EXERCICES

Exercices sur les modules

S'entraîner à utiliser des modules dans des fonctions Python.

Exercice 1

Exercice 1

1. Écrire une fonction `racine(n: int) -> float` qui prend en paramètre un entier `n` et renvoie comme résultat la racine carrée de `n`.

Cette fonction devra utiliser le module `math`, qui aura été importé de manière précise.

```
PYTHON
>>> racine(3)
1.732050808
```

2. Tester la fonction `racine()` en l'appelant dans la console, puis l'écrire dans le cahier.

☐ Exercice 1 terminé.

Exercice 2

Exercice 2

1. Écrire une fonction `hypotenuse(a: int, b: int) -> float` qui prend en paramètres deux entiers et renvoie comme résultat la valeur de l'hypoténuse d'un triangle rectangle dont les deux autres côtés ont pour longueurs les valeurs des paramètres.

Cette fonction doit utiliser la fonction `racine()` définie précédemment.

```
PYTHON
>>> hypotenuse(5, 3)
5.84
```

2. Tester la fonction `hypotenuse()` en l'appelant dans la console, puis l'écrire dans le cahier.

☐ Exercice 2 terminé.

Exercice 3

Exercice 3

1. Écrire une fonction `des() -> int` qui ne prend aucun paramètre et renvoie comme résultat un chiffre aléatoire parmi `1`, `2`, `3`, `4`, `5` et `6`.

Cette fonction devra utiliser le module `random`, qui aura été importé de manière globale.

PYTHON

```
>>> des()
5
>>> des()
2
```

2. Tester la fonction `des()` en l'appelant dans la console, puis l'écrire dans le cahier.

□ Exercice 3 terminé.

Exercice 4

Exercice 4

1. Écrire une fonction `attendre(duree: int) -> None` qui attend pendant la durée donnée, sans renvoyer de valeur.
2. Tester la fonction `attendre()` en l'appelant dans la console, puis l'écrire dans le cahier.

□ Exercice 4 terminé.