



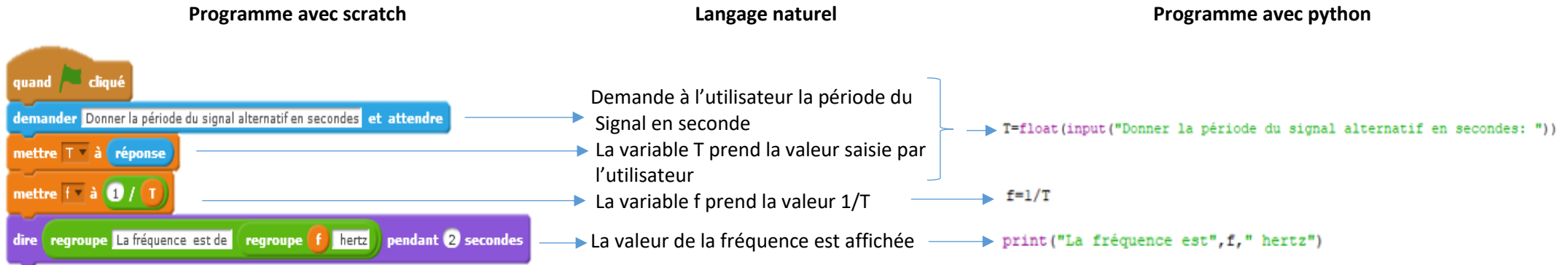
ATELIER PROGRAMMATION (PYTHON)

- Prise en main : commandes de base, boucles, listes, graphiques
- Capacités numériques dans les programmes de seconde et première spécialité voie générale

Formations Nouveaux programmes de lycée
Physique-Chimie (J2)
Mai - Juin 2019

QUELQUES COMMANDES POUR DEBUTER AVEC PYTHON

DECOUVERTE DU LOGICIEL PYTHON : Exemple de programme



LES COMMANDES UTILISEES :

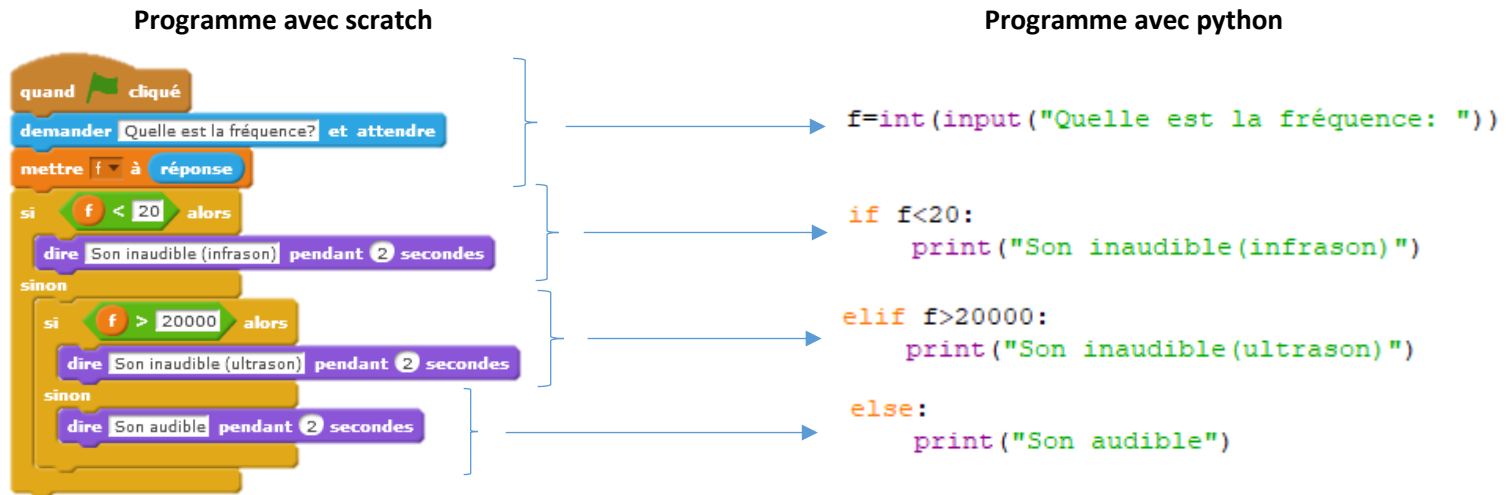
Que veut-on faire ?	Commande	Exemple	
Faire entrer un nombre entier	int(input())	Demander le nombre de personnes assistant à un spectacle de l'utilisateur et enregistrer le résultat donné comme étant la valeur de la variable a a=int(input("Combien de personnes ont assisté au spectacle ?"))	
Faire entrer un nombre réel	float(input())	Demander le prix d'une place de cinéma et enregistrer le résultat donné comme étant la valeur de la variable a a=float (input("Quel est le prix d'une place de cinéma ?"))	
Afficher une chaine de caractère	print()	a=5 print(a) → 5	a=5 print("a")→ a
		a=9 print(le carré de ",a, "est",a**2)	

