

Python et chimie

STœCHIOMÉTRIE MOLE-À-MOLE AVEC PYTHON

Rappel : Équation chimique équilibrée

Une équation chimique équilibrée montre les relations quantitatives entre chacune des espèces chimiques concernées dans une réaction chimique.

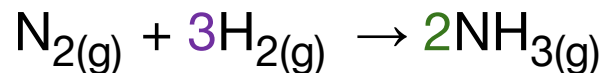
Cela signifie qu'une équation chimique équilibrée peut nous indiquer le **ratio du nombre de moles** de réactifs et les produits participant à une réaction chimique.

Par conséquent, une équation chimique équilibrée fournit des renseignements importants :

1. Le *type* et le *nombre* d'atomes et de molécules qui **interagissent et comment ils s'organisent**
2. Le **nombre relatif de moles** d'atomes et de molécules qui interagissent et se forment

Ratio de mole

Ex. Azote et hydrogène gazeux réagissent pour former de l'ammoniac.



Les coefficients de l'équation nous indiquent que 1 mole de N_2 réagit avec 3 moles de H_2 , formant 2 moles de NH_3 . Cette relation quantitative peut être écrite sous forme de ratio de mole :



Nous pouvons aussi écrire la relation entre deux espèces chimiques, plutôt que la réaction chimique entière. La relation entre N_2 et H_2 a un rapport de mole de :



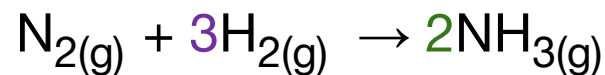
Ratio de mole



À l'aide de ce ratio de mole, on peut calculer la quantité de moles de H₂ nécessaires pour réagir pleinement avec une certaine quantité de N₂ ou créer une certaine quantité de NH₃.

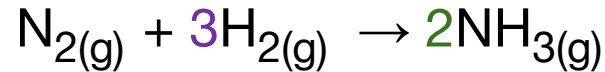
On peut utiliser des **ratios de moles** pour **convertir entre les quantités de deux substances dans une réaction chimique.**

Discussion de groupe : Si nous avons 2 mol de N₂, combien de mol de H₂ sont nécessaires pour réagir pleinement si nous utilisons l'équation suivante :

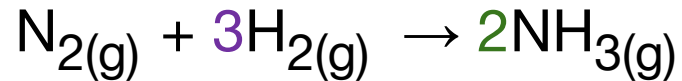


Ratio de mole

Discussion de groupe : Si nous avons 2 mol de N_2 , combien de mol de H_2 sont nécessaires pour réagir pleinement si nous utilisons l'équation suivante :



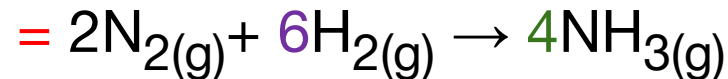
Réponse : _____



x2

x2

x2



Par conséquent, 6 mol de H_2 est nécessaire pour réagir pleinement avec 2 mol de N_2 .

Penser-jumeler-partager

Que nous apprend l'équation chimique équilibrée sur les moles et comment peut-elle être utilisée par un chimiste?

Penser-jumeler-partager

Que nous apprend l'équation chimique équilibrée sur les moles et comment peut-elle être utilisée par un chimiste?

Une équation chimique équilibrée peut nous indiquer le **ratio du nombre de moles** de réactifs et les produits participant à une réaction chimique.

On peut utiliser des **ratios de moles** pour **convertir entre les quantités de deux substances dans une réaction chimique.**

Stœchiométrie mole-à-mole

Les techniques suivantes peuvent être utilisées pour prédire le nombre de moles qui réagiront ou se formeront dans une équation :

- 1.Équilibrer l'équation (toujours).
- 2.Identifier ce qui est donné et nécessaire.
- 3.Utiliser l'équation ratio de moles

$$\# \text{ mol given} \times \left(\frac{\text{balanced mol of required}}{\text{balanced mol of given}} \right) = \# \text{ mol required}$$

Problème de pratique : En utilisant l'équation suivante,

$\text{N}_{2(g)} + 3\text{H}_{2(g)} \rightarrow 2\text{NH}_{3(g)}$, déterminez combien de moles de H_2 seraient nécessaires pour réagir pleinement avec 3 moles de N_2 .

