académie Reims

MISE EN ŒUVRE DE L'ALGORITHMIQUE ET DE LA PROGRAMMATION AU LYCÉE À L'AIDE DE python™

Radouan RAOUI, IA-IPR de mathématiques

Karl SKORNIK, chargé de missions

Janvier-Février 2018

SOMMAIRE

ALGORITHMIQUE ET PROGRAMMATION : QUE DISENT LES PROGRAMMES DU CYCLE 4 ET DE SECONDE GÉNÉRALE ET TECHNOLOGIQUE ?

ALGORITHMIQUE ET PROGRAMMATION AU CYCLE 4

Référence : Bulletin officiel spécial n°11 du 26 novembre 2015 - Annexe 3
http://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=94717

Au cycle 4, **les élèves s'initient à la programmation**, en développant dans une démarche de projet quelques **programmes simples**, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier. En créant un programme, ils développent des **méthodes de programmation**, revisitent les **notions de variables et de fonctions** sous une forme différente, et s'entraînent au raisonnement.

Attendus de fin de cycle

Écrire, mettre au point et exécuter un programme simple.

ALGORITHMIQUE ET PROGRAMMATION AU CYCLE 4

Connaissances et compétences associées	Exemples de situations, d'activités et de ressources pour l'élève
Décomposer un problème en sous-problèmes afin de structurer un programme ; reconnaître des schémas. **Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné.** Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs. Programmer des scripts se déroulant en parallèle. • **Notions d'algorithme et de programme.** • **Notion de variable informatique.** • **Déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles.**	Jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe. Réalisation de figure à l'aide d'un logiciel de programmation pour consolider les notions de longueur et d'angle. Initiation au chiffrement (Morse, chiffre de César, code ASCII, etc.). Construction de tables de conjugaison, de pluriels, jeu du cadavre exquis, etc. Calculs simples de calendrier. Calculs de répertoire (recherche, recherche inversée, etc.). Calculs de fréquences d'apparition de chaque lettre dans un texte pour distinguer sa langue d'origine : français, anglais, italien, etc.

ALGORITHMIQUE ET PROGRAMMATION AU CYCLE 4

Repères de progressivité

En cinquième, les élèves s'initient à la **programmation événementielle**. Progressivement, ils développent de nouvelles compétences, en programmant des actions en parallèle, en utilisant la **notion de variable informatique**, en découvrant les **boucles** et les **instructions conditionnelles** qui complètent les structures de contrôle liées aux événements.

ALGORITHMIQUE ET PROGRAMMATION AU CYCLE 4

Au collège, le logiciel **Scratch** offre un environnement d'édition et d'exécution des programmes. Il s'agit d'un logiciel gratuit et disponible sur toutes les plates-formes usuelles. Il permet de travailler tous les concepts figurant au programme du cycle 4, en particulier la programmation événementielle et la gestion des scripts s'exécutant en parallèle.

Référence :
https://scratch.mit.edu/

Exemple : Sujet de l'épreuve de mathématiques du Diplôme Nationale du Brevet 2017, exercice 2.

ALGORITHMIQUE ET PROGRAMMATION EN SECONDE GT

Référence : circulaire n° 2017-082 du 2 mai 2017 (aménagement du programme de mathématiques de seconde générale et technologique à la rentrée 2017)
http://cache.media.education.gouv.fr/file/18/95/3/ensel512_maths_757953.pdf

La **démarche algorithmique** est, depuis les origines, une **composante essentielle de l'activité mathématique**.

ALGORITHMIQUE ET PROGRAMMATION EN SECONDE GT

Au cycle 4, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Ce qui est proposé dans ce programme est une **consolidation des acquis du cycle 4** autour de deux idées essentielles :

- la **notion de fonction** d'une part, et
- la **programmation comme production d'un texte dans un langage informatique** d'autre part.

ALGORITHMIQUE ET PROGRAMMATION EN SECONDE GT

Dans le cadre de cette activité, les élèves sont entraînés :

- à décrire des algorithmes en langage naturel ou dans un langage de programmation ;
- à **en réaliser quelques-uns à l'aide d'un programme simple écrit dans un langage de programmation textuel** ;
- à interpréter des algorithmes plus complexes.

Un langage de programmation simple d'usage est nécessaire pour l'écriture des programmes. Le choix du langage se fera parmi les langages interprétés, concis, largement répandus, et pouvant fonctionner dans une diversité d'environnements.

ALGORITHMIQUE ET PROGRAMMATION EN SECONDE GT

L'**algorithmique** a une place naturelle dans tous les champs des mathématiques et les problèmes ainsi traités doivent être en **relation** avec les autres parties du programme (**fonctions**, **géométrie**, **statistiques et probabilité**, **logique**) mais aussi avec les **autres disciplines** ou la **vie courante**.

À l'occasion de l'écriture d'algorithmes et de petits programmes, il convient de donner aux élèves de **bonnes habitudes de rigueur** et de les entraîner aux **pratiques systématiques** de **vérification** et de **contrôle**. En programmant, **les élèves revisitent les notions de variables et de fonctions sous une forme différente**. Il convient d'y être attentif.

ALGORITHMIQUE ET PROGRAMMATION EN SECONDE GT

CONTENUS	CAPACITÉS ATTENDUES	COMMENTAIRES
Variables et instructions élémentaires	• choisir ou déterminer le type d'une variable (entier, flottant ou chaîne de caractères) ; • concevoir et écrire des affectations à des variables ; • écrire une formule permettant un calcul combinant des variables.	On commence par consolider les notions de variables, de boucles et d'instructions conditionnelles introduites au cycle 4 en complétant la programmation par blocs par l'utilisation d'un langage de programmation textuel.

