

ALGORITHMIQUE EN MATHS/SCIENCES

- Recommandations, conseils et pistes de réflexion -

Introduction

Au collège (notamment au cycle 4) les élèves s'initient à la programmation, notamment lorsqu'ils traitent le programme de mathématiques⁽¹⁾ ou lors des Enseignements Pratiques Interdisciplinaires (sous forme de projet bien souvent). Ainsi, arrivés en Lycée Professionnel, ils ont déjà écrit, mis au point et exécuté quelques programmes simples.

L'objectif visé n'est pas la connaissance experte et exhaustive d'un langage ou d'un logiciel particulier mais plutôt de revisiter les notions de variables et de fonctions sous une forme différente. Afin que les élèves ne perdent pas les acquis du collège dans ce domaine (en particulier ceux envisageant une poursuite d'études en STS), ce document donne quelques exemples structurants visant à intégrer l'algorithmique et la programmation dans le programme de Maths/Sciences et plus généralement en Lycée Professionnel. Cette intégration contribue par ailleurs à « l'apport des notions et des outils permettant de maîtriser et de comprendre les gestes, des démarches et des processus inhérents au domaine professionnel. **Elles servent à préparer les futurs diplômés aux évolutions possibles ou probables de leurs métiers respectifs, et leur permettent une future reconversion professionnelle.** »⁽²⁾

Les outils proposés pourront notamment être mis en œuvre dans les situations favorables à l'utilisation des TIC présentées dans les deux **documents institutionnels de référence d'accompagnement produits par l'Inspection Générale**^{(3) (4)}:

- Situations favorables à l'utilisation des TIC pour l'apprentissage des concepts ou la résolution de problèmes,
- Intégrer l'algorithmique et la programmation dans les apprentissages en baccalauréat en baccalauréat professionnel.

I. Fondamentaux

Ici, nous définirons ce que sont algorithmique et programmation et ce que les élèves savent en quittant le collège.

Bon nombre de situations de la vie courante font appel à l'algorithmique: l'utilisation d'une application, la gestion d'un feu tricolore, d'un ascenseur, ou encore d'une barrière de péage qui s'ouvre lorsque la somme demandée est réglée. **Un algorithme** est la succession d'un certain nombre d'opérations dont l'objectif est de répondre à une problématique donnée. Généralement, il est constitué de plusieurs phases:

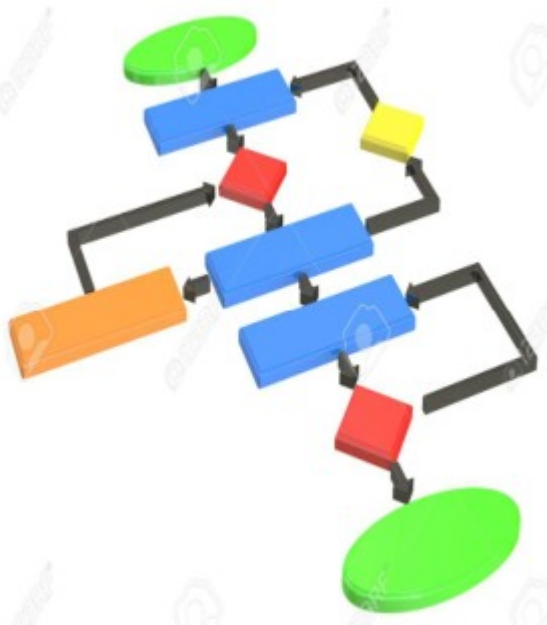
- la déclaration des variables qui seront utilisées
- l'initialisation de ces variables
- la phase de traitement
- la sortie de l'algorithme qui correspond souvent à l'affichage du/des résultats de la procédure

1. Méthodes pour présenter un algorithme

Le langage naturel : cette méthode consiste à utiliser des mots simples qui complètent la plupart du temps des verbes d'action dans un langage le plus clair possible.

Les logiciels : il existe de nombreuses applications permettant de rédiger un algorithme-LARP par exemple ⁽⁵⁾ - Certaines donnent même la possibilité de compiler un algorithme rédigé en amont.