Stœchiométrie mole-à-mole

Problème de pratique : En utilisant l'équation suivante,

$\text{N}_{2(g)} + 3\text{H}_{2(g)} \rightarrow 2\text{NH}_{3(g)}$, déterminez combien de moles de H_2 seraient nécessaires pour réagir pleinement avec 3 moles de N_2 .

Réponse :

1. Équation équilibrée : $\text{N}_{2(g)} + 3\text{H}_{2(g)} \rightarrow 2\text{NH}_{3(g)}$
2. Donné : 3 mol de N_2 Nécessaire : mol de H_2
3. Résoudre avec l'équation de ratio de mole

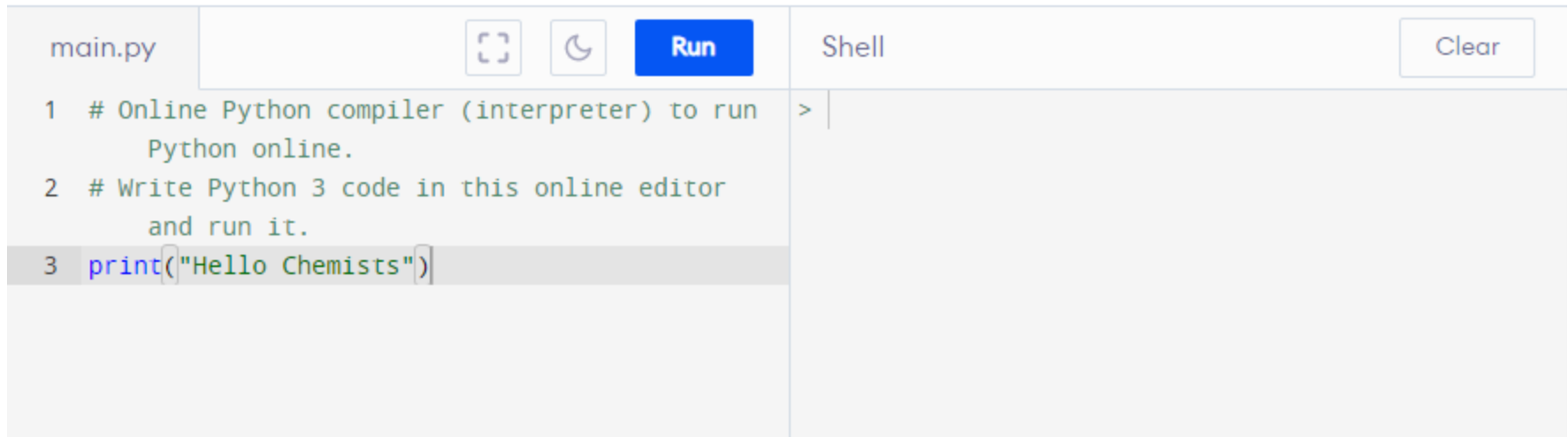
$$3 \text{ mol N}_2 \times \left(\frac{3 \text{ mol H}_2}{1 \text{ mol N}_2} \right) = 9 \text{ mol H}_2$$

Déclaration : Par conséquent, 9 moles de H_2 sont nécessaires pour réagir pleinement avec 3 moles de N_2 .

Intro à Python

1. Commencez par accéder à votre éditeur, compilateur ou interprète de code.
 - a. Programiz : Compilateur Python en ligne
<https://www.programiz.com/python-programming/online-compiler/>
 - b. Replit : <https://replit.com/>
 - c. Python en ligne : <https://www.online-python.com/>

Le code Python peut être écrit dans une console comme celle ci-dessous.