ALGORITHMIQUE ET PROGRAMMATION EN SECONDE GT

CONTENUS	CAPACITÉS ATTENDUES	COMMENTAIRES
Boucle et itérateur, instruction conditionnelle	• programmer une instruction conditionnelle ; • programmer une boucle bornée ; • programmer une boucle non bornée.	On formalise les notions de boucle bornée (for) et de boucle non bornée (while) et on introduit la notion nouvelle de fonction dans un langage de programmation.
Notion de fonction	• programmer des fonctions simples, ayant un petit nombre d'arguments.	**Il est intéressant de confronter les fonctions dans un langage de programmation avec les fonctions d'un tableur.**

LE DOCUMENT D'ACCOMPAGNEMENT « ALGORITHMIQUE ET PROGRAMMATION »

DOCUMENT D'ACCOMPAGNEMENT

Un document d'accompagnement (juin 2017) sur le thème « Algorithmique et programmation » en seconde générale et technologique est disponible : https://cache.media.eduscol.education.fr/file/Mathematiques/73/3/Algorithmique_et_programmation_787733.pdf

Ce document présente des activités permettant d'éclairer des résultats et des méthodes mathématiques au travers d'algorithmes simples. Il contient de nombreuses situations d'apprentissages utilisant la programmation en Python.

DOCUMENT D'ACCOMPAGNEMENT

- Statistiques descriptives
- Algorithme d'Euclide
- Changement de base de numération
- Logarithme entier en base 2
- Test de primalité
- Décomposition en produit de facteurs premiers
- Longueur d'un arc de courbe
- Résolution approchée d'une équation par dichotomie
- Stabilisation des fréquences
- Aiguille de Buffon
- Intervalle de fluctuation d'une fréquence au seuil de 95%
- Casser un bâton
- Urnes de Polyà
- Réfraction d'un rayon lumineux
- Recherche d'un minimum par une autre méthode
- Détermination des coordonnées des sommets d'un triangle dont on connaît les longueurs des trois côtés

DOCUMENT D'ACCOMPAGNEMENT

[…] À la différence du programme de mathématiques du cycle 4 du collège, il s'agit donc d'adosser explicitement les activités de la partie algorithmique et programmation aux mathématiques.

Cet enseignement a un double objectif : faire travailler des notions mathématiques du programme dans un contexte différent, et poursuivre chez les élèves le développement des compétences suivantes, déjà travaillées au cycle 4 :

- décomposer un problème ;
- reconnaître des schémas ;
- généraliser et abstraire ;
- concevoir des algorithmes et les traduire dans un langage de programmation.

DOCUMENT D'ACCOMPAGNEMENT

Les modalités de l'apprentissage correspondant peuvent être variées : **travail individuel ou en groupe, en salle informatique ou en salle banale, au tableau ou sur papier, sur tablette ou sur ordinateur**. […]

- Les notions mathématique et informatique de fonction relèvent du même concept universel. En informatique, une fonction prend un ou plusieurs arguments et renvoie une valeur issue d'un calcul.
- Le choix d'un langage textuel, comme **Python**, au lieu d'un langage par blocs, comme Scratch, permet aux élèves de se confronter à **la précision et la rigidité d'une syntaxe proche de celle des expressions mathématiques**, avec l'avantage de pouvoir bénéficier du contrôle apporté par l'analyseur syntaxique.

académie Reims

INTRODUCTION À PYTHON

académie Reims

python™

Cliquer sur le lien du site officiel

PREMIERS PAS AVEC PYTHON

Download Python 3.6.4

Cliquer sur le lien du téléchargement

L'ORIGINE DE PYTHON

- Le langage Python est né dans les années 1990, au CWI Amsterdam, développé par Guido Van Rossum. Il est nommé ainsi par référence aux « Monty Python », célèbre groupe d'humoristes britanniques, très appréciés par Van Rossum.

- Les deux versions récentes de Python sont : la version 2.7 et la version 3.6.

- À noter que les versions 3.x ne sont pas compatibles avec les versions 2.x.

PYTHON :
UN LANGAGE DE PROGRAMMATION

- Python est un langage de programmation professionnel, libre et gratuit.
- C'est un **langage interprété** ce qui signifie qu'il n'y a pas de phase de compilation qui traduit le programme en langage machine (comme c'est le cas pour les langages C ou C++ qui sont des langages compilés). Avec Python, les instructions sont traitées au fur et à mesure de leur lecture par l'interprète.
- Il est **portable** c'est-à-dire qu'il peut fonctionner sous différents systèmes d'exploitation comme Windows, Linux, Mac Os, etc. **/!\ uniquement sous Windows pour Edupython !**
- Quel que soit le système d'exploitation, on peut utiliser Python dans un terminal ou avec **IDLE** (environnement de développement fourni à l'installation).