DEFI 1 : Construire un programme permettant d'afficher la valeur du pH d'une solution à partir de la valeur de la concentration en ions [H₃O⁺] saisie par l'utilisateur

Remarque : le programme fait appel à l'utilisation du logarithme décimal. Celui-ci est contenu dans la bibliothèque MATH de python qu'il faut importer en début de programme.

Pour cela, il faut commencer le programme par la ligne : from math import*

LES COMMANDES CONDITIONNELLES : Exemple de programme



LES COMMANDES UTILISEES :

Que veut-on faire ?	Commande	Exemple
Conditions	if : Si elif : Sinon si else : Sinon	<pre>if x<5: x=x+3 elif x>5: x=x+5 else: x=x-1</pre>

DEFI 2 : Compéter le programme précédent (défi 1) afin qu'il nous indique si la solution est acide, basique ou neutre

2. LES BOUCLES AVEC PYTHON

EXEMPLE DE PROGRAMME : Programme permettant d'écrire le carré de tous les nombres allant de 1 à 10

```
for i in range(1,11):  
    print(i**2)
```

La variable i prend successivement les valeurs 1, 2, 3,10
Permet d'afficher toutes les valeurs de i^2

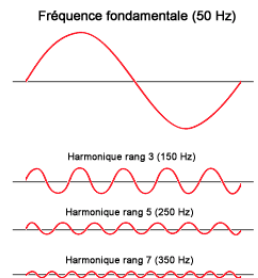
OU

```
i=1  
while i<11:  
    print(i**2)  
    i=i+1
```

On initialise la valeur de la variable i à 1
Tant que i est inférieur à 11
Affiche la valeur de i^2
Incréméte i de 1

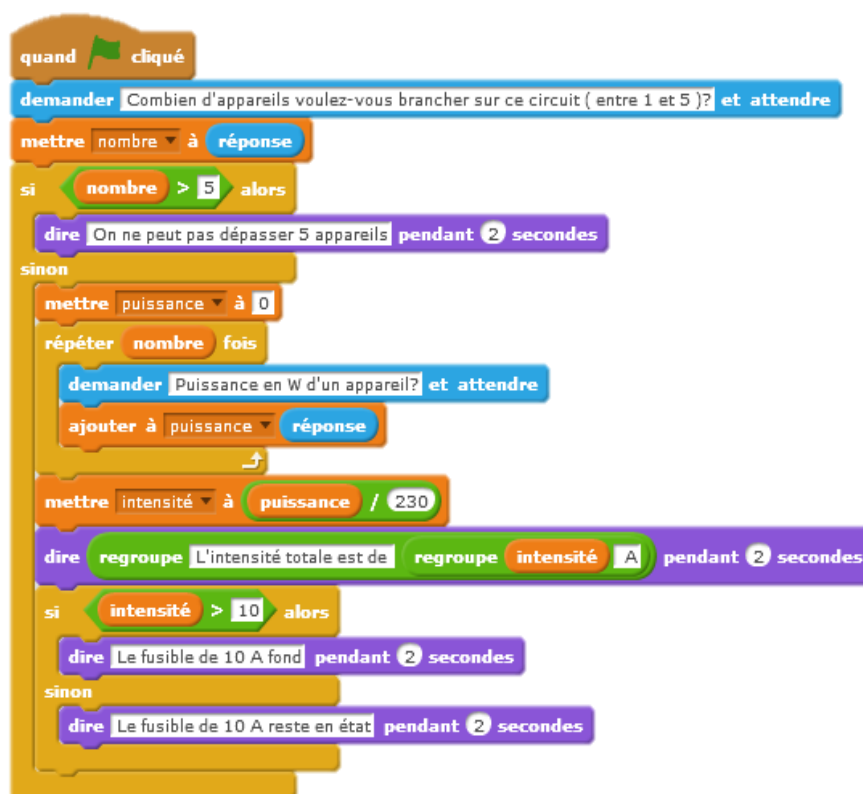
DEFI 3 : Un son est généralement un mélange de plusieurs sons différents. Il y a d'abord ce que l'on appelle la fondamentale (qu'on appelle aussi le premier harmonique), qui définit la hauteur du son et il se complète d'une multitude d'autres harmoniques. Il s'agit en fait de multiples de la fondamentale,

