

Étude des vecteurs vitesses lors d'une chute libre

Niveau (Thèmes)	BTS physique/info
Introduction	On étudie le mouvement d'une bille en chute libre (parabolique). Un travail informatique consiste à gérer les données, un autre à les tracer.
Type d'activité	TP de physique informatisée
Compétences	<p>Exemples d'intitulé de compétences :</p> <p>RESTITUER SES CONNAISSANCES</p> <p>S'APPROPRIER :</p> <ul style="list-style-type: none"> - Relier la situation/le problème à des informations fournies <p>ANALYSER :</p> <ul style="list-style-type: none"> - Relier différents types de représentation - Faire une estimation - Repérer ou sélectionner des informations utiles <p>RÉALISER</p> <ul style="list-style-type: none"> - écrire un résultat de façon adaptée <p>VALIDER</p> <ul style="list-style-type: none"> - Discuter de la validité d'une information
CRCN - Compétences Num.	3.3 - 3.4
Notions et contenus du programme	Mécanique
Objectif(s) pédagogique(s)	Traiter des données brutes. En extraire une information physique en adéquation avec un modèle mécanique. Utiliser des graphiques pour valider un modèle.
Objectifs disciplinaires et/ou transversaux	
Description succincte de l'activité	L'élève doit extraire les informations contenues dans un fichier texte, obtenu par un logiciel de pointage. Puis il doit effectuer des calculs, pour afficher les grandeurs souhaitées. Enfin, il compare au modèle de la chute libre
Découpage temporel de la séquence	10' : présentation activité 30' : traitement de la donnée brute 20' : affichage des graphiques 30' : modélisation et finalisation.
Pré-requis	une bonne connaissance de l'environnement python
Outils numériques utilisés/Matériel	python/matplotlib
Gestion du groupe Durée estimée	1H30.

Énoncé à destination des élèves

L'énoncé est contenu dans le fichier python à compléter avec l'éditeur python disponible. Fichier "traitement_donnees.py"

Manipulation de chaînes en python

Condition d'existence d'une chaîne dans une autre chaîne :

"toto" est-il dans la chaîne "tititatatoto" : "toto" in "tititatatoto"

Ce qui permet de faire des recherches comme : Si la chaîne ne contient pas "#".

caractère de séparation de partie

on veut séparer "toto tata" pour récupérer ["toto", "tata"]. La méthode utilisée est "strip('caractère de séparation')"

Exemple : "toto tata".strip(' ') séparera sur l'espace. Pour séparer sur une tabulation, le caractères est '\t'.

Caractères de fins de lignes

Pour éviter que python prenne en compte les fins de lignes, il lui dire de les enlever. Ceci se fait avec "rstrip()":

"chaîne\n".rstrip() renverra "chaîne" car \n est un caractère de fin de ligne.

Commentaire à destination des enseignant.e.s

Correction partie 1 :

```
for ligne in lignes[1:] :
    try : # nécessaire pour certaines lignes avec des caractères
        t, x, y = ligne.split('\t') # '\t' correspond au caractère "tabulation"
        listeTemps.append(float(t.replace(',', '.').rstrip('\n')))
        listeX.append(float(x.replace(',', '.').rstrip('\n')))
        listeY.append(float(y.strip().replace(',', '.').rstrip('\n')))
    except : pass
```

partie 2 :

```
plt.figure(1)
plt.plot(listeTemps, listeY, label = "ordonnées", '+')
plt.plot(listeTemps, listeX, label = "abscisses", '+')
plt.legend()
plt.show()
```

partie 3 :

```
vitessesX.append(echelleVitesses*(listeX[i+1]-listeX[i-1])/(listeTemps[i+1]-listeTemps[i-1]))
vitessesY.append(echelleVitesses*(listeY[i+1]-listeY[i-1])/(listeTemps[i+1]-listeTemps[i-1]))
vitesses.append(math.sqrt(vitessesY[i-1]**2+vitessesX[i-1]**2)) # on calcule la norme si besoin
```

partie 4 :

```
plt.figure(2)
plt.plot(vitesses, listeY[1:-1], label = "ordonnées", '+')
plt.legend()
plt.show()
```

Retour d'expérience :

Les plus-value pédagogiques (enseignants/élèves) :

Les freins :

Les leviers :

Les pistes pour aller plus loin ou généraliser la démarche :

Production d'élèves :

mettre lien, extrait de copies etc en s'assurant d'avoir les droits de diffusion auprès des élèves