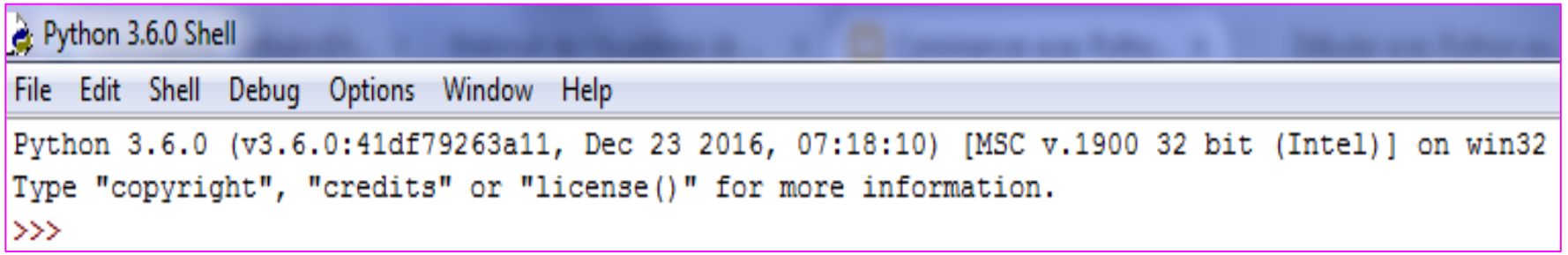
académie Reims

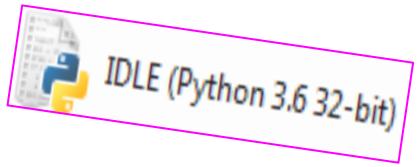
PYTHON :
UN LANGAGE DE PROGRAMMATION

```python
import math

def perimetre_cercle(un_rayon):
    """Calculer le périmètre d'un cercle à partir de son rayon.
        :param un_rayon: le rayon du cercle (positif)
        :return le périmètre d'un cercle de rayon un_rayon
    """
    diametre = 2 * un_rayon
    return math.pi * diametre
```

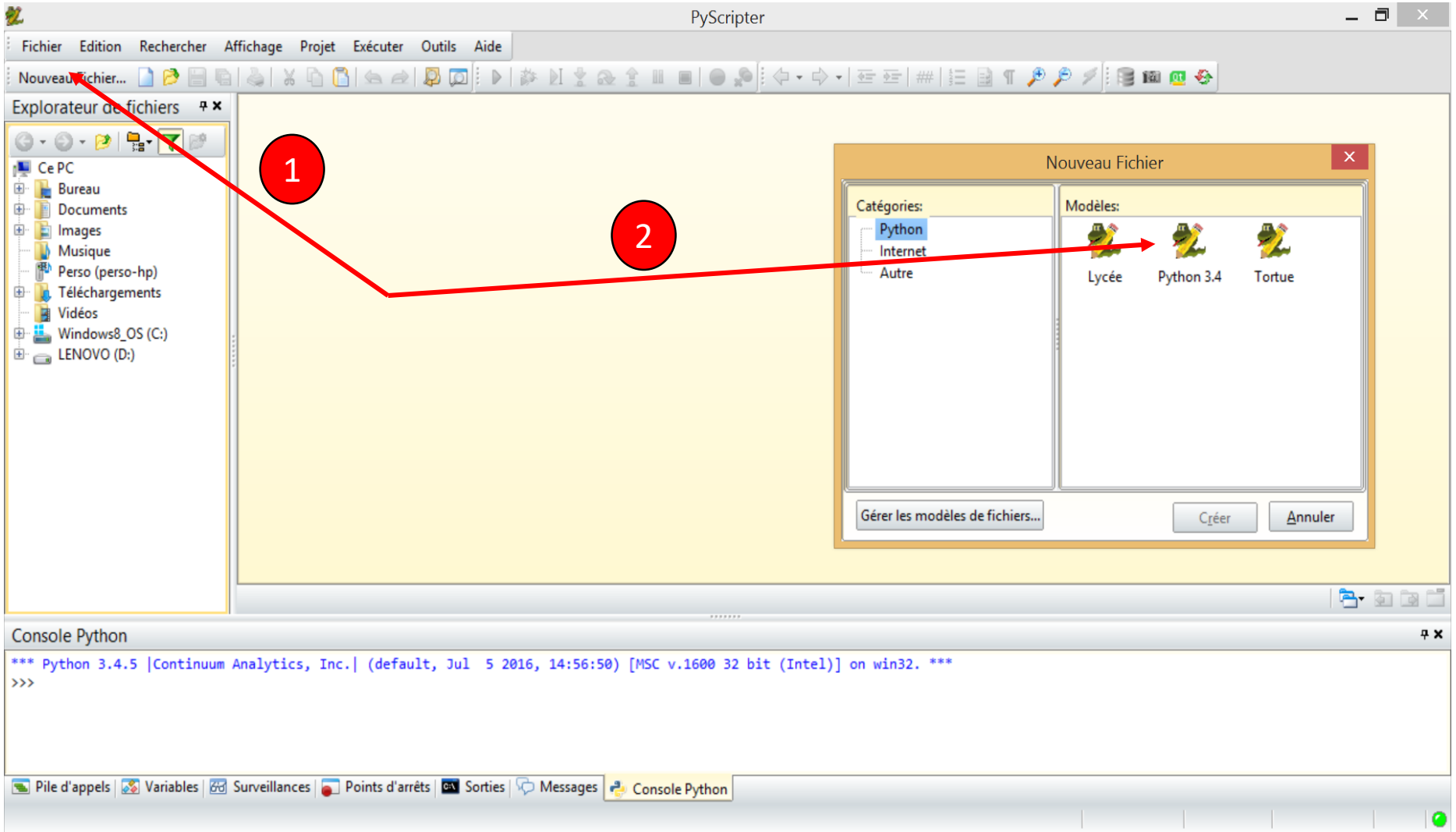
LANCER LA CONSOLE PYTHON, APRÈS INSTALLATION





- **Sous Windows** : passer par les menus Démarrer > Tous les programmes > Python 3.6 > **IDLE** (Python 3.6-32 bit).
- **Sous Linux** : lorsqu'il est installé sur le système, Python a créé un lien vers l'interpréteur sous la forme python3.X (le X étant le numéro de la version installée).
- **Sous Mac OS X** : chercher un dossier Python dans le dossier Applications, pour lancer Python, ouvrir l'application IDLE de ce dossier.