Les algorigrammes : à l'instar des organigrammes utilisés par de nombreuses entreprises, il est possible de présenter un algorithme sous une forme plus graphique que les deux méthodes ci-dessus. La symbolisation employée doit répondre à la norme ISO 5807 et diffère selon l'objectif en regard de la procédure associée.



2. Distinction entre algorithme et programme

On peut considérer un algorithme comme l'illustration des étapes d'un raisonnement ou d'une démarche en vue de résoudre un problème. Une fois l'algorithme posé (au moyen d'un algorigramme ou du langage naturel), il pourra ensuite être codé dans le langage souhaité : on obtient alors un programme. Il existe de nombreux langage de programmation plus ou moins évolués (Python, Ti-basic, scratch, AlgoBox, C++, Lua,...).

Un algorithme doit être totalement indépendant du langage de programmation utilisé.

3. Quelques exemples d'algorithmes au collège

Loin d'être exhaustive, la liste d'exemples suivante indique des algorithmes abordés par les collégiens :

- Parcours d'un labyrinthe
- Programme de calcul
- Programmation d'un dessin à l'écran
- Parcours aléatoire
- Algorithme du prix à la caisse
- Algorithme de dépassement de budget
- Petit jeu de multiplication
- Tabulation d'une fonction
- ...

Nous reviendrons sur quelques-uns de ces exemples dans le paragraphe suivant.

II. Activités envisageables en Bac Pro

Outre le programme ordinaire, l'EGLS et les différents projets pluridisciplinaires peuvent être propices à la mise en œuvre de l'algorithmique et de la programmation. Nous allons ici classer les algorithmes et programmes correspondants en 3 catégories de difficultés (simple↗/moyenne↗↗/complexe↗↗↗) relatives aux élèves de Bac Pro.

1. L'algorithme de calcul d'un coût ↗

Contenus concernés: ALGÈBRE-ANALYSE/résolution d'un problème du 1^{er} degré

Situation : dans un magasin, chaque stylo coûte 0,60€ et chaque clef USB coûte 9,90€.

Objectif : calcul d'un coût

Ce que fait l'algorithme : l'algorithme calcule le prix à la caisse en fonction du nombre de stylos et de clefs achetés.

Exemple d'algorithme codé avec AlgoBox :

Code de l'algorithme

```

1  FONCTIONS_UTILISEES
2  VARIABLES
3    stylo EST_DU_TYPE NOMBRE
4    USB EST_DU_TYPE NOMBRE
5    prix EST_DU_TYPE NOMBRE
6  DEBUT_ALGORITHME
7    LIRE stylo
8    LIRE USB
9    prix PREND_LA_VALEUR stylo*0.60+USB*9.90
10   AFFICHER prix
11  FIN_ALGORITHME

```

Application :

Antoine veut acheter 10 clefs USB et 3 stylos.

Saisie : USB=10 et stylo=3

Affichage : « 100.8 ». Cela signifie qu'Antoine va devoir payer 100,80€ à la caisse.

2. L'algorithme de dépassement de budget ↗

Contenus concernés: ALGÈBRE-ANALYSE/résolution d'un problème du 1^{er} degré

Situation : dans un magasin, chaque stylo coûte 0,60€ et chaque clef USB coûte 9,90€.

Objectif : possibilité d'achat avec un budget donné

Ce que fait l'algorithme : Diane dispose d'un budget de 30€. L'algorithme informe Diane lorsqu'elle dépasse son budget.

Exemple d'algorithme codé avec AlgoBox :

Code de l'algorithme

```

1  FONCTIONS_UTILISEES
2  VARIABLES
3    USB EST_DU_TYPE NOMBRE
4    stylo EST_DU_TYPE NOMBRE
5    prix EST_DU_TYPE NOMBRE
6  DEBUT_ALGORITHME
7    LIRE USB
8    LIRE stylo
9    prix PREND_LA_VALEUR USB*9.90+stylo*0.60
10   SI (prix>30) ALORS
11     DEBUT_SI
12     AFFICHER "Diane, tu dépasses ton budget !"
13     FIN_SI
14   AFFICHER prix
15  FIN_ALGORITHME

```

Application :

Diane veut acheter 3 clefs et 1 stylo.

