
1 | print(atm_en_pascal(1))
```

Sortie attendue:

101325

Nom du fichier: fonctions-15-pressure-[prenom].py

Exercice 16: Calcul du volume d'un cylindre

Objectif: Appliquer des formules géométriques simples.

Énoncé: Écris une fonction `volume_cylindre(rayon, hauteur)` qui calcule le volume d'un cylindre en utilisant la formule $V = \pi * rayon^2 * hauteur$.

Exemple:

```
1 | import math
2 | print(volume_cylindre(3, 5))
```

Sortie attendue:

Valeur du volume en unités cubiques

Nom du fichier: fonctions-16-cylindre-[prenom].py

Exercice 17: Conversion d'énergie cinétique

Objectif: Comprendre la relation entre différentes unités d'énergie.

Énoncé: Définis une fonction `joules_en_kilowattheures(joules)` qui convertit l'énergie en joules en kilowattheures, sachant que $1 \text{ kWh} = 3.6 * 10^6 \text{ joules}$.

Exemple:

```
1 | print(joules_en_kilowattheures(3600000))
```

Sortie attendue:

1.0

Nom du fichier: fonctions-17-joules-[prenom].py

Exercice 18: Conversion de décibels en intensité sonore

Objectif: Travailler avec des logarithmes dans des conversions.

Énoncé: Crée une fonction `decibels_en_intensite(decibels)` qui convertit des niveaux de décibels en intensité sonore en Watts par mètre carré en utilisant la formule $I = 10^{\frac{\text{decibels}}{10}} * 10^{-12}$.

Exemple:

```
1 | print(decibels_en_intensite(90))
```

Sortie attendue:

Valeur de l'intensité en W/m²

Nom du fichier: `fonctions-18-decibels-[prenom].py`

Exercice 19: Distance de freinage d'un robot

Objectif: Appliquer des concepts de physique de base à la robotique.

Énoncé: Écris une fonction `distance_freinage(vitesse, coefficient)` qui calcule la distance minimale nécessaire pour qu'un robot s'arrête complètement. Utilise la formule $distance = \frac{vitesse^2}{2 * coefficient * g}$, où *vitesse* est en m/s, *coefficient* est le coefficient de friction entre les roues et le sol, et *g* est l'accélération due à la gravité 9.81m/s².

Exemple:

```
1 | print(distance_freinage(10, 0.7))
```

Sortie attendue:

Valeur de la distance en mètres

Nom du fichier: `fonctions-19-freinage-[prenom].py`