LANCER LA CONSOLE PYTHON, APRÈS INSTALLATION SOUS EDUPYTHON



EXEMPLE DE PROGRAMME ÉDITÉ AVEC IDLE SOUS WINDOWS

- Voici un exemple de programme :

PyScripter - module1*

fichage Projet Exécuter Outils Aide

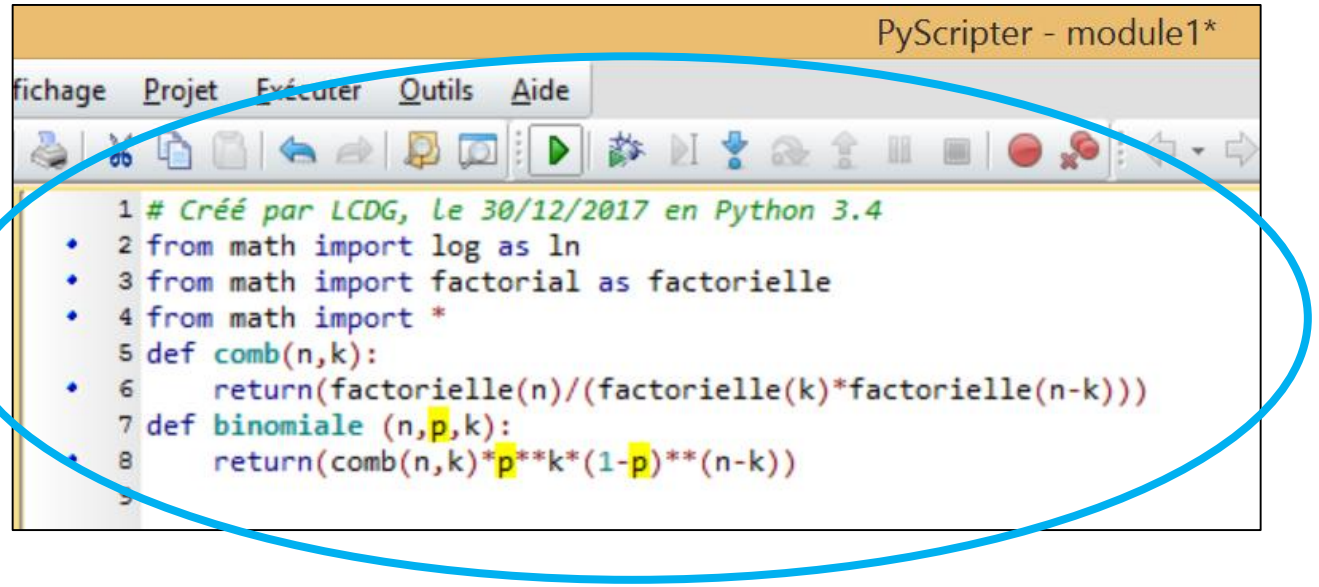
```
1  # Créé par LCDG, le 30/12/2017 en Python 3.4
2  from math import log as ln
3  from math import factorial as factorielle
4  from math import *
5  def comb(n,k):
6      return(factorielle(n)/(factorielle(k)*factorielle(n-k)))
7  def binomiale (n,p,k):
8      return(comb(n,k)*p**k*(1-p)**(n-k))
9
```

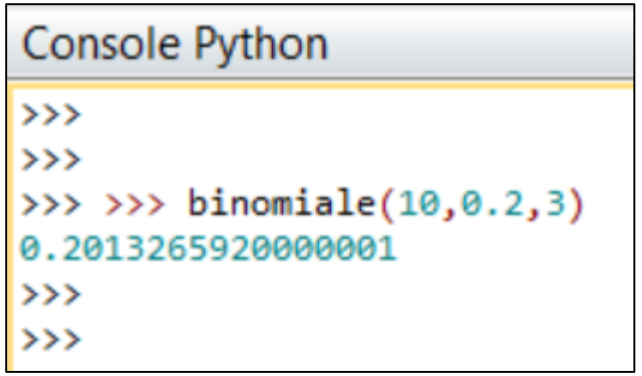
Console Python

```
>>>
>>>
>>> >>> binomiale(10,0.2,3)
0.2013265920000001
>>>
>>>
```