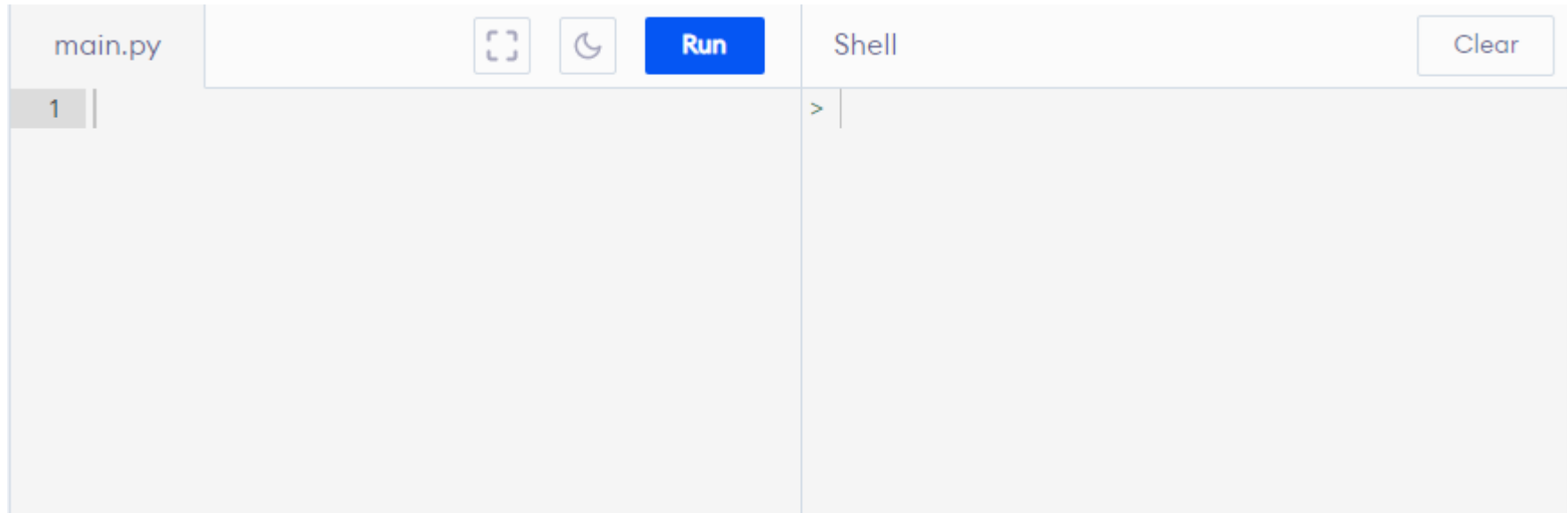
The screenshot shows an online Python editor interface. On the left, there is a code editor window titled 'main.py' with a 'Run' button. The code in the editor is:

```
1 # Online Python compiler (interpreter) to run  
  Python online.  
2 # Write Python 3 code in this online editor  
  and run it.  
3 print("Hello Chemists")
```

On the right, there is a 'Shell' window with a 'Clear' button. The shell prompt is '> |'.

