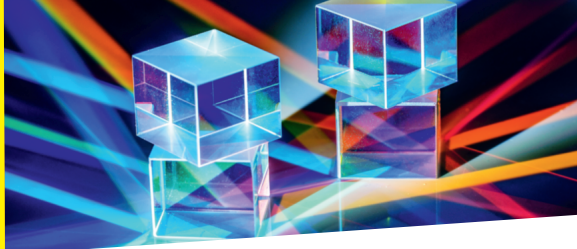


prépaBAC



2^{de}

Maths

**Cours &
entraînement**

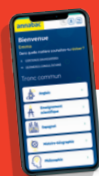
**NOUVEAU
BAC**

**Pour réussir tous
tes contrôles!**

- ✓ Un **cours** complet et visuel
- ✓ Les **méthodes** clés
- ✓ **300 exos & sujets** guidés
- ✓ Des **corrigés** expliqués

INCLUS

Des vidéos de révision



OFFERT Un abonnement
au site de révision **annabac**



Maths

Cours &
entraînement

► **Jean-Dominique Picchiottino**
Professeur agrégé de mathématiques
Lycée Pasquet (Arles)

Mode d'emploi

Voici les ressources que vous trouverez dans ce Prépacbac :

Dans chaque chapitre

Un cours visuel

1. Des fiches synthétiques et illustrées
2. Des méthodes détaillées
3. Une carte mentale récapitulative

Un entraînement progressif

1. Des quiz pour se tester
2. Des exercices guidés
3. Des sujets de type bac

Des corrigés détaillés



Et aussi : Des vidéos de révision.
Retrouvez leur liste **p. 8**

Nous vous recommandons d'utiliser régulièrement l'ouvrage, en fonction de vos besoins. Ainsi vous pourrez aborder vos contrôles en toute sérénité et acquérir les compétences nécessaires pour la 1^{re}.

L'auteur



Jean-Dominique
Picchiottino

SOMMAIRE

Algorithmique et programmation

1 Programmer en Python

COURS & MÉTHODES

1	Algorithmes	10
2	Instructions et fonctions	12
3	Pratique de la programmation en Python	14

MÉMO VISUEL	16
-------------------	----

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	18
-----------	--	----

CORRIGÉS	24
----------------	----

Nombres et calculs

2 Nombres rationnels



COURS & MÉTHODES

4	Diviseurs et multiples	28
5	Nombres pairs, impairs, premiers	30
6	Nombres décimaux et rationnels	32

MÉMO VISUEL	34
-------------------	----

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	36
-----------	--	----

CORRIGÉS	43
----------------	----

3 Calcul littéral et équations

COURS & MÉTHODES

7	Calcul littéral : addition et soustraction	52
8	Calcul littéral : multiplication, division et puissances	54
9	Développement et factorisation	56
10	Équations du premier degré à une inconnue	58
11	Systèmes de deux équations à deux inconnues	60

MÉMO VISUEL	62
-------------------	----

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	64
-----------	--	----

CORRIGÉS	73
----------------	----

4 Nombres réels et inégalités

COURS & MÉTHODES

12	Intervalle de $\mathbb{R}$	86
13	Distance et valeur absolue	88
14	Comparer deux réels	90
15	Inéquations	92
16	Racine carrée	94

MÉMO VISUEL	96
-------------	----

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	98
-----------	--	----

CORRIGÉS	108
----------	-----



Géométrie

5 Vecteurs et repérage dans le plan



COURS & MÉTHODES

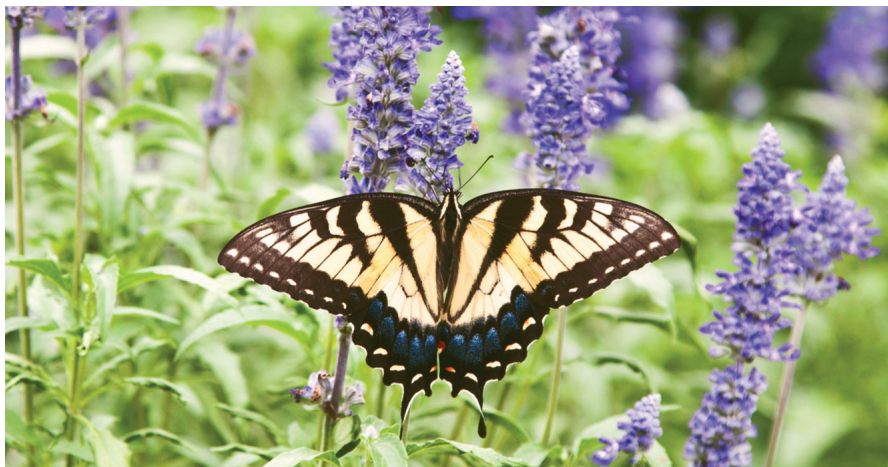
17	Translations et vecteurs	120
18	Addition de vecteurs	122
19	Multiplication d'un vecteur par un réel	124
20	Coordonnées d'un point, d'un vecteur	126
21	Vecteurs colinéaires	128
22	Coordonnées du milieu d'un segment, distance	130

