

Objectif de formation

Favoriser l'appropriation du langage Python en partant de programmes simples à modifier et à complexifier.

Obstacles prévisibles

Groupe hétérogène : Certains collègues peuvent ne pas connaître du tout le langage de programmation. D'autres peuvent avoir une maîtrise déjà avancée.

On essaie d'envisager une différenciation, en particulier au démarrage de l'activité pour faciliter l'appropriation des uns et permettre aux autres d'aller plus rapidement.

Enseignement du numérique : Certains collègues peuvent penser qu'il faut être un spécialiste du numérique pour utiliser un langage de programmation

On propose une mise en situation élève et on montre qu'on peut partir d'un programme créé par d'autres et le faire évoluer sans être un spécialiste du numérique. On utilise les compétences numériques des élèves voire on facilite leur appropriation.

On ne ferait pas de physique : Certains collègues peuvent penser qu'utiliser le numérique, c'est faire des choses en plus au lieu de faire de la physique.

La situation proposée cherche à faire travailler les élèves sur la compétence **Valider**. Le programme n'est qu'un outil pour travailler cette compétence. On utilise le numérique « comme on utilise les maths ».

Présentation du support choisi

On propose au stagiaire de jouer le rôle d'un élève et on lui propose une situation mettant en œuvre un programme en langage Python au service de l'acquisition de compétences, en particulier la compétence **Valider**.

L'élève doit réaliser les mesures (ici les mesures sont données) puis choisir le modèle pertinent pour modéliser le phénomène : 4 modèles sont proposés.

Le support pédagogique est divisé en deux parties principales. À gauche, un document de travail intitulé 'Modélisation de la réfraction' pour la classe de Seconde. Il contient des schémas de réfraction à l'interface air-eau, des questions de compréhension et des données de mesure. À droite, quatre programmes Python sont présentés, chacun correspondant à un niveau de difficulté (A, B, C, 4). Les programmes A, B et C utilisent des entrées fixes ou des listes de valeurs, tandis que le programme 4 utilise des entrées utilisateur. Les programmes A, B et C contiennent des erreurs de calcul ou de logique que l'élève doit identifier et corriger.

Modélisation_refraction_1_A : Niveau A

```
1 from math import *
2 print('Programme pour calculer la valeur de l'indice de réfraction "n2"')
3 print('Quel est l'indice du milieu du rayon incident "i1" ?')
4 n1=1.0
5 n2=1.5
6 print('Quel est l'angle d'incidence "i1" en degré ?')
7 i1=float(input("i1="))
8 i1=i1*pi/180
9 print('Quel est l'angle de réfraction "i2" en degré dans ce cas ?')
10 i2=float(input("i2="))
11 i2=i2*pi/180
12 n2=n1*i2/i1
13 print('')
14 print('Dans ce cas, on a : n2 =',n2)
15 print('Tu te charges d'ajuster le nombre de chiffres significatifs')
16 print('Essais avec un autre couple de valeur...')
```

Modélisation_refraction_1_B : Niveau B

```
1 from math import *
2 print('Exercice le bon modèle ?')
3 n=[1.0,1.3,1.4,1.5,1.6]
4 i=[1.0,1.3,1.4,1.5,1.6]
5 i1=float(input("i1="))
6 i1=i1*pi/180
7 print('Quel est l'indice du milieu du rayon incident "i1" ?')
8 n1=float(input("n1="))
9 for i in n:
10     print('Quand i1 =',i, 'la valeur de n2 est :',n2)
11     i2=float(input("i2="))
12     i2=i2*pi/180
13     n2=n1*i2/i1
14     print('')
15     print('Pour la mesure n°',i, 'la valeur de n2 est :',n2)
16 print('Tu te charges d'ajuster le nombre de chiffres significatifs')
17 print('Alors, les résultats sont-ils satisfaisants ?')
```

Modélisation_refraction_1_C : Niveau C

```
1 from math import *
2 print('Exercice le bon modèle ?')
3 i=[1.0,1.3,1.4,1.5,1.6]
4 n=[1.0,1.3,1.4,1.5,1.6]
5 print('Quel est l'indice du milieu du rayon incident "i1" ?')
6 i1=float(input("i1="))
7 for i in range(len(i)):
8     print('Quand i1 =',i, 'la valeur de n2 est :',n2)
9     i2=float(input("i2="))
10     i2=i2*pi/180
11     n2=n1*i2/i1
12     print('')
13     print('Pour le couple de mesure n°',i, 'la valeur de n2 est :',n2)
14 print('Tu te charges d'ajuster le nombre de chiffres significatifs')
15 print('Alors, les résultats sont-ils satisfaisants ?')
```

Modélisation_refraction_4

```
1 from math import *
2 print('Utilisation de la loi de Snell-Descartes')
3 print('Pour déterminer l'indice de réfraction, taper indice')
4 print('Pour déterminer l'angle de réfraction, taper angle')
5 n=input('n =')
6 if n=="indice":
7     print('Il faut donner les valeurs de n1, i1 et i2 pour calculer n2')
8 else:
9     if n=="angle":
10         print('Il faut donner les valeurs de n1, i1 et i2 pour calculer i2')
11     else:
12         print('Valeur non valide, au revoir')
```

On prévoit 3 « niveaux » de programme pour entrer dans l'activité avec les stagiaires :

- Niveau A : programme à ouvrir « Modelisation_refraction_1_A »
programme sans boucle – variables simples – calcul sur les variables
- Niveau B : programme à ouvrir « Modelisation_refraction_1_B »
programme avec boucle – liste explicite – calcul sur les listes
- Niveau C : programme à ouvrir « Modelisation_refraction_1_C »
programme avec boucle – quelques instructions non explicites

Le premier programme (1A, 1B ou 1C) utilise un modèle manifestement incorrect : $n_1 \times i_2 = n_2 \times i_1$. Il faut identifier, en lisant le programme, le modèle utilisé 1, 2, 3 ou 4. Pour cela, il faut passer de la forme : $n_2 = n_1 \cdot i_2 / i_1$ à : $n_1 \times i_2 = n_2 \times i_1$ donc mettre en œuvre un calcul littéral.

Pour tester les autres modèles, il faut transformer le programme, et donc passer :

- de la forme : $n_1 \times i_1 = n_2 \times i_2$ à la forme : $n_2 = n_1 \cdot i_1 / i_2$
- de la forme : $n_1 \times \sin(i_1) = n_2 \times \sin(i_2)$ à la forme : $n_2 = n_1 \cdot \sin(i_1) / \sin(i_2)$
- de la forme : $n_1 \times \tan(i_1) = n_2 \times \tan(i_2)$ à la forme : $n_2 = n_1 \cdot \tan(i_1) / \tan(i_2)$

Pour valider un modèle, on doit vérifier qu'il est pertinent pour l'ensemble des valeurs (ou alors on peut lui donner un domaine de validité, pour les petits angles pour les modèles 1 et 4 par exemple).

Remarques : on a choisi de convertir l'angle de ° en rad dans le programme initial pour éviter la difficulté de la conversion à la première lecture.

on a choisi de calculer n_2 , dans un premier temps, et non pas i_2 pour éviter d'avoir la difficulté de l'arcsinus et la difficulté de la conversion dans un sens puis dans l'autre.

Dans un deuxième temps, l'élève peut se confronter à la modification du programme pour en faire un programme qui utilise le modèle pour déterminer un angle de réfraction connaissant les indices et l'angle d'incidence.