Pour une fondamentale f_0 Hz, on obtient $2f_0$ Hz pour le deuxième harmonique, $3f_0$ Hz pour le troisième, $4f_0$ Hz pour le quatrième et ainsi de suite...



Ecrire un programme donnant les 5 harmoniques d'une fréquence saisie par l'utilisateur.

DEFI 4 : Traduire en langage python le programme ci-dessous permettant d'indiquer l'état d'un fusible de 10 A après le branchement de plusieurs appareils.



3. Les listes avec python

Exemple de programme utilisant les listes : Calculer la moyenne de 3 notes

```

note=[] → Créer une liste vide nommée « note »
for i in range (0,3):
    x=float(input("Note? :"))
    note.append(x) → Boucle : demande les 3 notes
                    → Ajoute la note saisie dans la liste « note »
somme=sum(note) → Permet de faire la somme de tous les éléments contenus dans la liste « note »
total=len(note) → Permet d'obtenir le nombre d'éléments contenus dans la liste « note »
moyenne=(somme/total)
print(note) → Permet d'afficher la liste de notes
print(moyenne) → Permet d'afficher la moyenne des 3 notes
    
```

Quelques commandes sur les listes :

Que veut-on faire ?	Commande
Créer une liste vide nommée « valeur »	valeur=[]
Créer une liste nommée « valeur » contenant les éléments 1, 2 et 3	valeur=[1,2,3]
Obtenir la valeur située au rang n de la liste « valeur »	valeur[n-1] <i>Attention, les indices commencent à 0</i>
Obtenir le nombre d'éléments contenus dans la liste « valeur »	len(valeur)
Obtenir la somme de tous les éléments contenus dans la liste « valeur »	sum(valeur)
Ajouter l'élément n à la liste « valeur »	valeur.append(n)
Afficher la liste « valeur »	print(valeur)

DEFI 6 : Lors d'un TP dont le but est de déterminer la relation sur la planète Terre entre la masse et le poids d'un objet, un groupe d'élèves a obtenu les résultats suivants

Masse (kg)	0,01	0,05	0,1	0,2	0,5
Poids(N)	0,1	0,5	0,95	2,1	5

Créer un programme permettant de calculer le rapport P/m pour chaque mesure. Insérer les valeurs obtenues dans une liste nommée « rapport »

Si besoin, les différentes étapes à suivre :

1. Créer une liste « masse » contenant les valeurs du tableau de résultat.
2. Créer une liste « poids » contenant les valeurs du tableau de résultat.
3. Créer une liste vide nommée « rapport »
4. Créer une boucle permettant de :
 - Calculer chaque rapport P/m
 - Ajouter le résultat dans la liste « rapport »
5. Afficher la liste « rapport »

4. Les graphiques avec python

Pour réaliser des graphiques avec python, il faut importer la bibliothèque MATPLOTLIB

Pour cela, il faut inscrire en début de programme : **import matplotlib.pyplot as plt**

Exemple de programme : Placer dans un repère les points de coordonnées (1 ; 8), (2 ;10) et (3 ;12) :

<code>import matplotlib.pyplot as plt</code>	→ Permet d'importer la bibliothèque Matplotlib
<code>x=[1,2,3]</code>	→ Créer une liste contenant les valeurs des abscisses des points
<code>y=[8,10,12]</code>	→ Créer une liste contenant les valeurs des ordonnées des points
<code>plt.scatter(x,y)</code>	→ Permet de placer les points de coordonnées (x ;y)
<code>plt.show()</code>	→ Permet d'afficher le graphique