MÉMO VISUEL	132
-------------	-----

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	134
-----------	--	-----

CORRIGÉS	142
----------	-----

6 Configurations du plan et trigonométrie



COURS & MÉTHODES

23	Quadrilatères particuliers	152
24	Triangles	154
25	Trigonométrie	156
26	Projection orthogonale	158

MÉMO VISUEL

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	162
-----------	--	-----

CORRIGÉS

.....	171
-------	-----

7 Droites dans le plan



COURS & MÉTHODES

27	Équations cartésiennes de droites	182
28	Équations réduites de droites	184
29	Coefficient directeur	186
30	Positions relatives de deux droites	188

MÉMO VISUEL

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	192
-----------	--	-----

CORRIGÉS

.....	201
-------	-----

Fonctions

8 Généralités sur les fonctions



COURS & MÉTHODES

31	Notions de fonction, d'image et d'antécédent	216
32	Représentation et lecture graphique	218
33	Variations d'une fonction	220
34	Fonctions paires et impaires	222

MÉMO VISUEL

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	226
-----------	--	-----

CORRIGÉS

234

9 Fonctions de référence

COURS & MÉTHODES

35	Fonctions affines	242
36	Fonction carré	244
37	Fonction cube	246
38	Fonction inverse	248
39	Fonction racine carrée	250

MÉMO VISUEL

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	254
-----------	--	-----

CORRIGÉS

264



Statistiques et probabilités

10 Informations chiffrées et statistiques

COURS & MÉTHODES

40	Informations chiffrées	276
41	Calculs statistiques	278

MÉMO VISUEL

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	282
-----------	--	-----

CORRIGÉS

.....	288
-------	-----

11 Probabilité et échantillonnage



COURS & MÉTHODES

42	Expériences aléatoires et probabilités	296
43	Calculs probabilistes	298
44	Échantillonnage	300

MÉMO VISUEL

EXERCICES	SE TESTER • S'ENTRAÎNER • OBJECTIF BAC	304
-----------	--	-----

CORRIGÉS

.....	311
-------	-----

INDEX	319
-------------	-----

Les vidéos de l'ouvrage



Retrouvez l'ensemble de ces vidéos
sur la chaîne Youtube **annabac**

► Maths 2^{de} – Les savoir-faire et démonstrations
hatier-clic.fr/mat_2



1 Programmer en Python



Alan Turing a imaginé une machine à calculer « universelle », capable de résoudre tout problème de calcul grâce à un algorithme. Durant la guerre, ses travaux ont permis aux Britanniques de déchiffrer les messages allemands codés avec la **machine Enigma**.

FICHES DE COURS

- | | | |
|---|--|----|
| 1 | Algorithmes | 10 |
| 2 | Instructions et fonctions | 12 |
| 3 | Pratique de la programmation en Python | 14 |

MÉMO VISUEL

16

EXERCICES

- | | | |
|--------------|---------------------------------------|----|
| SE TESTER | Exercices 1 à 3 | 18 |
| S'ENTRAÎNER | Exercices 4 à 12 | 19 |
| OBJECTIF BAC | Exercices 13 et 14 • Problèmes guidés | 22 |

CORRIGÉS

- | | |
|------------------|----|
| Exercices 1 à 14 | 24 |
|------------------|----|

1

Algorithmes

En bref

Au collège, vous avez utilisé le logiciel Scratch pour exécuter des algorithmes. Qu'est-ce qu'un algorithme ? Quel rapport y a-t-il entre un algorithme et un programme ?

I Algorithmes

En simplifiant, on peut dire qu'un algorithme est une **suite finie de tâches** (ou instruction) à effectuer (on dit aussi à *exécuter*) dans un ordre déterminé.

Exemples :

Algorithme en langage naturel

1. Mettre une pièce de 1 € dans la fente du distributeur.
2. Appuyer sur le bouton « soda au citron ».
3. Récupérer la cannette.

