


Student Worksheet: Build-your-own Valence finder

What is valence? – some atoms are "happy" alone and don't care to play or bond with others. Some atoms are "unhappy" alone and really want to play (bond) with others. What determines whether an atom is happy or unhappy and if it wants to form bonds?..... The number of electrons it has in its outer ring/shell in relationship to number possible in that ring/shell.

For the purposes of determining valence, it is useful to think of the electrons of a particular element as being arranged in layer or shells or rings. The first shell, located closest to the nucleus, can hold 2 electrons. The next two shells, can hold 8 each. The electrons an element has, fill up the closest shell or shells first and any that are left over, fill into the next shell out.

Every atom of any one element usually has the same number of electrons as any other atom of that element. And most elements usually have the same number of electrons as protons. The numbers of protons (and electrons) an element has is known as its atomic number and this number is different for each and every element. The atomic number for fluorine (F on the periodic table) is 7, which means it usually has 7 protons and 7 electrons.

Whenever you see a , partners should write out answers **together**. If you don't know the answer, ask another group for help before moving on.

Now let's figure out it's valence.



1. How many electrons are on its first shell? _____
2. How many electrons are on its second shell? _____
3. Are there any electrons on its third shell? _____
4. If having a full outer shell is what will make an element "happy" what does fluorine want to do to become happy (hint: it can do two things, but one is more likely)?

The number of electrons an atom/element wants to give or get to become happy is its valence number.



5. What is the valence number for fluorine? _____

Valence also has a sign (or charge) which is what charge the atom will be if it gives or gets electrons.



6. If an atoms gives up an electron or electrons, will it be negatively charged or positively charged – will it have more electrons than protons or less electrons than protons? _____
7. What is the valence charge for fluorine? _____

The combination of an element's valence number and valence charge IS its valence!

For this exercise we will be creating a computer program to determine valence automatically, for any element in the first three rows (periods) of the periodic table. As we work through a valence problem together, we will also write code in the program to do the calculations for us. After we finish writing all the code and save our program, all we need do is run it and enter the any value for an element's atomic number and the program will do the rest!

The programming language we will be using is Python. Don't be scared, it won't bite!


What is Python?


Python is a programming language. It is used to make programs. The programs make the computer do stuff. Computers are stupid but follow directions very well and ...fast. Everything that one does with/on a computer or the internet happens within some kind of program that someone created.


Python commands to create an interactive program:

- =** sets or resets the value for a variable. For example, "x = 5" sets a value for the variable x as 5.
- ==** tests to see if true
- print** this is needed in **a programming window** so that the the computer will display whatever comes after. For example, "print eval('35+24')" tells Python to display the answer to 35 + 24.
- input** allows the program to ask a question of the user and record the answer. For example, "c=input('How many pets do you have?')" will ask the user for his or her number of pets and store the number entered as the variable c.
- if** conditional statement to test that a condition has been met
- elif** next in a series of alternative conditional tests. Used after "if".
- else** last in a series of alternative conditional tests. Used after "elif" or "if".

Procedure:

 The first thing we need to do is open up the programming "shell" window. Your teacher will tell you where to find the icon for the application called IDLE. Click to open IDLE also known as the "shell" window. The "shell" window is where we eventually see our program run.

Whenever you see a, , one partner should be working on the computer.


 The next thing we want to do is open open up a new programming window. Go to the *File* menu and drag down to *New window*. The "programming" window is where we will be writing our code. NOTE: the way to easily tell the difference between the shell and prograqm window is that the shell will have the symbol >>> that precedes the blinking cursor.


Since we want our program to be a calculator that can use ANY atomic number let's make it ask for it by typing the following in the new program window:

```
n = input("What is the element's atomic number?")
```

What this does:

It asks the user for input (usually a number), which it will store in a named variable *n*

 The value will be stored in that variable (holder) and could be used by other equations or it can be displayed by.....this code: print n Let's code it.

 Now let's save our program as *ValenceFinder.py* and the go to the *Run* menu and choose *run module*. NOTE: The program cannot be run until it is saved first. When it asks you "What is the element's atomic number?", type it in and push return.


Now lets make the program be able to deal with, and find the valence for, the elements on the **first** row of the periodic table.


8. What is the atomic number for Hydrogen? _____




9. How many electrons does Hydrogen have? _____
10. What is the atomic number for Helium? _____
11. How many electrons does Helium have? _____
12. Why do you think these are the only two elements on the first row (hint: think about shells)? _____


For this section of our program we want to deal with only elements with these two atomic numbers. How can we specify only these two possible numbers: 1 or 2?

 Let's code it! In the programming window, erase *print n* and type: *if n<3:*

 To have the program deal with hydrogen, whose atomic number is 1, on the next line, indent with one push of the tab key and type *If n==1:* and push return to get to next line (note: if you put a colon after the if statement it should automatically indent). On this line: *print "I'm not happy. I have one electron in a two-possible shell. My valence is positive or negative 1 electron. I'd be happy to give or get one"*

 To have the program deal with helium, whose atomic number is 2, on the next line, indent with one push of the tab key and type: *else:* and on the next line: *print "I am very happy. I have two electrons in a two-possible shell. I don't want to give or get any electrons."*

Now let's make the program be able to deal with and find the valence for the elements on the **second** row of the periodic table. For this section of our program we want to deal with only elements with two atomic numbers between 2 and 10. How can we specify only these two possible numbers: 1 or 2?


