| Cycles de vie des produits avec Python | 9e année – Chimie |
|---|---|

# Plan de leçon

| | |
|---|---|
| Outil de programmation | Python |
| Compétences transversales | Programmation |

**Idées générales**

**C1. Relier la science à notre monde en changement**
évaluer les impacts sociaux, environnementaux et économiques de l'utilisation des éléments, des composés et des technologies associées

**A1. Compétences en investigation STEM**
appliquer des processus scientifiques et un processus de conception en ingénierie dans leurs enquêtes pour développer une compréhension conceptuelle de la science qu'ils apprennent, et appliquer des compétences en codage pour modéliser des concepts et des relations scientifiques

**Attentes précises**

**C1.1** analyser l'incidence sur la société, l'économie et l'environnement des procédés associés au cycle de vie des produits de consommation, en tenant compte des éléments et des composés dont ils sont constitués, et suggérer des façons d'en accentuer les effets positifs et d'en minimiser les effets négatifs.

**C1.2** examiner l'incidence de l'utilisation de technologies émergentes en chimie dans divers secteurs d'activité, y compris ceux associés aux métiers spécialisés, et décrire des facteurs qui influent sur le développement de ces technologies.

## Description

Ce programme aide les étudiants à comprendre comment les produits sont fabriqués, utilisés et recyclés en combinant la science et la programmation. Les étudiants utilisent Python pour modéliser les cycles de vie des produits et explorer de nouvelles technologies chimiques, tout en développant des compétences de base en codage de manière pratique et concrète.

## Matériel

- Ordinateur
- Outil basé sur des graphes
- IDE Python (Visual Studio)

## Aptitudes en pensée computationnelle

- Variables
- Boucles
- Visualisation des données
- Simulations simples

## Introduction

Les produits du quotidien—comme les bouteilles d'eau, les téléphones et les vêtements—sont le résultat de choix chimiques concernant les matériaux, l'énergie et la manière dont quelque chose peut être réutilisé ou recyclé. Dans cette leçon, les élèves explorent le « cycle de vie » d'un produit, de l'extraction des matières premières à l'élimination, et considèrent les impacts sociaux, environnementaux et économiques qui apparaissent à chaque étape.

Les étudiants utiliseront ensuite Python dans Visual Studio Code pour modéliser et visualiser les données du cycle de vie, en développant des compétences de base en codage (variables, boucles et analyse de données simple) tout en formulant des recommandations fondées sur des preuves pour réduire l'impact!

## Action

**Activité d'échauffement :**

Demandez aux élèves de choisir un produit courant et quotidien, comme une bouteille d'eau en plastique, un téléphone intelligent ou un T-shirt en coton. Ensuite, enquêtez ensemble sur son cycle de vie complet.

Votre cours commence par l'identification des matériaux utilisés pour fabriquer le produit, en tenant compte de leurs sources naturelles et de la manière dont ils sont extraits ou récoltés. Les étudiants recherchent l'origine des matières premières, les processus de fabrication impliqués et l'énergie nécessaire à la production.

Ils devraient également discuter de la manière dont le produit est transporté, vendu et finalement utilisé par les consommateurs, en notant tous les facteurs qui affectent sa durée de vie.

Enfin, demandez aux élèves d'explorer ce qui arrive au produit après sa durée de vie utile, en examinant les options de réutilisation, de recyclage ou d'élimination, et en tenant compte des impacts environnementaux à chaque étape.

**Modélisation Python :**

**Étape 1 : Installer Python**

Python est le langage de programmation utilisé pour cette leçon. Voici comment l'installer :

- Visitez le site officiel de Python : https://www.python.org/
- Cliquez sur Télécharger Python (choisissez la dernière version pour votre système d'exploitation) : Windows, macOS ou Linux).
- Exécutez l'installateur. Important : Sous Windows, cochez la case « Ajouter Python au PATH » avant de cliquer sur « Installer maintenant ».
- Suivez les instructions pour compléter l'installation.
- Pour confirmer l'installation, ouvrez l'invite de commandes de votre ordinateur (Windows) ou le Terminal (macOS/Linux) et tapez python

– version. Vous devriez voir un numéro de version (par exemple, Python 3.10.0).

**Étape 2 : Installer Visual Studio Code (VS Code)**

VS Code est un éditeur gratuit et convivial pour écrire et exécuter du code Python.

- Allez sur le site Web de VS Code : https://code.visualstudio.com/
- Téléchargez et installez VS Code pour votre système d'exploitation.
- Une fois installé, ouvrez VS Code.

**Étape 3 : Configurer un nouveau fichier Python dans VS Code**

Ouvrez VS Code.

- Cliquez sur Fichier > Nouveau fichier (ou Fichier > Ouvrir un dossier pour créer un dossier pour votre projet).
- Enregistrez votre nouveau fichier sous le nom product_lifecycle.py (ou tout autre nom se terminant par. py).
- Assurez-vous d'avoir installé l'extension Python dans VS Code. Si une invite apparaît, cliquez sur « Installer » ou recherchez « Python » dans la barre latérale des extensions et installez l'extension Python de Microsoft.

**Étape 4 : Installer la bibliothèque matplotlib**

Le code utilise matplotlib pour créer des graphiques. Voici comment l'installer :

- Dans VS Code, cliquez sur Affichage > Terminal pour ouvrir une fenêtre de terminal en bas de l'écran.
- Tapez la commande suivante et appuyez sur Entrée :
- pip install matplotlib
- Attendez que l'installation soit terminée. Vous devriez voir des messages indiquant le succès.

**Étape 5 : Copiez et collez le code Python fourni**

Copiez tout le code fourni dans le plan de leçon ci-dessous ces instructions.