EXEMPLE DE PROGRAMME ÉDITÉ AVEC IDLE SOUS WINDOWS

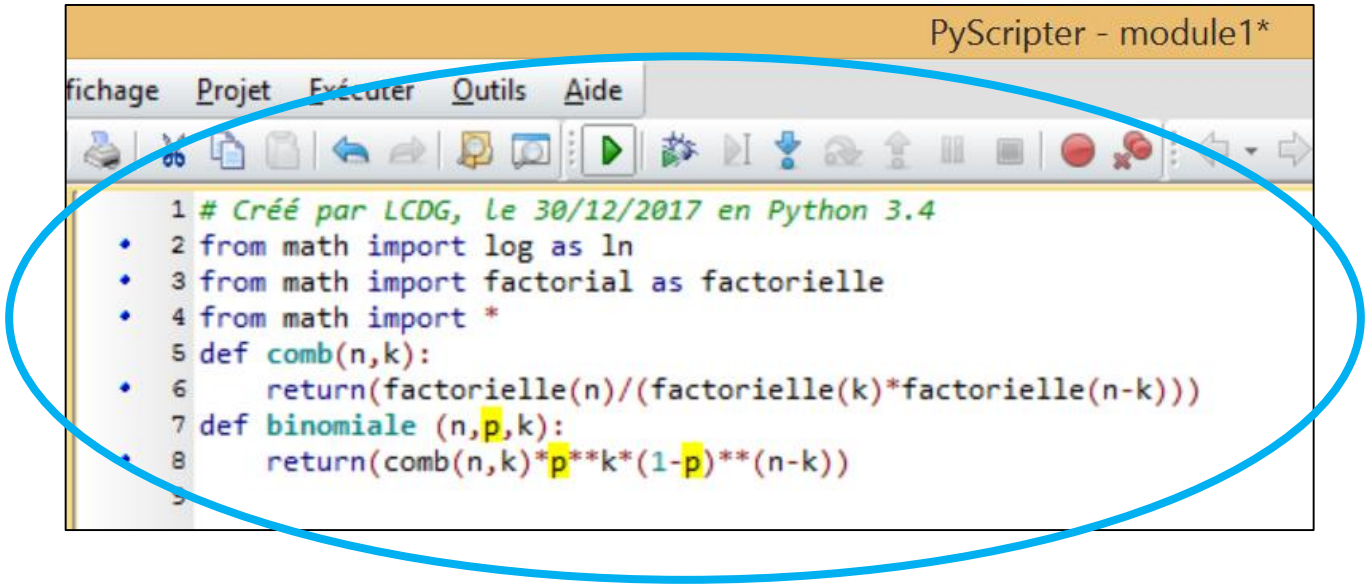
- Voici un exemple de programme : **le script se trouve dans l'éditeur (IDLE)**

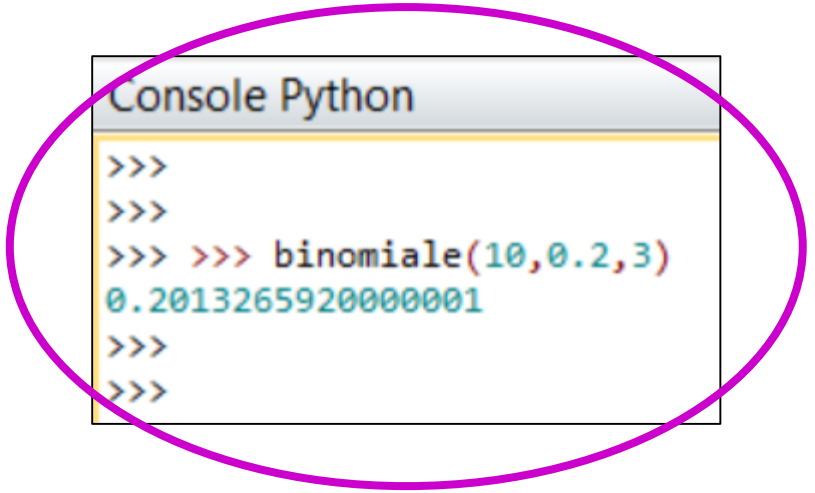




EXEMPLE DE PROGRAMME ÉDITÉ AVEC IDLE SOUS WINDOWS

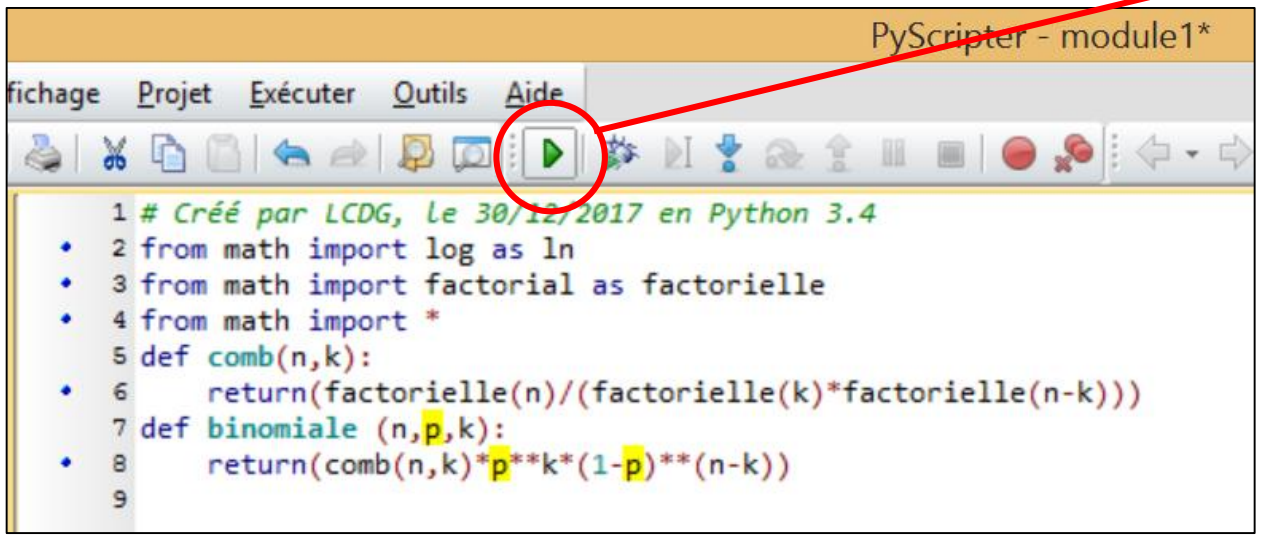
- Voici un exemple de programme : **le script se trouve dans l'éditeur (IDLE)** et **il s'exécute dans l'interpréteur (la console ou shell)**.

