

# Exercices Python

**!/ Tout les programmes doivent être résistant à des erreur de saisie**

## Exercice 1

Écrivez un programme qui calcule les 50 premiers termes de la table de 13  
Ex : 13, 26, 39, 52, 65, 78, 91, ...  
mais n'affiche que ceux qui sont des multiples de 7.  
Ex : 91, ...

## Exercice 2

Écrivez un programme qui demande une chaîne à l'utilisateur et qui détermine si cette chaîne est un [palindrome](#) ou pas.

radar => Oui	Noël a trop par rapport à Léon => Non
toto => Non	noel a trop par rap port a leon => Oui
toot => Oui	

## Exercice 3

Le jeu "deviner un nombre"  
Tirer au sort un nombre entre 1 et 100  
Le faire deviner à l'utilisateur en précisant "Plus haut" ou "Plus bas"  
Afficher le nombre de coups.

## Exercice 4 – Diviseurs

Écrivez un programme qui demande un nombre à l'utilisateur et qui affiche la liste de ses diviseurs.

Ex :  
Un nombre : 20  
Liste des diviseurs de 20 :  
2  
3  
4  
5  
10

Truc : utiliser la fonction range

## Exercice 5 – Saisie avancée

Reprendre l'exercice 4 mais boucler sur la saisie tant que la réponse n'est pas cohérente (nombre entier entre 1 et 1000)

## Exercice 6 – Truc

Écrivez un programme qui affiche les nombres divisible par 7 mais pas par 5 compris entre 2000 et 4000.

```
2000 => Non (pas divisible par 7)
2001 => Non (pas divisible par 7)
2002 => Oui (divisible par 7 et pas par 5)
2003 => Non (pas divisible par 7)
...
```

## Exercice 7 – Reverso

Écrivez un programme demande une phrase à l'utilisateur et qui inverse l'ordre des mots :

Ex :

```
Une phrase : Mon nom est Hugues
Hugues est non Mon
```

Truc : Découpez la chaîne sur les espaces

## Exercice 8 – Bidule

Écrivez un programme qui affiche les nombres compris entre 2000 et 4000 dont chaque chiffre est pair.

```
2000 => Oui
2001 => Non (1 est impair)
2002 => Oui
2003 => Non (3 est impair)
...
```

## Exercice 9 – Compteur

Écrivez un programme qui demande une phrase et qui affiche le nombre de lettre, de chiffres et de signes compris dedans

Lettres : a ~ z plus A ~ Z

Chiffres : 0 ~ 9

Signes : Tout le reste SAUF les espaces

Ex :

```
Hello world! 123
Lettres : 10
Chiffres : 3
Signes : 1
```

Truc : Regardez du côté des méthodes des strings (isalpha, isdigit, ...)

## Exercice 10 – Mot de passe

Écrivez une fonction qui vérifie la solidité d'un mot de passe reçu en paramètre.

La fonction répond vrai si le mot de passe est conforme, faux sinon.

Critères retenus :

- Au moins une lettre minuscule (a~z)
- Au moins une lettre majuscule (A~Z)
- Au moins un chiffre (0~9)
- Au moins un signe dans la série (\$£#-\_@)
- Au moins 10 caractères
- Au plus 36 caractères

## Exercice 11 – Multiples de 3 et 5

Si on liste les nombres inférieurs à 10 qui sont, soit multiple de 3 soit multiple de 5 on a :

3, 5, 6, 9

... leur somme est 23

Écrivez un programme Python qui calcule la somme des multiples de 3 ou 5 inférieurs à 1.000

## Exercice 12 - Fibonacci

La suite de Fibonacci c'est « Chaque rang est la somme des deux précédents ... et les 2 premiers valent 1 »

Rang	Fibonacci
1	1
2	1
3	2
4	3
5	5
6	8
7	13
...	...

Écrivez un programme Python qui calcule le rang N°.666 de la suite.

## Exercice 13 – Différence somme / carré

La somme des carrés des 10 premiers nombres c'est :

$$1^2 + 2^2 + 3^2 + \dots + 10^2 = 385$$

Le carré de la somme des 10 premiers nombres c'est :

$$(1 + 2 + 3 + \dots + 10)^2 = 55^2 = 3025$$

Et la différence des deux c'est :

$$3025 - 385 = 2640$$

Écrivez un programme Python qui calcule la même chose pour les 1.000 premiers nombres.

## Exercice 14 – Nombre premier

Un nombre premier c'est un nombre qui n'est divisible que par 1 et lui même.

Il n'y a pas moyen de calculer les nombres premiers. On ne peut que les tester un par un pour savoir si ils le sont ou pas ...

Les premiers que l'on rencontre sont

Rang	Nombre premier
1	2
2	3
3	5
4	7
5	11
6	13
7	17
...	...

Écrivez un programme Python qui calcule le 1.512<sup>eme</sup> nombre premier.

## Exercice 15 – Nombre premier, encore

Les nombres premiers inférieure à 10 sont :

2, 3, 5, 7

Leur somme est :

$2 + 3 + 5 + 7 = 17$ .

Écrivez un programme Python qui calcule la somme des nombres premier inférieure à 6.789.

## Exercice 16 – Puissance de deux

$2^{15} = 32768$

Et la somme de ses nombres c'est

$3 + 2 + 7 + 6 + 8 = 26$

Écrivez un programme Python qui calcule la somme des nombres de  $2^{666}$