Saisie : USB=3 et stylo=1

Affichage : « Diane tu dépasses ton budget ! 30,30 ». Cela signifie que Diane a dépassé son budget de 30,30 € (car elle en a pour 30,3€).

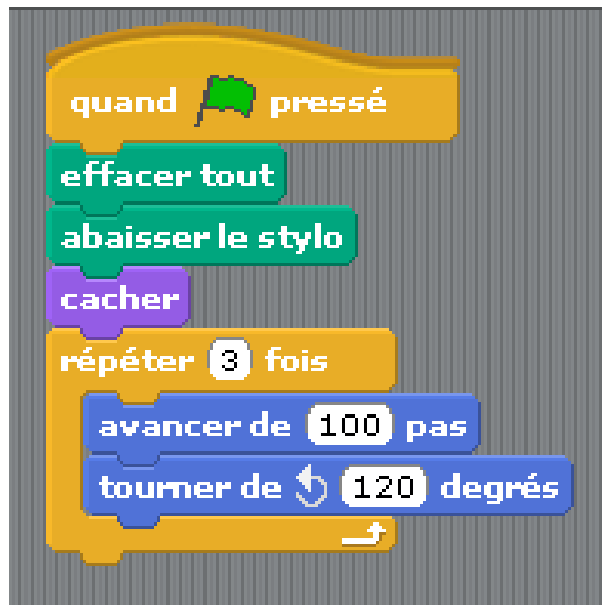
3. Algorithme qui trace un polygone régulier ↗

Contenus concernés : GEOMETRIE/De la géométrie dans l'espace à la géométrie plane

Objectif : tracé d'un triangle équilatéral

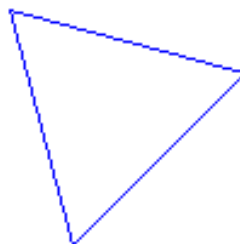
Ce que fait l'algorithme : l'algorithme trace un triangle équilatéral.

Exemple d'algorithme codé avec Scratch :



Application : on veut visualiser un triangle équilatéral à l'écran

Affichage :



Dans le même esprit, voyons maintenant un algorithme qui demande le nombre de côtés d'un polygone régulier, puis le trace à l'aide d'un robot mobile.

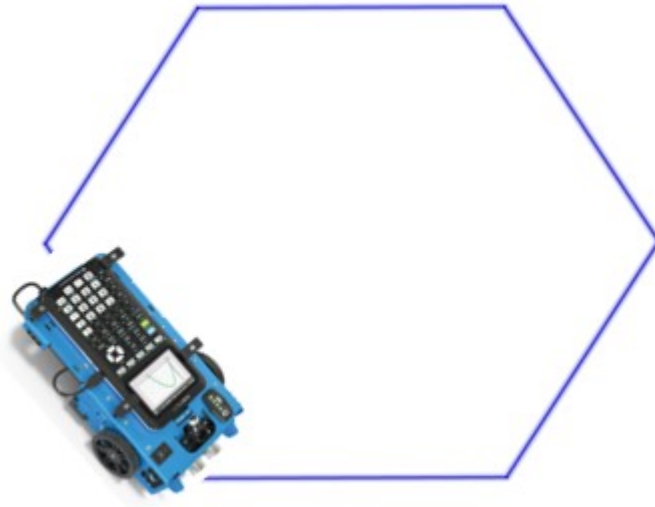
Algorithme en langage naturel:

- ♦ Se connecter au robot
- ♦ Entrer le nombre N de côtés
- ♦ $360/N \rightarrow A$
- ♦ *Pour* i allant de 1 à N
 - Avancer d'une unité
 - Tourner à gauche de A°
- Fin du Pour*