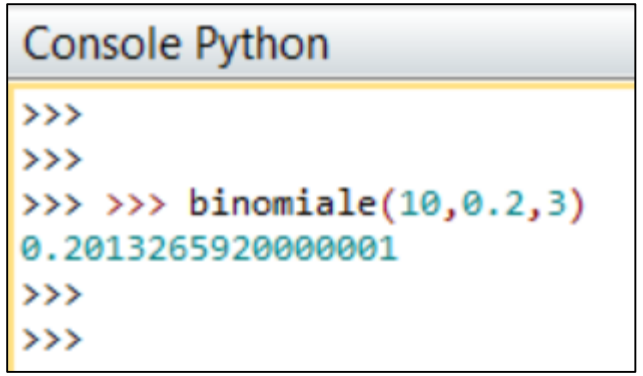




EXEMPLE DE PROGRAMME ÉDITÉ AVEC IDLE SOUS WINDOWS

- Voici un exemple de programme : le script se trouve dans l'éditeur (IDLE) et il s'exécute dans l'interpréteur (la console ou shell).
- Le plus souvent, le programme est saisi dans l'éditeur (IDLE), puis sauvegardé avec l'extension **.py** et lancé à l'aide de la touche 





académie Reims

QUELQUES EXEMPLES DE PROGRAMMATION EN PYTHON

EXEMPLE N°1
DE SCRATCH À PYTHON

2NDE

Référence : document d'accompagnement (page 6)

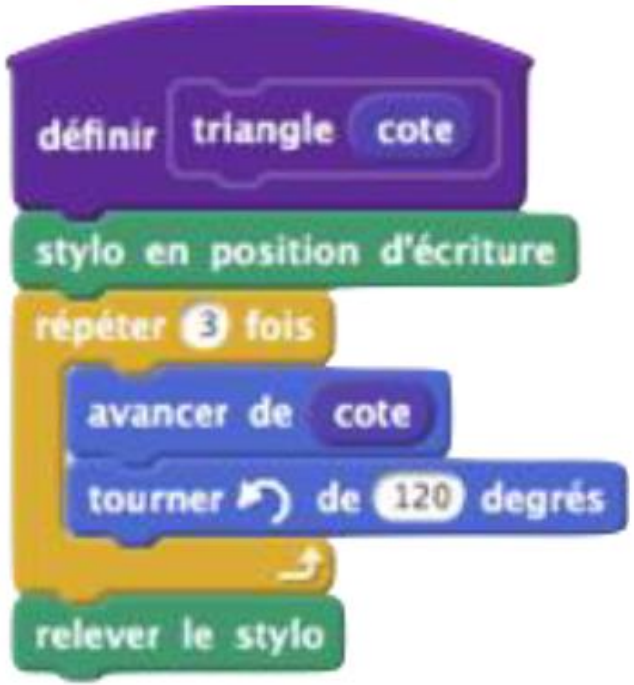
Les élèves de seconde ont suivi au collège un enseignement d'algorithmique et de programmation, dans le cadre des mathématiques et de la technologie. En mathématiques, à l'aide de Scratch, ils ont utilisé des boucles, des instructions conditionnelles.

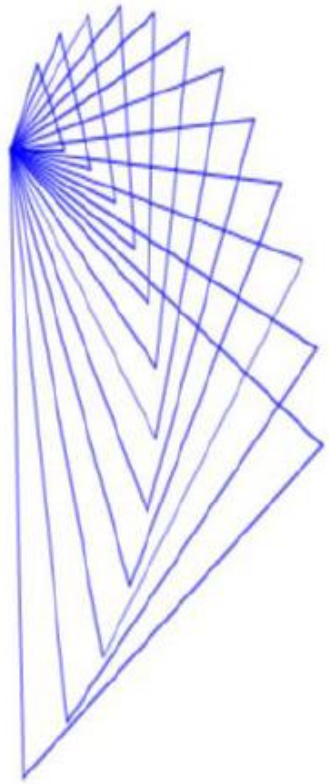
En classe de seconde, le passage de Scratch à Python peut être immédiat ou progressif, suivant les choix pédagogiques de l'enseignant. **Les deux langages comportent, au delà des différences évidentes de forme, des similitudes qui facilitent la transition.**

académie Reims

2NDE

EXEMPLE N°1
DE SCRATCH À PYTHON





EXEMPLE N°1
DE SCRATCH À PYTHON

```
import turtle

def figure():

    turtle.hideturtle()
    turtle.clear()
    turtle.up()
    turtle.setheading(0)
    turtle.goto(0,100)
    longueur = 40
    for i in range(12):
        triangle(longueur)
        turtle.right(8)
        longueur = longueur + 20

def triangle(cote):
    turtle.down()
    for i in range(3):
        turtle.forward(cote)
        turtle.left(120)
    turtle.up()
```

L'utilisation de la bibliothèque **Turtle** de Python permet d'écrire un programme très similaire.

Il suffit de taper **figure()** dans la console pour lancer l'exécution.

