

# Introduction à la Programmation en Python

## 1 Analyse et Simulation de programmes

### Exercice 1. Conversion en Secondes

- Simulez pas à pas la fonction `conversion_secondes` avec comme entrées : 12 jours, 12 heures, 12 minutes et 12 secondes.
- Annotez chaque ligne avec une description de son effet.
- Modifiez la fonction pour ajouter une entrée donnant un nombre d'années à convertir.

### Exercice 2. Branchement conditionnel

Simulez pas à pas la fonction `calculatrice` sur plusieurs entrées différentes, puis annotez chaque ligne avec une description de son effet.

Que se passe-t-il sur l'entrée  $a = 12$ ,  $b = 0$  et `operation = "%"` ?

### Exercice 3. Répétition

Simulez pas à pas la fonction `somme` avec comme entrée 6, puis annotez chaque ligne avec une description de son effet.

Modifier la fonction pour qu'elle calcule la factorielle de l'entrée.

### Exercice 4. Puissance

Testez la fonction `exp` avec pour entrée :

- $a = 12$ ,  $b = 3$
- $a = 12$ ,  $b = -3$

La fonction n'est pas correcte, elle n'est pas cohérente avec sa spécification. En simulant pas à pas, identifiez le problème, puis corrigez le.

### Exercice 5. Arithmétique

COURS

Voici deux algorithmes  $\mathcal{E}$  et  $\mathcal{P}$ .

---

#### Algorithme $\mathcal{E}$

---

**ENTRÉES:**  $a$  et  $b$  deux entiers

```

1. tant que  $a \neq b$  faire
2.   si  $a \geq b$  alors
3.      $a \leftarrow a - b$ 
4.   sinon
5.      $b \leftarrow b - a$ 
6.   fin si
7. fin tant que
8. renvoyer  $a$ 
```

---



---

#### Algorithme $\mathcal{P}$

---

**ENTRÉES:**  $n$  un entier

```

1.  $p \leftarrow$  VRAI
2.  $d \leftarrow 2$ 
3. tant que  $d^2 \leq n$  faire
4.   si  $n \bmod d = 0$  alors
5.      $p \leftarrow$  FAUX
6.   fin si
7.    $d \leftarrow d + 1$ 
8. fin pour
9. renvoyer  $p$ 
```

---

- Essayer de deviner leurs spécifications de sortie.
- Implémentez les algorithmes en Python.
- Simulez les algorithmes pas à pas.
- Vérifiez et éventuellement corrigez la spécification de sortie.

## 2 Premiers programmes

### Exercice 6. Logarithme entier

On rappelle que pour tout  $n \geq 1$  et  $b \geq 2$ , le logarithme entier en base  $b$  de  $n$  est l'unique entier  $l$  tel que

$$b^l \leq n < b^{l+1}$$

Implémentez en Python la fonction ayant les spécifications suivantes :

---

#### Algorithme `logarithme_entier`

---

**ENTRÉES:**  $n \geq 1$  et  $b \geq 2$  deux entiers

**SORTIES:** un entier. Le logarithme entier en base  $b$  de  $n$

---

### Exercice 7. Méthode de Halley pour le calcul de $\sqrt{2}$

Implémentez en Python la fonction `halley` qui calcule le  $n^{\text{ème}}$  terme de la suite récurrente suivante :

$$H_0 = 1, H_{n+1} = H_n \cdot \frac{(H_n^2 + 6)}{(3H_n^2 + 2)}$$

### Exercice 8. Suite de Fibonacci

COURS

Implémentez en Python la fonction `fibonacci` qui calcule le  $n^{\text{ème}}$  terme de la suite récurrente suivante :

$$F_0 = 0, F_1 = 1, F_{n+2} = F_{n+1} + F_n$$

### Exercice 9. Nombre parfait

Un nombre parfait est un nombre entier positif égal à la somme de ses diviseurs stricts (lui-même exclu).

Implémentez une fonction `parfait` qui prend en entrée un entier et renvoie vrai s'il est parfait.