Algorithme (codé en Ti-basic) :

```
NORMAL FLOTT AUTO RÉEL RAD MP
EDIT MENU: [alpha] [f5]
PROGRAM: POLYREG
:Send("CONNECT RV")
:Disp "NBRE DE COTES"
:Input N
:360/N→A
:For(I,1,N)
:Send("RV FORWARD 1")
:Send("RV LEFT eval(A) DEG
REES")
:End
```

Application dans le cas où N=6 :



4. Affichage d'une couleur avec une Led RVB ↗

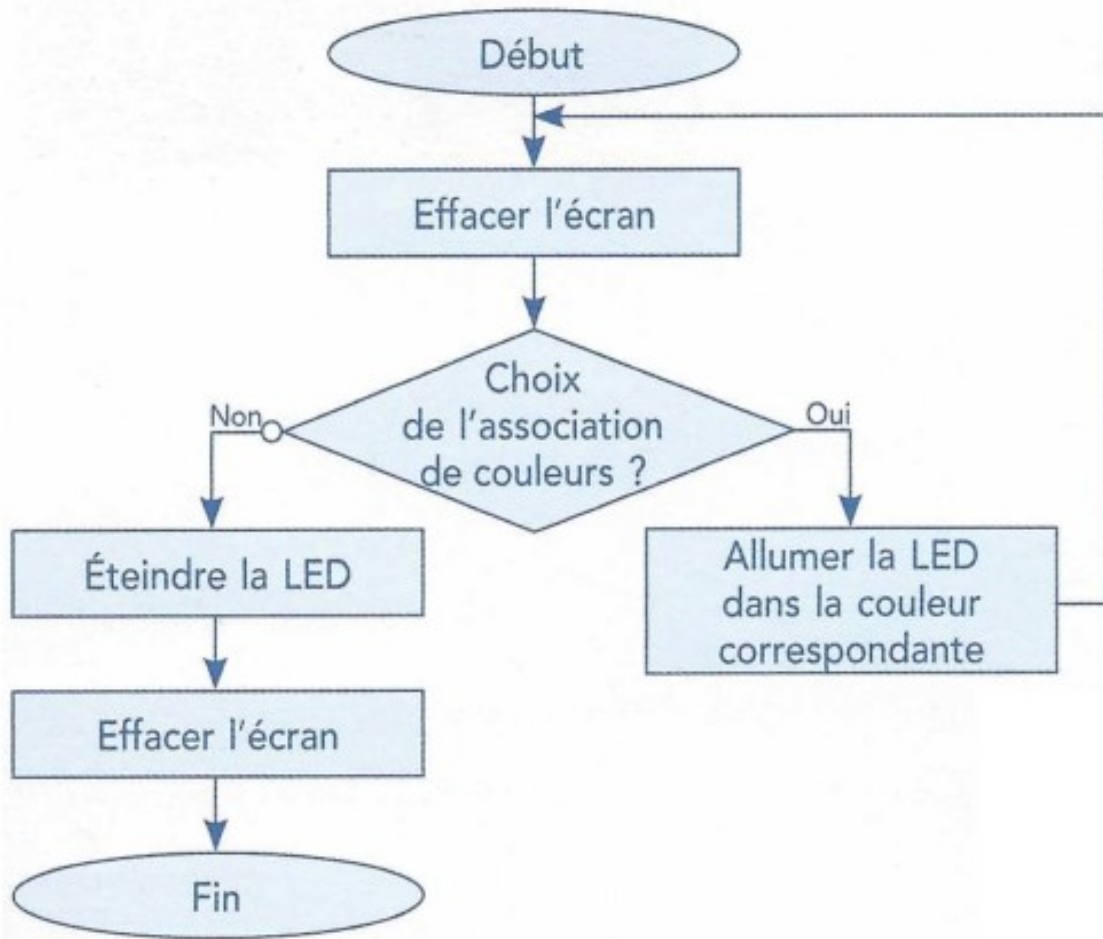
Chapitre concerné : Son et Lumière (SL5)

Ce que fait l'algorithme : l'algorithme demande trois valeurs des composantes (rouge, verte, bleue) et allume la Led avec la couleur correspondante sur un boîtier relié à la calculatrice.

