

# SNT - Exercices Python progressifs

De la première instruction aux fonctions

Nom : \_\_\_\_\_

Prénom : \_\_\_\_\_

Classe : \_\_\_\_\_

Date : \_\_\_\_\_

**Consigne générale.** Tous les exercices sont à programmer et à tester dans l'éditeur Python en ligne. Utilise la fiche débranchée comme aide pour retrouver la syntaxe. Avant de lancer un programme, essaie de prévoir son résultat.

**Règle :** quand un programme fonctionne, teste-le avec plusieurs valeurs. Un programme qui fonctionne une fois n'est pas forcément correct.

## Exercice 1 - Premier programme

**Objectif :** afficher du texte avec `print()`

Écris un programme qui affiche sur trois lignes : ton prénom, ta classe, puis la phrase *Je découvre Python*. Modifie ensuite ton programme pour afficher une quatrième ligne contenant le résultat de **12 + 7**.

```
print("Bonjour !")
```

**Aide :** Le texte placé entre guillemets est affiché tel quel. Un calcul peut être placé directement dans `print(...)`.

## Exercice 2 - Variables et calculs

**Objectif :** stocker et modifier des valeurs

Crée deux variables **largeur** et **longueur** correspondant aux dimensions d'un rectangle. Calcule puis affiche son périmètre et son aire. Change ensuite les valeurs pour vérifier ton programme.

```
largeur = 4
longueur = 7

# à compléter
```

**Aide :** Périmètre =  $2 \times (\text{longueur} + \text{largeur})$ . Aire = longueur  $\times$  largeur.

**Pour aller plus loin :** Ajoute une variable **unite** contenant « cm » et améliore l'affichage.

## Exercice 3 - Demander une information

**Objectif :** utiliser `input()` et convertir une saisie

Demande à l'utilisateur son prénom puis son âge. Affiche ensuite un message donnant son âge l'année prochaine.

```
prenom = input("Ton prénom : ")
age = int(input("Ton âge : "))

# à compléter
```

**Aide :** `input()` renvoie du texte. `int(...)` permet de transformer une saisie numérique en entier.

## Exercice 4 - Pair ou impair ?

**Objectif :** écrire une condition `if / else`

Demande un entier à l'utilisateur. Le programme doit afficher *pair* si le nombre est pair, sinon *impair*.

```
nombre = int(input("Un entier : "))

if _____:
    print("pair")
else:
    print("impair")
```

**Aide :** Le reste d'une division par 2 se calcule avec `%`. Exemple :  $7 \% 2$  vaut 1.

## Exercice 5 - Conseil météo

**Objectif :** enchaîner plusieurs conditions

Demande une température. Affiche : *Manteau* si elle est inférieure à 10 °C ; *Pull* si elle est comprise entre 10 et 19 °C ; *T-shirt* à partir de 20 °C.

```
temperature = int(input("Température : "))

if temperature < 10:
    print("Manteau")
elif _____:
    _____
else:
    _____
```

**Aide :** Avec if / elif / else, les conditions sont testées dans l'ordre.

## Exercice 6 - Compter avec for

**Objectif :** répéter un nombre connu de fois

Écris un programme qui affiche les nombres de 1 à 10, un par ligne. Modifie-le ensuite pour afficher uniquement les nombres pairs de 2 à 20.

```
for i in range(1, 11):
    print(i)
```

**Aide :** range(début, fin) s'arrête juste avant la valeur **fin**. Il peut aussi recevoir un pas : range(début, fin, pas).

## Exercice 7 - Table de multiplication

**Objectif :** utiliser une boucle avec une variable

Demande un nombre à l'utilisateur, puis affiche sa table de multiplication de 1 à 10 sous une forme lisible, par exemple :  $7 \times 3 = 21$ .

```
nombre = int(input("Nombre : "))

for i in range(1, 11):
    # à compléter
```

**Aide :** À chaque tour de boucle, i prend une nouvelle valeur.

## Exercice 8 - Compte à rebours

**Objectif :** utiliser une boucle while

Écris un compte à rebours de 10 à 1 puis affiche *Décollage !*. Cette fois, utilise obligatoirement **while**.

```
n = 10

while _____:
    print(n)
    _____

print("Décollage !")
```

**Aide :** La variable qui contrôle une boucle while doit généralement être modifiée dans la boucle.

## Exercice 9 - Saisie contrôlée

**Objectif** : répéter jusqu'à obtenir une valeur valide

Demande à l'utilisateur une note comprise entre 0 et 20. Tant que la valeur n'est pas dans cet intervalle, redemande une note. Lorsque la note est valide, affiche *Note enregistrée*.

```
note = int(input("Note sur 20 : "))

while _____:
    print("Valeur incorrecte")
    note = _____

print("Note enregistrée")
```

**Aide** : La valeur est incorrecte si elle est **inférieure à 0 OU supérieure à 20**.

## Exercice 10 - Premières fonctions

**Objectif** : définir, appeler et tester une fonction

Écris les trois fonctions suivantes puis teste chacune d'elles avec plusieurs valeurs :

1. **double(n)** renvoie le double de n ;
2. **carre(n)** renvoie le carré de n ;
3. **est\_majeur(age)** renvoie True si age est supérieur ou égal à 18, sinon False.

```
def double(n):
    return _____

# écrire ensuite carre(n) et est_majeur(age)

print(double(6))
```

**Aide** : Une fonction est définie avec **def**. **return** renvoie un résultat que l'on peut stocker ou afficher.

**Pour aller plus loin** : Crée une fonction **maximum(a, b)** qui renvoie le plus grand des deux nombres.

## Exercice 11 - Défi final : le nombre mystère

**Objectif :** combiner while, conditions et fonction

Programme un jeu du nombre mystère. Pour commencer, fixe toi-même **mystere = 37**. Le joueur propose des nombres jusqu'à trouver. Après chaque proposition, le programme doit indiquer *Trop petit !* ou *Trop grand !*.

**Étape 1.** Complète la fonction **comparer()**.

**Étape 2.** Complète la boucle principale.

**Étape 3.** Ajoute un compteur d'essais et affiche le nombre d'essais à la fin.

```
def comparer(proposition, mystere):
    if proposition < mystere:
        return "Trop petit !"
    elif _____:
        return "Trop grand !"
    else:
        return "Bravo !"

mystere = 37
proposition = 0
essais = 0

while _____:
    proposition = int(input("Ta proposition : "))
    _____
    print(comparer(proposition, mystere))

print("Nombre d'essais :", essais)
```

**Aide :** La boucle doit continuer **tant que la proposition est différente du mystère**.

**Pour aller plus loin :** Quand tout fonctionne, remplace 37 par un nombre aléatoire entre 1 et 100 avec **import random** puis **random.randint(1, 100)**.

## Bilan

À la fin de cette activité, tu dois être capable d'écrire et tester un programme utilisant : **print()**, des **variables**, **input()**, des calculs, **if / elif / else**, **for**, **while** et une **fonction** définie avec **def** et **return**.