- Collez-le dans votre fichier .py dans VS Code.
- Enregistrez votre fichier (Fichier > Enregistrer ou appuyez sur Ctrl+S).

**Étape 6 : Exécuter le code Python dans VS Code**

Avec votre fichier. py ouvert, cherchez un petit bouton triangulaire « Exécuter » en haut à droite (ou faites un clic droit dans l'éditeur et sélectionnez « Exécuter le fichier Python dans le terminal »).

Le programme commencera à s'exécuter dans la fenêtre du terminal ci-dessous.

**Étape 7 : Interagir avec le programme et voir les résultats**

Le programme affichera une liste de matériaux (par exemple, plastique, verre, aluminium) et vous demandera d'en choisir un, de saisir « personnalisé » ou de taper « quitter ».

- Si vous sélectionnez un matériau, vous verrez des informations à son sujet ainsi qu'un graphique montrant la consommation d'énergie à travers les

différentes étapes de son cycle de vie (par exemple, extraction, fabrication, distribution, utilisation, élimination/recyclage).

- Si vous choisissez « personnalisé », vous pouvez saisir votre propre matériau, ajouter une description et entrer les valeurs d'énergie pour chaque étape.
- Après avoir consulté le tableau, le programme vous demandera si vous souhaitez rechercher un autre matériau ou quitter.

**Étape 8 : Dépannage des problèmes courants**

- Python non trouvé : Assurez-vous d'avoir installé Python et de l'avoir ajouté au PATH lors de l'installation.
- Matplotlib non installé : Vérifiez bien que vous avez exécuté pip install matplotlib dans le terminal de VS Code.
- Erreurs de code : Confirmez que vous avez copié le code exactement. Les fautes de frappe ou les lignes manquantes peuvent causer des problèmes.
- Le programme ne s'exécute pas : Assurez-vous d'exécuter le bon fichier et d'avoir l'extension Python installée.
- Le graphique n'apparaît pas : Parfois, les graphiques s'ouvrent dans une fenêtre séparée; vérifiez derrière d'autres fenêtres ou votre barre des tâches.

Dans le code d'exemple ci-dessous, nous avons plusieurs matériaux « préprogrammés », ces matériaux sont des espaces réservés pour donner à vos étudiants une idée de la façon de formater leurs résultats. Une fois ce code écrit, les étudiants devraient être encouragés à ajouter leurs propres produits ou matériaux avec leurs propres données pour afficher la quantité de MJ ou de Mega Jules d'énergie utilisée à chaque étape de leurs matériaux ou produits.

**Code d'exemple :**

```
# Import the matplotlib library for plotting graphs
import matplotlib.pyplot as plt



# Preprogrammed materials and their data
materials = {
    # Each material is a dictionary entry with info and energy usage per stage
```

```
    "Plastic": {
        "info": "Plastic is made from petroleum. It is lightweight but can take hundreds of years to
decompose.",
        "energy": [10, 25, 5, 2, 8]  # Energy used (in MJ) for each lifecycle stage
    },
    "Glass": {
        "info": "Glass is made from sand. It is recyclable and can be reused many times.",
        "energy": [15, 20, 7, 2, 5]
    },
    "Aluminum": {
        "info": "Aluminum is made from bauxite ore. It is highly recyclable and saves energy when
reused.",
        "energy": [20, 30, 10, 3, 4]
    }
}



# List of product lifecycle stages, used for both chart labels and user input
stages = ["Raw Material Extraction", "Manufacturing", "Distribution", "Usage",
"Disposal/Recycling"]



# Main menu function for user interaction
def main_menu():
    while True:  # Loop until the user chooses to exit
        print("\nAvailable materials:")
        # Print each material name
        for name in materials:
            print(f"- {name}")
        print("- custom (enter your own material)")
        print("- exit (quit the program)")

        # Prompt user to select material, custom, or exit
        choice = input("Type the name of a material from the list above, 'custom' to enter your own,
or 'exit' to quit: ").strip()

        if choice.lower() == "exit":  # Exit condition
            print("Goodbye!")
            break
```

```
    elif choice in materials:  # If user selects a preprogrammed material
        info = materials[choice]["info"]
        energy = materials[choice]["energy"]
        material = choice
    elif choice.lower() == "custom":  # If user wants to enter a custom material
        material = input("Enter the material name: ")
        info = input(f"Enter a brief description for {material}: ")
        energy = []
        print(f"Enter the energy used (in MJ) for each stage of {material}:")
        # Prompt for energy value for each lifecycle stage
        for stage in stages:
            value = float(input(f"  {stage}: "))
            energy.append(value)
    else:  # If input is invalid
        print("Invalid choice. Please try again.")
        continue

    # Display selected material and its info
    print(f"\nMaterial: {material}")
    print(f"Info: {info}")

    # Plot energy usage for each lifecycle stage using a bar chart
    plt.bar(stages, energy, color='skyblue')
    plt.xlabel('Lifecycle Stage')
    plt.ylabel('Energy Used (MJ)')
    plt.title(f'Energy Use Across Product Lifecycle: {material}')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

    # Ask if the user wants to see another material
    again = input("Would you like to look up another material? (yes/no): ").strip().lower()
    if again != "yes":  # Exit loop if user says no
        print("Goodbye!")
        break



# Run the menu function if script is executed directly
if __name__ == '__main__':
    main_menu()
```

**Consolidation et approfondissement**

Les étudiants peuvent maintenant utiliser ce code pour ajouter toutes sortes de matériaux et calculer leurs émissions sur le cycle de vie.

**Évaluation**

Ajouter plus de matériaux et conserver des notes de recherche sur les matériaux fournissent d'excellents matériaux pour l'évaluation.

**Ressources supplémentaires**

https://www.python.org/
https://calculator.dev/environment/lca-calculator/
https://code.visualstudio.com/