Quelques commandes sur les graphiques :

Que veut-on faire ?	Commande
Donner le titre « graph » à notre graphique	<code>plt.title("graph ")</code>
Nommer l'axe des abscisses « vitesse »	<code>plt.xlabel("vitesse")</code>
Nommer l'axe des ordonnées « temps »	<code>plt.ylabel("temps")</code>
Placer les points de coordonnées (x,y) et les relier <i>Les valeurs de x et de y ayant été définies en amont dans le programme</i>	<code>plt.plot(x,y)</code>
Placer le point de coordonnées (50 ;100)	<code>plt.scatter(50,100)</code>
Axe des abscisses de 0 à 100 Axe des ordonnées de 0 à 3	<code>plt.axis([0, 100, 0,3])</code>
Afficher le graphique	<code>plt.show()</code>

DEFI 7 : Lors d'un TP dont le but est de déterminer la relation sur la planète Terre entre la masse et le poids d'un objet, un groupe d'élèves a obtenu les résultats suivants

Masse (kg)	0,01	0,05	0,1	0,2	0,5
Poids(N)	0,1	0,5	0,95	2,1	5

Créer un programme permettant de placer les points de coordonnées (masse ; poids) dans un repère. Nommer le graphique et les deux axes.

- 1 - Recension des items « capacités numériques » des programmes de seconde et de première enseignement de spécialité :

Programme de seconde

Page 3

Capacité numérique : Représenter l'histogramme associé à une série de mesures à l'aide d'un tableur.

Page 9

Capacité numérique : représenter les positions successives d'un système modélisé par un point lors d'une évolution unidimensionnelle ou bidimensionnelle à l'aide d'un langage de programmation.

Page 10

Capacité numérique : représenter des vecteurs vitesse d'un système modélisé par un point lors d'un mouvement à l'aide d'un langage de programmation.

Page 13

Capacités numériques : représenter un nuage de points associé à la caractéristique d'un dipôle et modéliser la caractéristique de ce dipôle à l'aide d'un langage de programmation.

Programme de spécialité physique-chimie de première

Page 4

Capacité numérique : Représenter l'histogramme associé à une série de mesures à l'aide d'un tableur.

Page 6

Capacité numérique : Déterminer la composition de l'état final d'un système siège d'une transformation chimique totale à l'aide d'un langage de programmation.

Page 11

Capacité numérique : Utiliser un langage de programmation pour étudier la relation approchée entre la variation du vecteur vitesse d'un système modélisé par un point matériel entre deux instants voisins et la somme des forces appliquées sur celui-ci.

Page 13

Capacité numérique : Utiliser un langage de programmation pour effectuer le bilan énergétique d'un système en mouvement.

Page 14

Capacités numériques : Représenter un signal périodique et illustrer l'influence de ses caractéristiques (période, amplitude) sur sa représentation. Simuler à l'aide d'un langage de programmation, la propagation d'une onde périodique.

- 2 - Analyse et appropriation de la mise en œuvre du langage python.

Programme de seconde :

Représenter les positions successives d'un système modélisé par un point lors d'une évolution unidimensionnelle ou bidimensionnelle à l'aide d'un langage de programmation.

- Ouvrir le script **Seconde_representation_positions_reduit.py** et compléter la ligne 10.
- Exécuter ce script en appuyant sur la touche F5 du clavier.
- Ouvrir le script **Seconde_positions_animation.py** et l'exécuter.
- Les lignes 15 à 17 de ce script construisent les listes des abscisses x et des ordonnées différemment par rapport au script précédent : par comparaison, identifier les différences ; en particulier, que signifie l'instruction : `x = position[i][1]` ?
- Quel est le rôle des lignes 8 à 12 ?
- Quelle instruction permet le positionnement successif des points ?

Représenter des vecteurs vitesse d'un système modélisé par un point lors d'un mouvement à l'aide d'un langage de programmation.

- Ouvrir le script **Seconde_vect_vitesse_reduit.py**
- Compléter, en vous inspirant des scripts précédents, les lignes 9 à 11 pour construire les listes x et y
- La fonction `quiver` de la bibliothèque `matplotlib.pyplot` mise en œuvre ligne 23 requiert des paramètres (éléments figurant dans la parenthèse suivant le nom de la fonction) : repérer le rôle de ces paramètres en tentant de les modifier.