Intro à Python

2. Effacez tout dans la console pour qu'elle soit vide.
3. Le programme de code est exécuté de haut en bas.



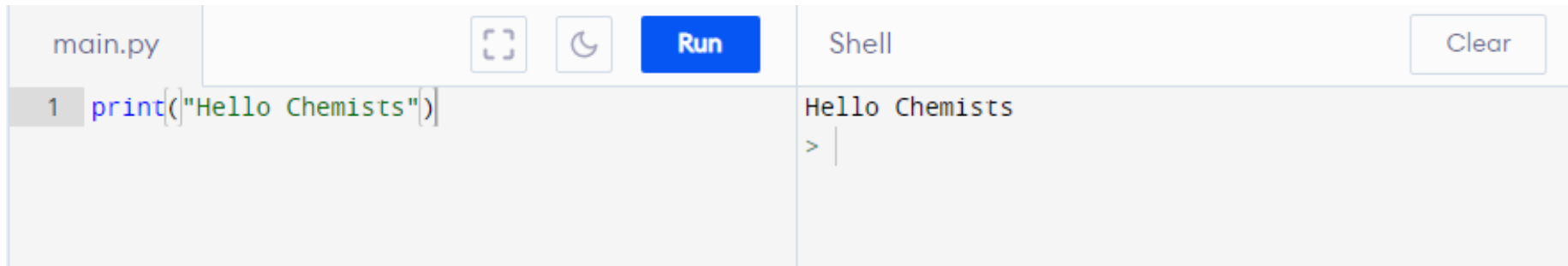
Intro à Python : Vocabulaire

imprimer : `print()` est l'une des fonctions les plus basiques de Python, car elle nous permet d'afficher le résultat du code que nous avons écrit. La fonction `print()` imprime littéralement (affiche) le résultat à l'écran.

Habituellement, quelque chose est nécessaire à l'intérieur des parenthèses. S'il y a du texte de données, des guillemets simples ou doubles entourent le texte de données, également appelé chaîne. Une chaîne combine plusieurs morceaux de texte.

Essayons.

1. Entrez `« Bonjour les chimistes »` dans `print(« Bonjour les chimistes »)`
2. Cliquer sur « Exécution ».



The screenshot shows a Python IDE interface. On the left, a code editor window titled 'main.py' contains the following code on line 1: `print("Hello Chemists")`. Above the code editor are icons for file operations and a blue 'Run' button. On the right, a 'Shell' window displays the output 'Hello Chemists' followed by a prompt '> |'. A 'Clear' button is located in the top right corner of the shell window.

Intro à Python : Vocabulaire

Variables : Une variable est utilisée pour stocker temporairement des données (texte, nombres, etc.) dans la mémoire d'un ordinateur. Exemple, le prix d'un produit ou le nom d'une personne.

La création d'une variable s'appelle « déclaration de la variable ». Vous nommez votre variable et lui attribuez une valeur, mais certaines règles existent :

1. Vous **pouvez** utiliser des **lettres**, des **chiffres** et des **traits de soulignement** (`_`) dans un nom de variable.
2. Les noms de variables sont **sensibles à la casse** : exemple, Exemple, et eXemple sont toutes des variables différentes en ce qui concerne Python.
3. Vous **ne pouvez pas** démarrer un nom de variable avec un **chiffre**.
4. Vous **ne pouvez pas utiliser un mot réservé** (c'est-à-dire, `def`, `if`, `else`, `False`, `True`, `None`...) comme **nom de variable**. Si vous n'êtes pas sûr de savoir quels mots sont réservés (c'est-à-dire que Python les utilise pour certaines fonctions), tapez Aide (« mots clés ») dans votre interpréteur de lignes de commande.
5. Vous **attribuez** une valeur à une variable **à l'aide du signe égal** (`=`). D'abord vient le nom de la variable, puis le signe égal, puis la valeur :

```
    sujet_préfér  = « la chimie! »  
     ge = 20
```

Intro à Python : Vocabulaire

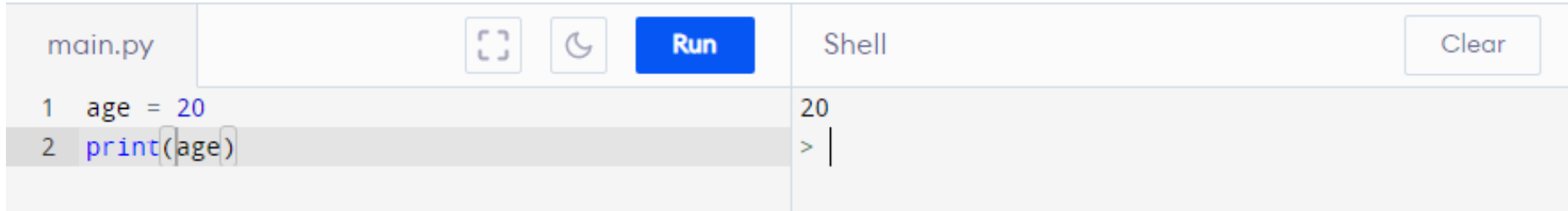
Exemples de variables :