EXEMPLE N°2
LONGUEUR D'UN SEGMENT

2NDE

Écrire un programme qui affiche la longueur d'un segment [AB].

```
1 # Calcul de la longueur d'un segment [AB]
2 from math import *
3 XA=float(input("Donner l'abscisse de A : "))
4 YA=float(input("Donner l'ordonnée de A : "))
5 XB=float(input("Donner l'abscisse de B : "))
6 YB=float(input("Donner l'ordonnée de B : "))
7 L=sqrt((XB-XA)**2+(YB-YA)**2)
8 print ("La longueur L est égale à :", L)
```

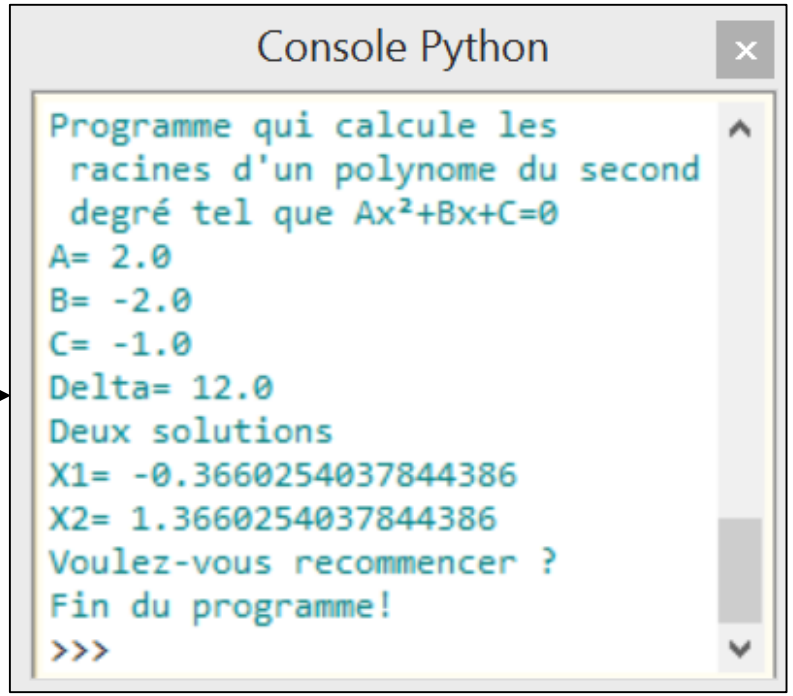
Console Python

```
*** Python 3.4.5 |Continuum
 Analytics, Inc.| (default, Jul  5
 2016, 14:56:50) [MSC v.1600 32 bit
 (Intel)] on win32. ***
>>>
La longueur L est égale à : 6
.514598989960933
>>>
```

EXEMPLE N°3
RACINE D'UN TRINOME

Écrire un programme qui calcule le discriminant Δ puis en conséquence trouve les solutions éventuelles de l'équation.

Exécution du programme dans la console



académie Reims

1ERES

EXEMPLE N°3
RACINE D'UN TRINOME

```
from math import *
# Permet d'importer toutes les fonctions du module math
# Ici, on a besoin de la fonction racine carrée sqrt(nombre)
recommencer=0 # Permet de redémarrer le programme lorsqu'il est fini
while recommencer!=1:
    print("Programme qui calcule les racines d'un polynome du second degré tel que Ax²+Bx+C=0")
    A=float(input('A=')) # On rentre la valeur de A
    B=float(input('B=')) # On rentre la valeur de B
    C=float(input('C=')) # On rentre la valeur de C
    delta=B*B-4*A*C # On calcule delta, le discriminant, en fonction de A,B et C
    print("Delta=",delta) # On affiche la valeur de delta
    if delta <0:
        print("Pas de solutions")# Lorsque delta est négatif, il n'y a pas de solutions
    if delta ==0:
        print("Une solution") # Lorsque delta est égale à  0, il y a une solution X
        x=-B/2*A # Calcul de X
        print("X=",x) # On affiche la solution
    if delta >0:
        print("Deux solutions") # Lorsque delta est positif,il y a deux solutions, X1 et X2
        racine_carre_delta=sqrt(delta) # On calcule la racine carrée de delta
        k=-B-racine_carre_delta # Variable qui va intervenir dans le calcul de X1
        l=-B+racine_carre_delta # Variable qui va intervenir dans le calcul de x2
        m=2*A # Variable qui va intervenir dans le calcul de X1 et X2
        x1=k/m # Calcul de X1
        x2=l/m # Calcul de X2
        print("X1=",x1) # On affiche la première solution
        print("X2=",x2) # on affiche la deuxième solution
        print("Voulez-vous recommencer ?")
    recommencer=int(input('0. Oui\n1. Non\n'))
print("Fin du programme!")
```

EXEMPLE N°3
RACINE D'UN TRINOME

Quelques remarques

Il sera bienvenu de modifier ce type de programme pour y apporter de la clarté :

- en recourant à la notion de fonction (cf. programme de seconde en vigueur) ;
- en utilisant, par exemple, des fonctions distinctes pour le calcul du discriminant et des racines ;
- éviter les commentaires inutiles ou superflus.