 Let's code it! In the programming window, on the next line, backspace (twice) to outdent in order to line up with the first *if* statement, so that we can specify an alternative to *if n<3* and type: *elif 2<n<11:*

What this does:

This specifies a range of atomic numbers for the second row which will be dealt with by code following the statement.



13. If we know how many total electrons our elements has, how can we get rid of the electrons on the first shell in order to determine how many are left over to fill in the second shell?

 Let's code it! On the next line type: *v=n-2*

There are eight possible electrons on this shell. Only one is "happy", but there are three possible types of "unhappy".




14. Can you think of what the three possible types of unhappy elements there could be? -


Let's think about the one "happy" element.....

15. If we want the program to deal with the one "happy" valence element, we need help the program to recognize it. How many electrons would a "happy" element in this row have in its outer shell?



16. How can we write a conditional if statement to verify that? _____

 Code it on the next line.


 And put this on the next line: *print "I am very happy. I have 8 electrons in an eight-possible outer shell."*


Let's think about one of the three "unhappy" types elements that will want to give away electrons (which will give them a positive valence).....



17. If we want the program to recognize with the “unhappy” **positive** valence elements, we need help the program to recognize it. How many electrons would a positive valence “unhappy” element in this row have? _____

18. How can we write a conditional if statement to verify that? _____

 Code it on the next line. Use *elif* and a *:* as this is the second conditional test in a series. AND backspace once to outdent to line up with *if v==8*:

 And put this on the next line: *print "I'm not happy. I have", v, " electrons in a eight-possible outer shell. My valence is plus", v, "electrons. I want to give", v, " electrons away and when I do I will be more positively charged."*


NOTE: The *,* *v*, inserted a various point in between quoted text will tell the program to print the value held in the *v* variable, at those points in the printed text. Make sure the commas and quotation marks are exact.


Let's think about another of the three "unhappy" types elements that will want to get electrons (which will give them a negative valence).....



19. If we want the program to recognize with the “unhappy” **negative** valence elements, we need help the program to recognize it. How many electrons would a negative valence “unhappy” element in this row have? _____

20. How can we write a conditional if statement to verify that? _____

 Code it on the next line. Use *elif* and a *:* as this is another conditional test in a series. AND backspace once to outdent to line up with *if v==8*:


 And put this on the next line: *print "I'm not happy. I have", v, " electrons in a eight-possible outer shell. My valence is negative ", 8-v, ". I want to get ", 8-v, " electrons and when I do I will be more negatively charged"*


Let's think about of the three "unhappy" types elements that would be happy to give or get electrons (which would give it positive OR negative valence)...



21. If we want the program to recognize with the “unhappy” valence elements, we need help the program to recognize it. How many electrons would a positive OR negative valence “unhappy” element in this row have? _____

22. How can we write a conditional if statement to verify that? _____

 Code it on the next line. Use *elif* and a *:* as this is another conditional test in a series. AND backspace once to outdent to line up with *if v==8*:

 And put this on the next line: *print "I'm not happy. I have", v, " electrons in a eight-possible outer shell. My valence is negative ", 8-v, ". I want to get ", 8-v, " electrons and when I do I will be more negatively charged."*

We're done for the second row possibilities. You can save and run it.

23. What do you think would be a quick way of creating code for the third row?




Outdent all the way, before you paste.

BTW – recycling code and is often done. It saves a lot of typing time!



24. What do we need to change here *elif 2<n<11*: ?

 Code it. Save and run it, using a (atomic) number for any element in the third row?

25. What's wrong? _____

We need to fix something in the program's code. This is called de-bugging.



26. What do we need to fix? _____

Code it. Save and run it, using a (atomic) number for any element in the third row?

Now the last thing we need to do is write a line that will deal with numbers beyond the third row. We don't really need to test or set up an *if* condition. All we need to do to catch all the atomic numbers we have not already recognized is outdent all the way and put *else*:

And put this on the next line *print "I'm not on the first three rows so you don't need to know now."*

Save and run it!

In our program we had to tell the computer how to recognize and indentify valence for any of the elements in first three periods of the table.



27. How did we initially set up the section that dealt with ONLY items on the first row? _____

28. The second? _____

29. The third? _____

30. How did we enable the program to identify and deal with any of several elements in either the second or third row, which would have a negative valence?

31. A positive valence?

32. A positive or negative valence?

33. Why did we not have to go through such elaborate programming for the first row?

If you want you can email the file to yourself, so you can use it to check any valence problems you practice at home. To use your program, you'll need the Python shell (IDLE comes in the bundle) which you can download for free here: <http://python.org/download/>



Build-your-own Valence Finder lesson by Mark Wenning is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/).