PYTHON — LES FONCTIONS

DÉFINIR ET RÉUTILISER DU CODE AVEC def

1. À quoi ça sert ?

Une fonction regroupe des instructions pour une tâche.

- Réutiliser du code sans le recopier.
- Organiser le programme en petites tâches.
- Faciliter les modifications.

Tu connais déjà print(), input() et len().
Avec def, tu peux créer tes propres fonctions.

2. Définir une fonction

```
def dire_bonjour():
    print("Bonjour !")
```

- def annonce la définition de la fonction.
- dire_bonjour est son nom.
- () accueille les éventuels paramètres.
- : annonce le bloc, indenté de 4 espaces.

Définir une fonction ne l'exécute pas.

3. Appeler une fonction

Écrire son nom suivi de parenthèses :

```
dire_bonjour()
dire_bonjour()
```

Affiche Bonjour ! deux fois, sur deux lignes.

Chaque appel exécute les instructions de la fonction.

PYTHON — LES FONCTIONS

TRANSMETTRE DES INFORMATIONS

4. Les paramètres et les arguments

Un paramètre permet de recevoir une information.

```
def saluer(prenom):
    print("Bonjour", prenom)

saluer("Léa")
saluer("Adam")
```

À l'écran : Bonjour Léa, puis Bonjour Adam.

prenom est un paramètre de la fonction.
"Léa" est un argument : la valeur transmise.

Transmettre la valeur d'une variable

```
nom = input("Quel est ton prénom ? ")
saluer(nom)
```

5. Utiliser plusieurs paramètres

Séparer les paramètres par des virgules.

```
def presenter(prenom, age):
    print(prenom, "a", age, "ans.")

presenter("Léa", 16)
```

Résultat : Léa a 16 ans.

Les arguments correspondent aux paramètres dans l'ordre.
prenom reçoit "Léa" ; age reçoit 16.

PYTHON — LES FONCTIONS

RENVOYER ET RÉUTILISER UN RÉSULTAT

6. Renvoyer une valeur avec return

```
def carre(nombre):
    return nombre * nombre

resultat = carre(5)
print(resultat)
```

Résultat : 25

1. carre(5) transmet la valeur 5 à nombre.
2. La fonction calcule 5 × 5.
3. return renvoie 25, stocké dans resultat.

return termine immédiatement la fonction.

7. print() et return : deux rôles différents

print(valeur) affiche une valeur à l'écran.

return valeur renvoie une valeur au programme.

```
def double(nombre):
    return nombre * 2

resultat = double(4)
print(resultat + 1)
```

Résultat : 9

Le résultat renvoyé peut servir dans un autre calcul.

Sans return, une fonction renvoie None.

Afficher avec print() ne suffit pas à renvoyer un résultat.

PYTHON — LES FONCTIONS

EXEMPLES ET POINTS À RETENIR

8. Exemple : un tarif selon l'âge

Tarifs fictifs de cinéma :

```
def calculer_tarif(age):
    if age < 12:
        return 5
    elif age < 18:
        return 7
    else:
        return 10

age_client = int(input("Ton âge ? "))
prix = calculer_tarif(age_client)
print("Le billet coûte", prix, "euros.")
```

À 15 ans, la fonction renvoie 7.

Conditions : 4 espaces ; return dans les blocs : 8 espaces.

9. Les variables locales

```
def additionner(a, b):
    total = a + b
    return total

somme = additionner(3, 4)
print(somme)  # affiche 7
```

total et les paramètres sont locaux à la fonction.
return transmet le résultat, récupéré ici dans somme.

À RETENIR • ÉVITER LES ERREURS

• Définir avec def, terminer par : et indenter le bloc.
• Appeler avec (), en fournissant les arguments attendus.
• Utiliser return pour réutiliser un résultat.

def définit • l'appel exécute • return renvoie