âge = 20

Essayons d'afficher (print) la variable d'âge :

1. Entrez : âge = 20
2. Entrez : print(âge)
3. Cliquer sur « Exécution ».

Notez que nous n'ajoutons pas dans les guillemets « » parce que nous voulons imprimer la valeur de la variable d'âge. Si nous incluons les guillemets, print(« âge »), nous verrions **l'âge** du texte affiché.



```
main.py [ ] [ ] Run Shell [Clear]
1 age = 20
2 print(age)
20
> |
```

Intro à Python : Vocabulaire

Exemples de variables :

```
sujet_préfér  = « la chimie! »
```

Vous pouvez imprimer à la fois une chaîne (plusieurs morceaux de texte) et une variable en incluant la chaîne dans les guillemets qui suivent par une virgule et le nom de la variable

Essayons d'afficher (print) la variable `sujet_préfér ` :

1. Entrez : `sujet_préfér  = « la chimie! »`
2. Entrez : `print(« Mon sujet préféré est »,sujet_préfér )`
3. Cliquer sur « Exécution ».

```
main.py ☐ ☐ Run Shell Clear  
1 favourite_subject = "chemistry!"  
2 print("My favourite subject is", favourite_subject)  
My favourite subject is chemistry!  
> |
```

Intro à Python : Vocabulaire

entrée : `input()` est une fonction qui nous permet d'obtenir l'entrée de l'utilisateur en affichant une incitation. Par exemple, le nom d'une personne ou son numéro favori.

Essayons.

1. Entrez : `entrée_nom = input(« Bonjour, quel est votre nom? »)`
2. Entrez dans `print(« Bonjour », entrée_nom)`
3. Cliquer sur « Exécution ».

```
main.py [ ] [ ] Run Shell Clear
1 name_input = input("Hello, what is your name?") Hello, what is your name?|
2 print("Hello",name_input)
```

1. L'utilisateur tape ensuite son nom et appuie sur la touche Entrée.

```
main.py [ ] [ ] Run Shell Clear
1 name_input = input("Hello, what is your name?") Hello, what is your name? Emily
2 print("Hello",name_input) Hello Emily|
```


Intro à Python : Vocabulaire

Entier relatif : `int()` est un nombre entier qui est positif ou négatif (exemple : 10)

Flotteur : un nombre `float()` est un nombre positif ou négatif qui contient un nombre décimal (exemple : 10,1)

Chaîne : une `str()` une combinaison de plusieurs morceaux de texte




Essayons un calcul avec des entiers relatifs.

1. Tapez ce qui suit :
2. Cliquer sur « Exécution ».
3. Terminer les entrées en répondant aux questions.

Discussion de classe :

Que s'est-il passé?

Pouvons-nous résoudre ceci?

```
main.py   
1 Pet_Dog_input = int(input("How many students
    have a pet dog?"))
2 Pet_Cat_input = input("How many students have a
    pet cat")
3 CatDogSum = Pet_Dog_input + Pet_Cat_input
4 print(CatDogSum, "students have a cat or dog as
    a pet")
```

Intro à Python

Discussion de classe :

Que s'est-il passé?

Un bogue! Cela signifie que quelque chose ne va pas dans notre code. Le programme va essayer de vous aider en affichant une erreur.

Oui, nous pouvons! Comment?

Pouvons-nous résoudre ceci?

main.py



```
1 Pet_Dog_input = int(input("How many students
    have a pet dog?"))
2 Pet_Cat_input = input("How many students have a
    pet cat")
3 CatDogSum = Pet_Dog_input + Pet_Cat_input
4 print(CatDogSum, "students have a cat or dog as
    a pet")
```

Nous avons oublié int()



Intro à Python : Bogue

Discussion de classe :

Que s'est-il passé?

