

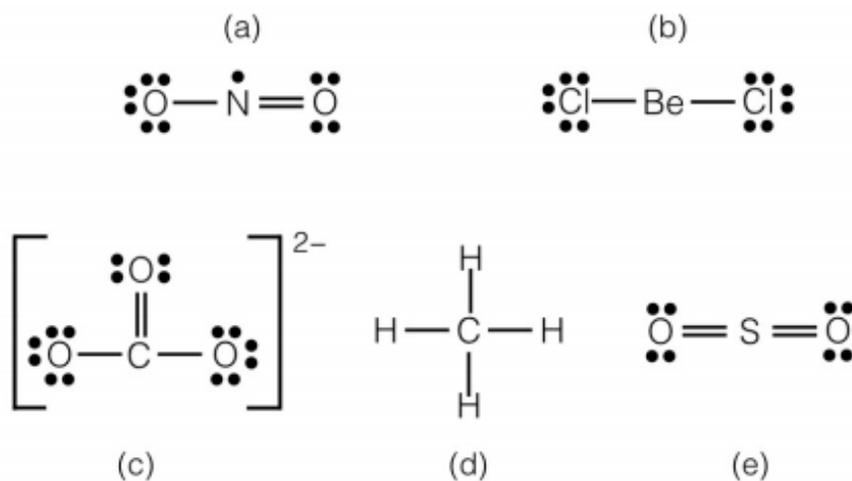


# Python Coding Lesson

Lesson Title:	Lewis Structures of Covalent Species
Grade Level:	High School
Subject:	Chemistry
Contributed By:	C. Le and E. Ermer

## Engagement:

- SWBAT evaluate the functionality of a python code to help with drawing Lewis Dot structures of covalent compounds and polyatomic ions.
- SWBAT interpret and enhance the python code to address any limitations it has for accurately predicting the number of bonds in any covalent compound or polyatomic ion.
- We've talked about the process for determining correct Lewis structures of covalent compounds and polyatomic ions. Can you come up with an equation that could summarize or simplify the rules?
- In your notebooks, draw the correct Lewis structure for carbon monoxide. How confident are you in your answer?
- Some Lewis structures are challenging/infuriating. If you had a chemistry wish that granted you one piece of information (besides the Lewis structure itself, smart alec) what do you wish you could have besides the formula that would help you draw or choose the correct Lewis structure every time?



Source: <https://www.clutchprep.com/chemistry/practice-problems/5660/which-one-of-the-following-lewis-structures-is-definitely-incorrect-160>



# Python Coding Lesson

## Exploration:

- Today we will explore a program that is designed to help draw correct Lewis structures for every covalent compound or polyatomic ion. When a user inputs the chemical formula into the program, the program responds with how many bonds that formula contains. It sounds simple, but it is actually fairly complicated.
- First, students will evaluate the effectiveness of the code by inputting formulas of compounds we did in class previous days into the program and seeing how accurate the program is at predicting the number of bonds. Make a list of which compounds break the program or where the program does not accurately predict the correct number of bonds.
- Student pairs will run the attached program.
- As a class, we will compile a list of these compounds/species and ask the class to: 1) verify that the data collected is corroborated and not just inputted incorrectly by a student or a glitch in a certain computer. 2) see if they can find a pattern for the types of species that “break” the program.
- Then, students will be given access to the “back end” of the code and, in pairs, will be asked to decipher what each piece of the code is doing and why or why not it makes sense to do that to return the correct number of bonds. There will be some guiding questions to help some pairs get started, such as: “How do you think the program would change if “int” was left out of line 8?” or “How would the program respond if Te or Xe is input into the program?”

## Explanation:

- After the majority of students have had time to discuss this with their partner, students will pair with a different student and compare their analysis and predictions and discuss any disagreements.
- Students from different groups will share with the class their analysis of what each section of the code does and how it will respond to the different scenarios. Students will ask clarifying questions or suggest alternate explanations if they don't completely understand or agree with the analysis. After presenting, the code will be changed and we can see how accurate the predictions of how the program would respond were and discuss any surprises.



## Python Coding Lesson

### Elaboration:

- Students will then be instructed to work as pairs to change the code so that the program will accurately predict the number of bonds in the compounds that broke the code before as well as the compounds that the older version of the code worked for.
- Students will then have an opportunity to share the changes they made to the code and why it is more effective. We can test these programs to see how they respond to new species that haven't been tested yet.

### Evaluation:

- You will not be able to use this computer program to help you on a test. How can you "write" this program into your brain so that you will be able to draw correct Lewis structures for covalent compounds or polyatomic ions without the aid of this program? What steps of writing a correct Lewis structure did this program help you to recognize or understand better?

### Code:

```
print ("Please input your molecule with a space between each element and number; For example, input 'N 1 O 2 - 1' for NO2-")
```

```
molecule = input()
```

```
x = molecule.split()
```

```
for item in x:
```

```
    index = x.index(item)
```

```
    if item.isnumeric() == True:
```

```
        x[index] = int(item)
```

```
total = 0
```

```
for element in x:
```

```
    index = x.index(element)
```

```
    if type(element) == str:
```

```
        if element == 'H':
```



## Python Coding Lesson

```
total += 1 * x[index + 1]
elif element == 'C':
    total += 4 * x[index + 1]
elif element in ['N', 'P']:
    total += 3 * x[index + 1]
elif element in ['O', 'S', 'Se']:
    total += 2 * x[index + 1]
elif element in ['F', 'Cl', 'Br', 'I']:
    total += 1 * x[index + 1]
elif element == '-':
    total += -1 * x[index + 1]
elif element == '+':
    total += 1 * x[index + 1]
bonds = total/2
print ("Number of bonds = " + str(bonds))
```