## **Exercise 1: Working with Pandas**

### **Analyzing Chemical Formulas from a CSV File**

In this exercise, you'll learn how to read data from a CSV file named MP\_data.csv using Pandas, a powerful data manipulation library in Python. Your task is to analyze the composition column in the dataset to determine the number of binary, ternary, and quaternary compounds it contains.

**Objectives:**

1. Load the dataset from MP\_data.csv.
2. Count and categorize the compounds based on the number of distinct elements they contain:
   * **Binary compounds** have exactly two different elements.
   * **Ternary compounds** have exactly three different elements.
   * **Quaternary compounds** have exactly four different elements.

**Hint:** To find the number of distinct elements in a chemical formula, use the Composition class from Pymatgen and get the length of the elements attribute like this:

num\_elements = len(Composition(formula).elements).

## **Exercise 2: Data Visualization with Matplotlib**

### **Creating a Bar Chart of Space Group Frequencies**

In this exercise, you will practice your data visualization skills by generating a bar chart that represents the frequency of different space groups in a dataset. You will use Python's Matplotlib library, a powerful tool for creating a wide range of static, animated, and interactive visualizations.

**Objectives:**

1. Read data from a CSV file MP\_data.csv using Pandas.
2. Extract the spacegroup\_number column to analyze the frequency of each space group.
3. Plot the results using a bar chart where:
   * The x-axis represents the space group numbers.
   * The y-axis shows the count of occurrences for each space group in the dataset.

**Hint:** Use value\_counts() on the spacegroup\_number column to easily count how many times each space group appears, and then use plt.bar() to create the bar chart.

## **Exercise 3: Analyzing Chemical Compositions with Pymatgen**

### **Identifying Complex Lithium-containing Materials**

This exercise involves using the Pymatgen library, a powerful tool for materials analysis in Python. You will practice reading chemical data from a CSV file and using Pymatgen to derive detailed composition information.

**Objectives:**

1. Load a CSV file containing chemical formulas.
2. Use Pymatgen to extract the full chemical formula and the number of distinct elements for each compound.
3. Identify and count all materials that contain Lithium and have more than three different elements.

**Hint:** Check for the presence of Lithium in each formula by using 'Li' in comp.formula, where comp is a Composition object created from each formula. This method checks if 'Li' is part of the formula string, ensuring accurate detection of Lithium in the composition. Count the elements as shown in the provided code to identify compositions with more than three elements.

## **Exercise 4: Handling Crystallographic Information Files with Pymatgen**

### **Analyzing Structures from CIF Files**

This exercise leverages the Pymatgen library's capabilities to process crystallographic information files (CIFs). You will gain hands-on experience with loading structural data and analyzing it to extract meaningful information.

**Objectives:**

1. Access and load structures from five CIF files located in a specified directory (../cifs/).
2. For each structure, determine the total count of distinct atomic numbers and element types.
3. Compile the results into a Pandas DataFrame for easy viewing and analysis.

**Hint:** Use Structure.from\_file(os.path.join(folder\_path, file\_name)) to load each CIF file. After loading, create a list of atomic numbers for each element in the structure using a list comprehension: [element.Z for element in structure.composition.elements].

## **Exercise 5: Visualization of Crystal Structures with VESTA and Python**

### **Exploring Crystal Structures Using VESTA and Script-based Visualization**

This exercise combines hands-on practice with VESTA, a software for visualizing crystal structures, and a Python script for generating visualizations of CIF files. You'll identify specific crystal structures and learn to automate the visualization process.

**Objectives:**

1. Manually visualize five CIF files using VESTA to identify two cubic materials. Take screenshots of the identified cubic structures.
2. Optionally, use a provided Python script to automate the visualization of a list of CIF files from a folder, enhancing your skills in script-based data processing and visualization.

**Part 1: Manual Visualization with VESTA**

* Open each of the five CIF files in VESTA.
* Examine the structure and symmetry information to determine if they are cubic.
* Capture and save screenshots of the two cubic structures you identify.

**Part 2: Automated Visualization with Python (Optional)**

* Utilize the visual\_cif.py script provided in the codes folder to generate structure images for a list of CIF files.
* Modify the script as needed to suit your specific dataset or visualization preferences.

**Hints:**

* **VESTA:** Open VESTA, go to 'File' > 'Open' and select your CIF file. To identify cubic materials, check the symmetry or space group listed in the properties.
* Structures in a Cubic Space Group **(#195-#230)** <https://www.atomic-scale-physics.de/lattice/spcgrp/cubic.html>