Application :

Durant son concert, un chanteur a besoin d'être éclairé à l'aide d'une lumière jaune. Les deux éclairagistes vérifient le matériel dont la salle de concert est pourvue. Il s'avère que les projecteurs ne disposent que de filtres bleus, rouges et verts.

Les éclairagistes peuvent-ils répondre aux besoins du chanteur?

**L'algorithme :**

Exemple d'algorithme codé en Ti-basic :

```

NORMAL FLOTT AUTO RÉEL RAD MP
ÉDIT MENU: [α]Pha.] [f5]

PROGRAM: RVB
: Effécran
: Input "ROUGE:", R
: Input "VERT:", V
: Input "BLEU:", B
: Send("SET COLOR eval(R) e
val(V) eval(B)")

```

5. Résolution des équations du second degré ↗↗

Chapitre concerné : ALGEBRE-ANALYSE/du premier au second degré

Ce que fait l'algorithme : il effectue la résolution des équations $ax^2 + bx + c = 0$ en utilisant la méthode du discriminant.

Exemple d'algorithme codé avec Algobox :

Code de l'algorithme

```

1  FONCTIONS_UTILISEES
2  VARIABLES
3  a EST_DU_TYPE NOMBRE
4  b EST_DU_TYPE NOMBRE
5  c EST_DU_TYPE NOMBRE
6  delta EST_DU_TYPE NOMBRE
7  X_1 EST_DU_TYPE NOMBRE
8  X_2 EST_DU_TYPE NOMBRE
9  X_0 EST_DU_TYPE NOMBRE
10 DEBUT_ALGORITHME
11 LIRE a
12 LIRE b
13 LIRE c
14 delta PREND_LA_VALEUR pow(b,2)-4*a*c
15 SI (delta>0) ALORS
16   DEBUT_SI
17   X_1 PREND_LA_VALEUR (-b-sqrt(delta))/(2*a)
18   X_2 PREND_LA_VALEUR (-b+sqrt(delta))/(2*a)
19   AFFICHER X_1
20   AFFICHER X_2
21   FIN_SI
22   SINON
23     DEBUT_SINON
24     SI (delta==0) ALORS
25       DEBUT_SI
26       X_0 PREND_LA_VALEUR -b/(2*a)
27       AFFICHER X_0
28       FIN_SI
29     SINON
30       DEBUT_SINON
31       AFFICHER "il n'y a pas de solution réelle."
32       FIN_SINON
33     FIN_SINON
34 FIN_ALGORITHME

```


Application : On souhaite résoudre l'équation $x^2 - 5x + 6 = 0$

Saisie : a=1, b=-5, c=6

Affichage : « 2 et 3 ». Cela signifie que $S=\{2 ;3\}$.

Le codage de l'algorithme précédent, cette fois-ci avec le langage de programmation Python, est par exemple :

```
from math import *
a=float(input("Valeur de a :"))
b=float(input("Valeur de b :"))
c=float(input("Valeur de c :"))
Delta=b**2-4*a*c
if Delta>0:
    x1=(-b-sqrt(Delta**0.5))/(2*a)
    x2=(-b+sqrt(Delta**0.5))/(2*a)
    print(x1)
    print(x2)
elif Delta==0:
    x0=-b/(2*a)
    print(x0)
else:
    print("pas de solution")
```

Remarque : nous observons ici que Python est un langage de plus bas niveau qu'AlgoBox. Un langage est dit de haut niveau lorsqu'il est facilement accessible au novice. Ainsi Scratch et AlgoBox sont des langages de haut niveau car ils ont l'avantage d'être rédigés dans un français très explicite « tant que », « jusqu'à », « pour chaque », « afficher », « saisir » etc.

Python est quant à lui, un langage de plus bas niveau dont la syntaxe est moins intuitive que les deux cités précédemment mais est aussi un langage plus évolué dans le sens où il permet de traiter des classes d'objets beaucoup plus élaborés.

