|  |  |
| --- | --- |
| Class | Wellington Year 11-12 Chemists |
| Lesson 3 | Arrays and plotting graphs In matplotlib.  Manually creating database or boiling points and plotting it  London forces/van der waals |
| Building on | Carbon atoms in chains and rings  Fractional distillation  Variation in physical properties  Cracking  Skeletal formulae and using computers.  smiles strings |
| Last lesson | SMILES strings can be used to represent molecules in computers.  How to use and manipulate strings in Python  How to apply this in chemistry |
| Learning objectives | Students should know that variables can be stored in a list or array  They should be able to use a for loop to iterate over a list or array  Understand that an array should have only one data type in it, while a list can have any number of types  Lists can be manipulated using a for loop, numpy arrays can be manipulated directly. |
| Requirements: | Jupyter notebook with RDKit, pandas, matplotlib numpy env to work in |
| 55 min lesson |  |
| Starter  5-10 mins | What is the smiles string for Propane?  > prop = ‘CCC’  How would you change this string using Python to the one for hexane?  Concatenate ‘CCC’ or do prop = prop+prop or prop\*2  How would you index the third character in a string variable called ‘name’?  >name[2]  How would you get the 2nd to 5th characters of the same string?  >name[1:5]  What does this code show?  >print(‘I’m feeling fine’)  SyntaxError  What does this code return?  >3\*’Hello!’  Hello!Hello!Hello! |
| Arrays  10 mins | We can store the information we generate in a list.  Arrays and lists are defined in Python using square brackets.  An empty list is defined by []  A list has an order.  For example,in the notebook I have defined a list of names. Please change these names to names you like, add other names if you like too, then run this block.  You can index by using the name of the array e.g. names[0] gives you the first value, names[-1] gives you the last value, names[1] gives the second value, e.t.c..  What happens if you try to index something that isn’t there, i.e. names[9]? – (try it!) IndexError  You can append to an array by using a method on the array itself.  names.append(‘Fred’) adds the name ‘Fred’ to the list.  Work down to the set of questions and do them. |
| For Loops  10 mins | If you want to say hello to everyone on the list, it would be annoying to write print(‘Hello ‘ + ‘Alice’ + ‘!’) out for each of the people in the list.  So we use a for loop as a way of repeating the same code again and again.  The syntax is simple: first you put the word for, then you put the variable name that will substitute for each of the members of the array one after the other, then you put in, and last you put the name of the array you will be iterating over.  You can use any array, though be careful if you have different data types, the same code might not work for everything.  You can substitute range(n) for any list of numbers 0 – n.  Run through code blocks on for loops, then work though the TODO in block 19. |
| Using RDKit  20 mins | We can use range to define as many alkanes as we want.  See blocks 20-21.  Complete the TODO in block 22  If you finish that, try to get the skeletal formula for each one to show. You may need to google how to do this.  Try to do the same for other homologous serieses, such as alkenes, alcohols |
| Class activity | Drawing out all the molecules.  There is more than one way to do this, hwoever I’ve chosen the one that I think will require the least extra learning.  If you ask the kids to try this, they might suggest putting rdc.Draw.MolToImage(mol) in the for loop.  This should work, however due to the vagaries of how jupyter notebooks work, it doesn’t.  You can do rdc.Draw.MolToMPL(mol) which uses matplotlib.pyplot library to generate the image, however the results aren’t fabulous.  You could use rdc.Draw.MolToFile(mol, filename) which is quite fiddly as you need to specify the filename, and make it different for each one. You can do this using the string concatenation or formatting learned earlier.  These image files will be saved in the same directory as the ipynb notebook is saved in, so you can find them later if you need.  I would use rdc.Draw.MolsToGridImage([mol1, mol2, …]) which takes a list of mol objects that you can make in the same loop you used previously, and then prints all of them out at once.  It does take an optional argument molsPerRow, which you can change from the default 3 to another number if you like.  Draw from class their own ideas, and encourage them to try them. Learnng is happening when red error messges appear ☺ |
|  |  |
|  |  |
| Summary  2 mins | Learned about lists and for loops in python, how to repeat code without having to write it out lots of times.  Learn how to apply this knowledge to chemistry, in making names of compoutns, and printing them out. |
|  |  |