

# Les fonctions



## Définition d'une fonction :

- Blocs de code réutilisables qui effectuent une tâche spécifique.
- Permet de structurer le code, de le rendre plus lisible et maintenable.
- Une fonction est définie en utilisant le mot-clé **def**, suivi par un nom de fonction, puis des parenthèses () qui peuvent inclure des noms de paramètres séparés par des virgules. Après les parenthèses, le corps de la fonction est défini avec une **indentation**

# Fonction sans paramètre ni retour



- Une fonction simple qui n'attend aucun argument et ne retourne aucune valeur.

```
def alerter():  
    print("Intrusion détectée dans le réseau !")  
  
alerter()
```

# Fonction avec paramètre sans retour



- Fonction avec un paramètre: (adresse\_ip).

```
def alerter(adresse_ip):  
    print(f"Intrusion détectée depuis l'adresse IP : {adresse_ip}")  
  
alerter("192.168.1.10")
```

# Fonction avec paramètres multiples



- Fonction prenant plusieurs arguments. Ici, le nombre de paquets reçus et perdus

```
def analyser_trafic(paquets_recus, paquets_perdus):  
    print(f"Paquets reçus : {paquets_recus}, Paquets perdus : {paquets_perdus}")  
  
analyser_trafic(5000, 200)
```

# Fonction avec paramètres par défaut



- Les paramètres par défaut permettent de simplifier l'appel de fonction lorsque certaines valeurs sont récurrentes.

```
def analyser_trafic(paquets_recus=5000, paquets_perdus=200):  
    print(f"Paquets reçus : {paquets_recus}, Paquets perdus : {paquets_perdus}")  
  
analyser_trafic()  
analyser_trafic(7000)
```

# Fonction avec valeur de retour



- Retourne une valeur après un calcul, ici la durée totale de connexion

```
def calculer_temps_connexion(temps_initial, temps_final):  
    return temps_final - temps_initial  
  
duree_connexion = calculer_temps_connexion(10, 18)  
print(f"Durée de connexion : {duree_connexion} heures")
```



# Exemple d'utilisation

```
def analyse_trafic_reseau(paquets, seuil_anomalie):
    anomalies_detectees = 0
    for paquet in paquets:
        if paquet > seuil_anomalie:
            print(f"Anomalie détectée : Paquet {paquet} dépasse le seuil {seuil_anomalie}")
            anomalies_detectees += 1
        else:
            print(f"Paquet {paquet} est dans la limite normale.")

    print(f"Nombre total d'anomalies détectées : {anomalies_detectees}")

    if anomalies_detectees > 3:
        print("Alerte : Trop d'anomalies détectées dans le réseau !")
    else:
        print("Le trafic est dans les limites acceptables.")

paquets_reseau = [120, 300, 450, 600, 50, 700, 900, 80, 250]
seuil_critique = 500
analyse_trafic_reseau(paquets_reseau, seuil_critique)
```