Représenter un nuage de points associé à la caractéristique d'un dipôle et modéliser la caractéristique de ce dipôle à l'aide d'un langage de programmation.

- Ouvrir le script **Seconde_nuage_points_modelisation.py** et l'exécuter.
- Quel est le rôle des instructions lignes 5 et 6 ?
- La fonction `input()` permet de saisir une donnée en ligne de commande (shell) : dans quelle variable est enregistrée la première donnée saisie ?
- Sachant que la fonction `input()` retourne une chaîne de caractères, qu'est-ce qui justifie l'usage de la fonction `float()` dans laquelle est placée la fonction `input` lignes 10 et 11 ?
- `while "condition"` est une boucle conditionnelle, ie qui s'exécute tant que "condition" est vrai. Analyser la ligne 9, quel événement met un terme à cette boucle ?
- A quelle bibliothèque appartient la fonction `scatter` de la ligne 18 et quel est son rôle ?
- A quelle bibliothèque appartiennent les fonctions des lignes 19 et 20 ? Rechercher leurs rôles et paramètres sur internet.
- Quelle fonction permet le calcul de la régression linéaire ?
- Expliquer le rôle des paramètres de la fonction `plot` ligne 22.
- Modifier ce script pour l'adapter à un autre exercice nécessitant le tracé d'un nuage de points.

Programme de première

Utiliser un langage de programmation pour étudier la relation approchée entre la variation du vecteur vitesse d'un système modélisé par un point matériel entre deux instants voisins et la somme des forces appliquées sur celui-ci.

- Ouvrir le script **Premiere_trace_vect_acquisition_reduit.py**
- En vous inspirant des lignes 19 à 21, écrire une boucle lignes 23 à 25 permettant de compléter les listes des accélérations des ax et ay et tester votre script.
- Placer lignes 31 et 32 les instructions permettant le tracé des vecteurs accélération.
- Ajouter un paramètre à la fonction de tracé pour représenter en rouge les vecteurs accélération.
- Ouvrir le script **Premiere_trace_vect_parabole.py** et l'exécuter (c'est bien plus beau non ?)
- Qu'est-ce qui distingue ce script du précédent ?
- Analyser les instructions créant les listes x et y

Déterminer la composition de l'état final d'un système siège d'une transformation chimique totale à l'aide d'un langage de programmation.

Soit le bilan chimique : $C_2H_6 + 3O_2 = CO_2 + CO + 3H_2O$

état initial : $n_{(C_2H_6)} = 2 \text{ mol}$ $n_{(O_2)} = 3 \text{ mol}$

- Ouvrir le script **Seconde_bilan_chimique.py** et l'exécuter en décrivant le système ci-dessus.
- Vérifier que le résultat fourni par le programme est correct.
- Tester ce script pour un autre système chimique.
- Analyser étape par étape les instructions composant ce script ; ce travail est facilité par l'ajout de print() affichant les résultats des calcul intermédiaires.
- Le script **Seconde_bilan_chimique_histo.py** est complété avec un tracé format histogramme du bilan final

Pour vous aider, quelques écailles du serpent python :

Le premier élément d'une liste a pour indice 0. Exemple si $L=[4,8,6]$ alors $L[0]$ vaut 4

$L=[0]*n$ crée une liste contenant n fois la valeur 0

Si L1 et L2 sont deux listes, alors $L=L1+L2$ crée une liste L composée des éléments de L1 puis de ceux de L2

Si chaîne1 et chaîne2 sont deux chaînes de caractères, alors chaîne=chaîne1+chaîne2 concatène ces deux chaînes en une seule

Si a est une variable numérique, alors str(a) transforme cette valeur numérique en une chaîne de caractères (type string)

round(x,2) arrondit la valeur de la variable x à deux décimales

- Question subsidiaire : adapter ce script pour traiter le cas des systèmes comportant des produits dans l'état initial.

- 3 - Acquisition de données via l'application PHYPHOX.

- Effectuer l'acquisition d'une grandeur (champ magnétique, accélération, luminosité ...) et exporter ces mesures via une messagerie.

- 4 - Histogrammes et tableur

Représenter l'histogramme associé à une série de mesures à l'aide d'un tableur.

- Traiter les mesures acquises au point – 3 -