Algorithme dans Scratch



Lorsqu'on lance l'algorithme dans Scratch en cliquant sur le drapeau, le lutin dit « Hello ! » pendant trois secondes, puis l'ordinateur joue le son « miaou ».



À NOTER

Les instructions du logiciel Scratch sont prédéfinies et ont la forme de pièces de puzzle.

II Programmes

1 Programme, script, code

- Un **programme**, que l'on dénomme aussi **script**, est la traduction d'un algorithme dans un langage compréhensible par un utilisateur (personne ou machine).
- Si on veut faire exécuter un algorithme par un ordinateur, on doit utiliser un langage adapté (par exemple, Python → [FICHE 2](#)), qu'on appelle aussi un **code**.

2 Variables informatiques

Lorsque l'ordinateur exécute un code, les **noms** des variables ne changent pas, mais leurs **contenus** peuvent changer à chaque exécution d'une instruction.

Exemple : Ci-contre, la variable x contient au départ la valeur 0 (c'est l'initialisation) puis x prend la valeur 3, puis y prend la valeur 3^2 ; puis x prend la valeur 6, puis y prend la valeur 6^2 et ainsi de suite jusqu'à ce que x prenne la valeur 30 puis y la valeur 900.



Méthode

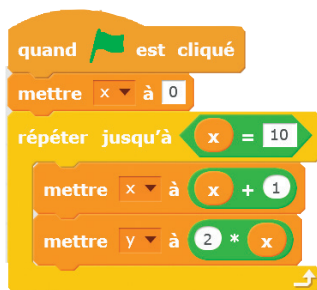
Simuler l'exécution d'un programme

On considère le programme Scratch ci-contre.
Quel est le contenu des variables x et y lorsque l'exécution du code s'achève ?



CONSEILS

Exécutez le code pas à pas et visualisez les différents contenus de x et de y en respectant la condition de la répétition.



SOLUTION

Lors du lancement du programme, la variable x contient le nombre 0 et la variable y est vide.

Le tableau suivant indique en bleu le contenu des variables x et y lors des trois premières exécutions des instructions.

Étiquette →	x	y	
État 1	0	vide	État initial
Exécution des instructions			Première exécution
État 2	1	2	
Exécution des instructions			Deuxième exécution
État 3	2	4	
Exécution des instructions			Troisième exécution
État 4	3	6	

Le code s'achève lorsque la variable x contient la valeur 10. La variable y contient alors la valeur 20, calculée à la dixième exécution.

En effet, l'algorithme ordonne d'exécuter les instructions tant que le contenu de la variable x est inférieur ou égal à 10. Avant l'éventuelle onzième exécution, l'algorithme vérifie si $x = 10$. C'est le cas, il s'arrête alors. Le contenu de la variable y est donc 20.

Étiquette →	x	y	
État 10	9	18	
Exécution des instructions			Dixième exécution
État 11	10	20	
Exécution des instructions			Onzième exécution
Arrêt du programme			Fin du script

2

Instructions et fonctions

En bref

En Seconde, vous allez étudier les concepts de base de la programmation, et notamment apprendre le langage Python. Ce langage permet de s'initier à la notion de fonction.

I Instructions

- **Affecter une valeur** v à une variable x consiste à placer dans la boîte étiquetée x la valeur v .

Exemple :

Algorithme	Code Python	Sens
$x \leftarrow 5$	<code>x=5</code>	Affecter la valeur 5 à la variable x .

- L'**instruction conditionnelle** `if ... then ... else` se traduit par « si ... alors ... sinon ».

Exemple :

Algorithme	Python	Sens
Si $x \leq 0$ alors $y \leftarrow 2x$ sinon $y \leftarrow 3x + 1$	<code>if x<=0: y=2*x else: y=3*x+1</code>	L'utilisateur ayant saisi une valeur de x : si le contenu de la variable x est négatif ou nul, affecter la valeur $2x$ à la variable y , sinon lui affecter la valeur $3x + 1$.

Attention, dans le langage Python, il faut respecter **scrupuleusement** la ponctuation, le passage à la ligne et **tous** les espaces indiqués.

MOT CLÉ

L'ajout d'espaces en début de ligne s'appelle une **indentation**.

II Boucles itératives

L'instruction `for ...` permet d'exécuter des blocs d'instructions se répétant un **nombre connu** de fois, par exemple 10 fois.