6. Intensité lumineuse et éclairage automatique ↗↗

Chapitre concerné : Activité en cycle terminal (module spécifique SL7)

Ce que fait l'algorithme : il demande à un capteur de lire la valeur de la luminosité ambiante, puis si cette valeur est au-dessous d'un seuil fixé, il allume une led.

Algorithme en langage naturel :

- ↳ Lire la valeur du capteur de luminosité.
- ↳ Stocker cette valeur dans la variable E
- ↳ Si $E \leq 3$ *
 - Alors allumer la Led
 - Sinon éteindre la Led
- FinSi

Algorithme (codé en Ti-basic) :

```

NORMAL FLOTT AUTO RÉEL RAD MP
ÉDIT MENU: [alpha][45]
PROGRAM: CAPLUM
:0→E
:11→S
:Send("CONNECT LED 1 TO OU
T 1")
:While S≠45
:Send("READ BRIGHTNESS")
:Get(E)
:Disp "E= ",E
:If E≤2
:Then
:Send("SET LED 1 TO ON")
:Else
:Send("SET LED 1 TO OFF")
:End
:getKey→S
:End

```

Montage :**7. Algorithme de résolution de l'équation $f(x) = c$ par dichotomie ↗↗↗**

Chapitre concerné : ALGÈBRE-ANALYSE/fonctions de la forme $f+g$ et kf

Ce que fait l'algorithme : il détermine une solution, si elle existe, de l'équation $f(x) = c$ sur l'intervalle $[a ; b]$ avec un encadrement à une précision donnée.

L'algorithme en langage naturel:

Entrer la valeur de a

Entrer la valeur de b

Entrer la valeur p de la précision voulue

Tant que $|b-a| > p$

$$\frac{(a+b)}{2} \rightarrow m$$

Si $f(a) \times f(m) \leq 0$

Alors $m \rightarrow b$

Sinon $m \rightarrow a$

Fin Si

Fin Tant que

Afficher m

L'algorithme (codé en Ti-basic)

```

NORMAL FLOTT AUTO RÉEL RAD MP
EDIT MENU: [a.] [pho.] [f5]
PROGRAM:DICHOTO
:Input A
:Input B
:Input P
:While abs(B-A)>P
:(A+B)/2→M
:If Y1(A)*Y1(M)≤0
:Then
:M→B
:Else
:M→A
:End
:End
:Disp M

```

Remarques :

- À travers l'étude de cet algorithme, on explicite aux élèves que résoudre l'équation $f(x) = c$ revient à résoudre l'équation $f(x) - c = 0$.
- On peut effectuer une différenciation pédagogique en demandant d'entrer l'expression de la fonction dans l'écriture du programme ou encore en rajoutant des instructions/messages afin de rendre l'exécution du programme plus conviviale comme c'est le cas dans la version ci-dessous codée en Python.

L'algorithme (codé en Python) relatif à la résolution de l'équation $x^3 + 2x^2 + 10x - 20 = 0$ sur l'intervalle [1 ; 2] :

```

a=float(input("valeur de a?"))
b=float(input("valeur de b?"))
precision=float(input("Donnez la précision:"))
while abs(b-a)>precision:
    m=(a+b)/2
    if (a**3+2*a**2+10*a-20)*(m**3+2*m**2+10*m-20)<0:
        b=m
    else: a=m
print("la valeur approchée est comprise entre ",a," et ",b)

```

Cette situation est proposée dans le document "Situations favorables à l'utilisation des TIC pour l'apprentissage des concepts ou la résolution de problèmes en Bac Pro"⁽⁵⁾ qui en propose quelques autres en regard du référentiel de Maths/Sciences.

III. Ressources

➔ Où trouver les logiciels de codage utilisés dans ce document : Scratch, Ti-basic, Algobox, et Python ?