Un bogue! Cela signifie que quelque chose ne va pas dans notre code. Le programme va essayer de vous aider en affichant une erreur.

```
main.py [Run] Shell [Clear]
1 Pet_Dog_input = int(input("How many students
  have a pet dog?"))
2 Pet_Cat_input = input("How many students have a
  pet cat")
3 CatDogSum = Pet_Dog_input + Pet_Cat_input
4 print(CatDogSum, "students have a cat or dog as
  a pet")
How many students have a pet dog?5
How many students have a pet cat4
Traceback (most recent call last):
  File "<string>", line 3, in <module>
TypeError: unsupported operand type(s) for +: 'int'
  and 'str'
> |
```

Cette erreur nous dit que nous essayons d'ajouter un entier relatif et une chaîne, qui représente du texte. Nous ne pouvons pas calculer les nombres avec une chaîne.

Intro à Python : Bogue

Essayons de **corriger le bogue** et de coder un calcul avec des entiers relatifs.

1. Tapez ce qui suit :
2. Cliquer sur « Exécution ».
3. Terminer les entrées en répondant aux questions et en pressant sur la touche entrée.

main.py



Run

Shell

Clear

```
1 Pet_Dog_input = int(input("How many students
    have a pet dog?"))
2 Pet_Cat_input = int(input("How many students
    have a pet cat"))
3 CatDogSum = Pet_Dog_input + Pet_Cat_input
4 print(CatDogSum, "students have a cat or dog as
    a pet")
```

```
How many students have a pet dog?5
How many students have a pet cat4
9 students have a cat or dog as a pet
> |
```

Intro à Python : Calculateur de code

Entier relatif : `int()` est un nombre entier qui est positif ou négatif (exemple : 10)

Flotteur : un nombre `float()` est un nombre positif ou négatif qui contient un nombre décimal (exemple : 10,1)



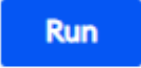
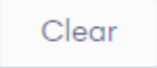
Chaîne : une `str()` une combinaison de plusieurs morceaux de texte

Tâche 1 : Essayez de créer du code qui effectue un calcul avec des entiers relatifs et des flottants.

Intro à Python : Calculateur de code

Tâche 1 : Essayez de créer du code qui effectue un calcul avec des entiers relatifs et des flottants.

Voir Exemple de code :

main.py	  	Shell 
<pre>1 Candy_Have = float(10.80) 2 Group_Input = int(input("How many people are in a group? Choose a number between 1 and 5.")) 3 CandyDivide = Candy_Have/Group_Input 4 print("Each person in the group will have", CandyDivide, "pieces of candy")</pre>	<pre>How many people are in a group? Choose a number between 1 and 5.3 Each person in the group will have 3.6 pieces of candy > </pre>	

Intro à Python : Code de ratios de moles

Tâche 2 : Essayez de créer du code qui effectue un **calcul de ratios de moles** avec des entiers relatifs et des flottants.

Intro à Python : Code de ratio de moles

Tâche 2 : Essayez de créer du code qui effectue un **calcul de ratios de moles** avec des entiers relatifs et des flottants.

Voir Exemple de code :

main.py 📄 🌙 Run	Shell Clear
<pre>1 Mol_Given_Input = float(input("How many moles do you have or will create?")) 2 Required_Balanced_Mol_Input = int(input("What is the coefficient, representing the mole ratio, of the required atom/molecule in the balanced equation? ")) 3 Given_Balanced_Mol_Input = int(input("What is the coefficient, representing the mole ratio, of the given atom/molecule in the balanced equation? ")) 4 Mol_Required_Input = Mol_Given_Input *(Required_Balanced_Mol_Input /Given_Balanced_Mol_Input) 5 print("Therefore", Mol_Required_Input, "mol will be required or produced.")</pre>	<pre>How many moles do you have or will create?2.5 What is the coefficient, representing the mole ratio, of the required atom/molecule in the balanced equation? 3 What is the coefficient, representing the mole ratio, of the given atom/molecule in the balanced equation? 1 Therefore 7.5 mol will be required or produced. > </pre>