Exemple :

Algorithme	Python	Sens
$s \leftarrow 0$ Pour $k = 1$ à 10, $s \leftarrow s + 1/k$. Afficher la valeur de s	<code>s=0 for k in range(1,11): s=s+1/k print(s)</code>	Initialisation : création d'une variable s contenant 0. Calculer, puis afficher : $s = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{10}$

L'instruction `range(a,b)` ordonne à une variable (ici k) de prendre toutes les valeurs entières de **a inclus à b exclu**.

L'instruction `print(s)` ordonne d'afficher la valeur contenue dans la variable s .



À NOTER

L'instruction `while` permet d'exécuter une boucle avec condition ➔ **MÉTHODES**.

Méthodes

1 | Comprendre une boucle avec condition

a. Décrire la tâche du code suivant, écrit en Python :

```
s, k = 0, 0 #initialisation
while s < 10: #boucle avec condition
    k = k + 1
    s = s + 1/k #fin de la boucle
print(s, k) #sortie de la boucle : afficher s et k.
```



À NOTER

On peut insérer des commentaires dans le code à l'aide du symbole # (dièse).

b. Quelles sont les valeurs affichées en fin d'exécution ?



CONSEILS

- a. Il faut comprendre ce que doit exécuter le code tant qu'une certaine condition est réalisée.
- b. Écrivez le contenu de la variable s sous la forme d'une somme.

SOLUTION

a. Tant que la variable s est inférieure à 10, on augmente de 1 la valeur contenue dans la variable k , puis on ajoute l'inverse de cette valeur dans la variable s .

b. À la sortie de la boucle, la variable s contient la somme $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$, où n est le plus petit entier naturel k tel que $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k} > 10$.

Remarque : En exécutant le code, on trouve $s \approx 10,000043$ et $n = 12\,367$.

2 | Définir une fonction

a. Décrire le fonctionnement de la fonction g définie dans le code suivant.

```
def g(x): #définition de la fonction g
    if x > 0:
        return(x) #affichage de g(x)
    else:
        return(-x) #affichage de g(x)
```



À NOTER

On définit une fonction avec les termes *def* pour nommer la fonction et *return* pour faire afficher l'image d'un nombre par la fonction.

b. Quelles sont les valeurs affichées lorsqu'on saisit $g(10)$, $g(-20)$ et $g(0)$?



CONSEILS

- a. Traduisez en langage naturel les valeurs de $g(x)$ selon le signe de x .

SOLUTION

- a. Si x est strictement positif, alors $g(x) = x$ et si $x \leq 0$ alors $g(x) = -x$.
- b. On a bien sûr $g(10) = 10$, $g(-20) = 20$ et $g(0) = 0$.

En bref

Le langage Python est extrêmement puissant. Pour exploiter ses fonctionnalités, il est parfois nécessaire d'ajouter à sa version de base (on dit aussi sa distribution) des compléments appelés bibliothèques.

I Les types de variables

En Seconde, on étudie des variables informatiques de trois types.

- Le type **int** (qui est l'abrégié de *integer*, mot anglais signifiant « entier ») pour les nombres entiers.
- Le type **float** pour les nombres réels à virgule dite flottante.
- Le type **string** pour les chaînes de caractères.
- Pour connaître le type d'une variable *x* il suffit de taper `type(x)` à la console.

Exemple :

```
>>> x = 3.14
>>> type(x)
<class 'float'>
```

II L'instruction `input`

L'instruction `input` sert à dialoguer avec l'utilisateur. Elle définit une variable contenant ce que l'utilisateur a saisi.

Exemple :

Code Python	Affichage
<pre>nom = input('Quel est ton prénom ?') age = input('Quel âge as-tu ?') print('Bonjour', nom, 'tu as', age, 'ans.')</pre>	<pre>Quel est ton prénom ? Polycarpe Quel âge as-tu ? 16 Bonjour Polycarpe tu as 16 ans.</pre>

Remarques : • Par défaut, l'instruction `input` considère la donnée saisie par l'utilisateur comme une chaîne de caractères.

- Si on veut absolument avoir un nombre entier dans la variable *age*, on utilise l'instruction `int`, exemple : `int(input('Quel âge as-tu ?'))`. Si l'utilisateur saisit un nombre non entier ou une chaîne de caractère, un message d'erreur s'affiche à la fin de l'exécution. On a de même l'instruction `float(input())`.

III Les tracés

La bibliothèque *matplotlib* (*mathematic plot library*) contient des outils permettant de tracer des courbes en 2D.

Exemple : La commande `plt.plot(objet)` ordonne de tracer un objet géométrique, par exemple un point.



À NOTER

On utilise cette bibliothèque avec le nom court *plt* comme suit :
import matplotlib.pyplot as plt