- **Scratch** est disponible sous Linux, OsX et Windows. Le lien suivant pointe vers la version Windows : <https://scratch.fr.uptodown.com/windows/telecharger>
Les fichiers générés ont une extension en *.sb*
- **Ti-basic** : il est intégré à toutes les calculatrices Texas Instruments
Les fichiers générés ont une extension en *.8xp*
N.B : les autres calculatrices telles que Casio, HP, NumWorks possèdent également leur langage de programmation permettant d'implémenter des algorithmes. Le choix de tel ou tel autre langage est uniquement d'ordre pédagogique.
- **Algobox** peut être installé en fonction de l'environnement utilisé en suivant le lien suivant : <http://www.xm1math.net/algobox/download.html>
Les fichiers générés ont une extension en *.alg*
- **Python** peut être installé de diverses manières (il est même nativement intégré à certains systèmes d'exploitation). Le lien ci-après explique les différentes procédures d'installation suivant l'environnement utilisé :
<https://fr.wikihow.com/installer-Python>
Les fichiers générés ont une extension en *.py*

➔ Quelles ressources ?

- **Scratch**
Tutoriels :
<https://www.magicmakers.fr/scratch-2-scratch-online-tutoriels>

Ressources pédagogiques :
<https://mathsciences.ac-versailles.fr/spip.php?article1327>
- **Ti-basic**
Tutoriels :
<https://data.brains-master.com/pdf/122757-apprendre-le-ti-basic-pas-a-pas.pdf>
<http://sdz.tdct.org/sdz/le-ti-basic-3.html#LeTi-Basic>

Ressources pédagogiques :
<https://education.ti.com/fr/activities/france/search>
<https://www.lestutosmaths.fr/fr/seconde/introduction-algorithmique>

- **Algobox**

Tutoriels :

<http://www.xm1math.net/algobox/tutoalgobox/index.html>

Ressources pédagogiques :

http://eduscol.education.fr/bd/urtic/maths/index.php?rpt=algobox&id_etab=0&id_niveau=0&id_theme=0&id_sstheme=0&id_logiciel=0&id_b2i=0&id_aca=0&id=&commande=chercher&ok=Chercher

- **Python**

Tutoriels :

https://kooor.developpez.com/tutoriels/python/apprendre_python_video/?page=les-bases-de-la-syntaxe-python

Ressources pédagogiques :

<https://ent2d.ac-bordeaux.fr/disciplines/mathematiques/des-tutoriels-pour-debuter-avec-le-langage-python/>

À noter également que le site de l'**IREM de l'Académie de la Réunion** regorge de ressources très intéressantes permettant d'intégrer l'algorithmique dans l'enseignement des Maths-Sciences : <http://irem.univ-reunion.fr/spip.php?article929>

Enfin, la **Délégation Académique au Numérique Educatif (DANE)** propose aux enseignants des ressources et des formations permettant d'intégrer l'algorithmique et la programmation à nos enseignements⁽⁶⁾.

Rédacteurs : Lionel VARICHON, IEN Maths-Sciences
Alexandre TECHER
Nicolas GIGANT
Eric TANG-KWOR
James PAINIAYE

(1)http://cache.media.eduscol.education.fr/file/Algorithmique_et_programmation/67/9/RA16_C4_MATH_algorithmique_et_programmation_N.D_551679.pdf

(2)
http://cache.media.education.gouv.fr/file/Fevrier/19/0/Rapport_Villani_Torossian_21_mesures_pour_enseignement_des_mathe_matiques_896190.pdf

(3)
http://cache.media.eduscol.education.fr/file/Mathematiques/91/6/Ress_prog-TIC_bacpro_237916.pdf

(4)
http://cache.media.eduscol.education.fr/file/Programmes/45/3/Integrer_l_algorithmique_et_la_programmation_dans_les_appre_ntissages_en_baccalaureat_professionnel_843453.pdf

(5)
<https://www.tice-education.fr/index.php/tous-les-articles-et-ressources/articles-informatiques/582-larp-logiciel-dalgorithms-et-de-resolution-de-problemes>

(6)
<https://pedagogie.ac-reunion.fr/ecole-numerique/se-former/article-catalogue-fil/news/detail/News/24-coder-cest-fun-et-plus-facile-quon-ne-limagine.html>