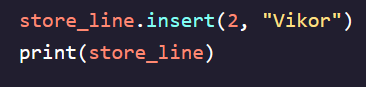
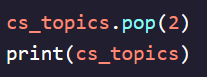
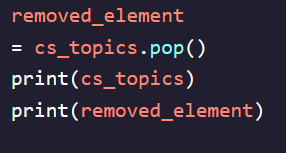
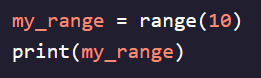
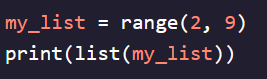
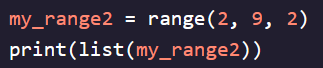
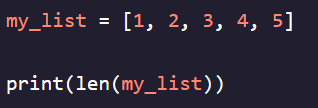
**Python List Methods:**

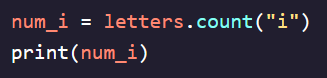
***- .insert()*** – Used to insert an element into a specific index of a list  
- Takes in two inputs:  
1. The index you want to insert into  
2. The element you want inserted at the specified index  
- When we insert a new data point all elements from the specified index to the last one are shifted one index to the right (index 3 🡪 index 4)  


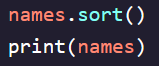
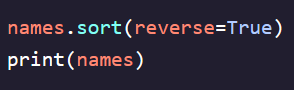
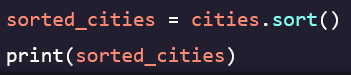
***- .pop()*** *–* Used to remove an element from a specific index of a list or from the end  
- Takes an (optional) input:  
1. The index for the element you want to remove  
2. If none provided will remove the last element in list  
- Can assign *.pop()* function to a variable and view what elements were removed  


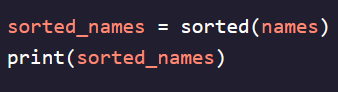
***- range()*** – Used to create a sequence of integers  
- Takes a single input and generates numbers **starting at 0** and ending at number **before** desired input (range(5) 🡪 0, 1, 2, 3, 4  
- Does not create a typical list but instead makes a *range object  
-* Range objects do not need to be converted to lists in order to determine their length- If you call *range()* with two inputs then it will create a list starting at the first number up until just **before** the second number  
  
- If you use three inputs you can create a list that skips numbers according to the value of the third number  


***- list()*** – Used to create a list of numbers from input   


***- len()*** – Used to get the number of elements in a list   


**- *.count()*** – Used to count the number of occurrences of a specific element in a list  
- Returns a value so we can assign it to a variable and use it  
- Can be used to count element appearances (sublist) even in a 2D list  
 

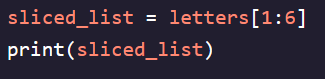
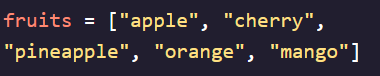
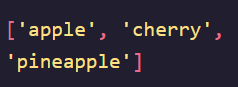
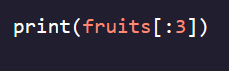
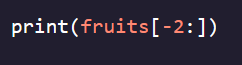
***- .sort()*** – Used to sort a list and get sorted variables  
- By default, sorts lists in **alphabetical, ascending** order (A, B,C…)  
- Can also sort in **reverse, descending** order (Z, X, Y…)  
- Does not return any value and therefore does not need to be assigned to any variable to use  
- Changes original list  
   

***- sorted()*** – Different from *.sort()* because it:  
1. Comes **before** a list not after it  
2. Generates an entirely new list, leaves original list completely unchanged  


***- zip()*** – Allows us to combine lists without needing to recreate as 2D list (but makes them **Tuples)**  
- Creates a *zip object* which stores new list in and call *zip()* function with list names as arguments  
A screenshot of a computer

Description automatically generated   
- *Zip object* cannot be printed as is and requires the use of the *list()* function to display fully  
   
   
- New data set is **NOT** a list but an immutable **Tuple**

**Slicing Lists:**

- Used when we want to extract only a portion of a list  
- *letters[start:end] ­*  
*start* – This is the index of the first element we want to include in our selection *end* – This is the index of **one more** than we want to include in our selection (we want 5th index, we input 6)  
  
  
- If we want to take elements all the way from the first index until a specified one, we can just leave the *start* element blank  
  
- We can also use it in reverse if we want to slice elements from the end of a list all the way to the start  
  
- Can use to slice all but the last element from a list as well  