académie Reims

EXEMPLE N°3
RACINE D'UN TRINOME

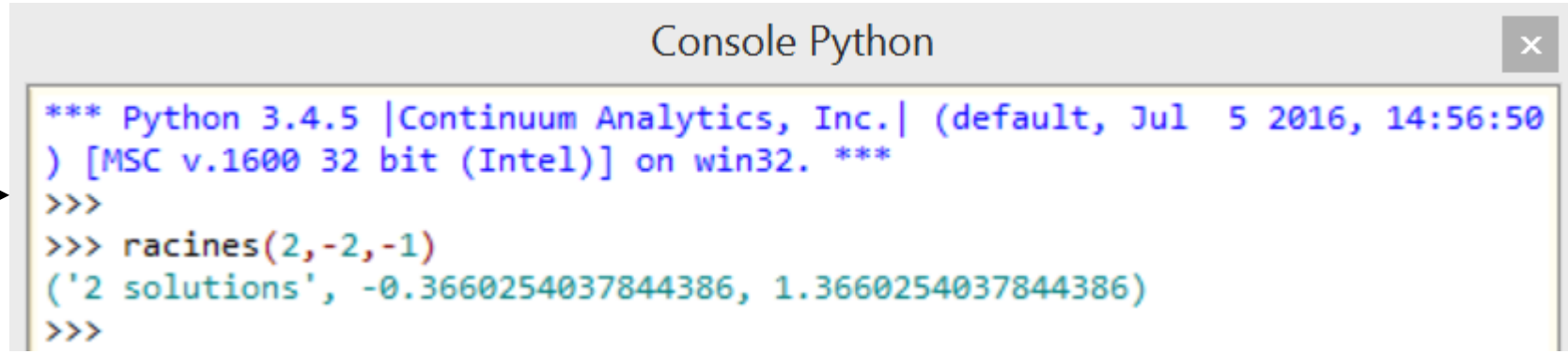
1ERES


```
1  from math import *
2
3  def delta(a,b,c):
4      assert a!=0
5      return(b*b-4*a*c)
6
7  def racines(a,b,c):
8      if delta(a,b,c)>0:
9          return('2 solutions',(-b-sqrt(delta(a,b,c)))/(2*a),(-b+sqrt(delta(a,b,c)))/(2*a))
10     elif delta(a,b,c)==0:
11         return('1 solution',-b/(2*a))
12     else:
13         return('pas de solution')
```

EXEMPLE N°3
RACINE D'UN TRINOME

Écrire un programme qui calcule le discriminant Δ puis en conséquence trouve les solutions éventuelles de l'équation.

Exécution du programme dans la console



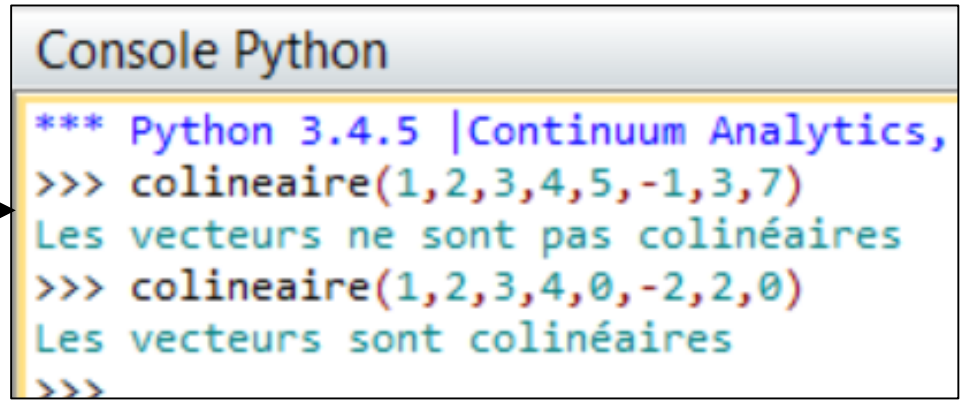
EXEMPLE N°4
COLINÉARITÉ DE DEUX VECTEURS

Connaissant les coordonnées des points A, B, C et D, déterminer si les vecteurs $\overrightarrow{AB}$ et $\overrightarrow{CD}$ sont colinéaires.

```
#Programme qui teste si deux vecteurs AB et CD sont colineaires :
def colineaire(xA,yA,xB,yB,xC,yC,xD,yD):
    m=xB-xA
    n=yB-yA
    p=xD-xC
    q=yD-yC
    if m*q-n*p==0 :
            return ("Les vecteurs sont colineaires")
    else :
            return ("Les vecteurs ne sont pas colineaires")
```

EXEMPLE N°4
COLINÉARITÉ DE DEUX VECTEURS

Exécution du programme dans la console, en appelant la fonction *colineaire()*



EXEMPLE N°5
DÉCOMPOSITION EN FACTEURS PREMIERS

Écrire un programme qui affiche la décomposition en facteurs premiers d'un nombre entier supérieur ou égal à deux.

Théorème fondamental de l'arithmétique

Tout nombre $n \geq 2$ se décompose, de façon unique, en produit de facteurs premiers $n = p_1^{r_1} \times p_2^{r_2} \times \cdots \times p_k^{r_k}$ où tous les nombres p_i sont des nombres premiers distincts avec $p_1 < p_2 < \cdots < p_k$ et les r_i sont des exposants entiers supérieurs ou égaux à 1.

EXEMPLE N°5

DÉCOMPOSITION EN FACTEURS PREMIERS

```python
# liste des facteurs premiers

def decomp_Factprem(n):
    assert ((type(n) == int) and n>0)
    d = 2
    while n>1:
        while n%d==0:  # % calcule le reste de division de n sur d
            n = n//d
            print("facteur trouvé:", d)
        d=d+1
```

Console Python

```
>>> decomp_Factprem(60)
facteur trouvé: 2
facteur trouvé: 2
facteur trouvé: 3
facteur trouvé: 5
>>>
```

WEBOGRAPHIE

WEBOGRAPHIE (NON EXHAUSTIVE) POUR PROGRESSER AVEC PYTHON

- Pour débuter avec Python au lycée : http://python.lycee.free.fr/
- Python pour débutant ou pour avancé : http://apprendre-python.com/
- Apprendre Python en vidéos (par Denis Sanson) : « Les pythonneries » http://www.dailymotion.com/playlist/x22t3u_universal_avs_apprendre-le-python/1#video=xbk6lj
- Des ressources sur le site Planète Maths de l'Académie de Grenoble.
- Site de Monsieur Hassnaoui (professeur de mathématiques dans l'Académie de Reims) : http://www.isn-ozanam.fr/
- Des exercices de base avec Python : proposés par l'IREM de la Réunion