### Exercice 10. Syracuse

COURS

*Les mathématiques ne sont pas encore prêtes pour de tels problèmes – Paul Erdős*

Une suite de Syracuse est définie par la relation de récurrence suivante :

$$S_{n+1} = \begin{cases} \frac{S_n}{2} & \text{si } S_n \text{ est pair,} \\ 3S_n + 1 & \text{si } S_n \text{ est impair.} \end{cases}$$

La conjecture de Syracuse stipule que, quelque soit  $S_0 \in \mathbb{N}^*$ , la suite de Syracuse débutant par  $S_0$  atteint 1. On notera qu'une fois la valeur 1 atteinte, la suite boucle sur les valeurs 1, 4, 2, 1, ...

Implémentez une fonction `syracuse` qui prend en entrée un entier strictement positif  $s_0$  et qui renvoie le premier rang auquel la suite atteint la valeur 1.

Est-ce que la fonction `syracuse` termine sur toutes les entrées ?

### 3 Génération de chaînes de caractère

*c.f. le mémo sur la manipulation de chaînes de caractères*

#### Exercice 11. Pyramide

Implémentez une fonction `pyramide` qui prend en entrée un entier  $n \geq 0$  et qui renvoie une pyramide d'allumettes à  $n$  étages en chaîne de caractères. *e.g.* pour  $n = 4$  :

```
|
||| | | | |
|||||
|||||||
```

#### Exercice 12. Table ASCII

Implémentez une fonction `table_ascii` qui ne prend aucune entrée et renvoie la table ascii dans une chaîne de caractère sous forme de tableau à double entrée :

	0	1	2	3	4	5	6	7	8	9
0										
10										
20										
30				!	"	#	\$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	'	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~			

#### Exercice 13. Table de multiplication

Implémentez une fonction `table_multiplication` qui prend un entier en entrée et renvoie la table de multiplication de 0 à  $n$  dans une chaîne de caractère sous forme de tableau à double entrée. *e.g.* pour  $n = 4$  :

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	6	8
3	0	3	6	9	12
4	0	4	8	12	16

### 4 POUR S'ENTRAÎNER

#### Exercice 14. Suites adjacentes

Implémentez en Python la fonction `adjacentes` qui prend en argument un flottant  $\epsilon$  et qui calcule le premier rang  $n$  termes des suites adjacentes suivantes, à partir duquel  $|a_n - b_n| < \epsilon$  :

$$b_{n+1} = \frac{a_n + b_n}{2}, \quad a_{n+1} = \frac{2}{b_{n+1}}, \quad a_0 = 1, b_0 = 2$$

#### Exercice 15. Méthode de Héron

La suite  $x^a$  définie si dessous converge vers  $\sqrt{a}$ .

$$\forall n \in \mathbb{N} \quad x_{n+1}^a = \frac{x_n^a + \frac{a}{x_n^a}}{2}, \quad x_0^a = a$$

Implémentez une fonction `heron` qui prend en argument deux entiers  $a$  et  $n$  positifs et renvoie  $x_n^a$ . Trouvez le rang à partir duquel  $x_n^2$  approche  $\sqrt{2}$  à  $10^{-10}$  près ( $\approx 1.4142135623$ ).

#### Exercice 16. Nombres Chanceux d'Euler

Un nombre chanceux d'Euler est un nombre  $n > 1$  tel que pour tout  $0 \leq i \leq n-2$ ,  $i^2 + i + n$  est premier.

Implémentez une fonction `chanceux_euler` qui prend en entrée un entier  $n$  et renvoie vrai si et seulement si  $n$  est chanceux.

### 5 POUR ALLER PLUS LOIN

#### Exercice 17. Fonction 91 de McCarthy

La fonction 91 de McCarthy est définie pour  $n \in \mathbb{N}$  par la formule suivante :

$$f(n) = \begin{cases} n - 10 & \text{si } n > 100 \\ f(f(n + 11)) & \text{sinon.} \end{cases}$$

Implémentez une fonction `mccarthy` qui prend en entrée un nombre entier  $n$  positif et renvoie  $f(n)$ . Observez le retour de la fonction pour les valeurs plus petites ou égales à 101.