|  |  |  |  |
| --- | --- | --- | --- |
| Category | Task | Syntax | Comments |
| **Lists** | Sorting | * sorted (t) * sorted (t , key = ‘function\_name’)   Use “reverse = True” for descending order | The second sorted is used when there are lists within lists and we want to sort based on a specific element |
| Appending/  Deleting | t = []  t.append (‘a’)  item = t.pop() |  |
| Iterating a list | for **index**, **item** in **enumerate** (t): | Index gives the index value and item gives the actual item in the list |
| Copy a list | import copy  c= copy.**deepcopy(a)** | Creates a copy of a into c, Makes sure the the changes made to a do not affect c. |
| **Tuples** | Definition | x = ( ‘John’, 2, 90 ) |  |
| **Dictionaries** | Definition | eng2sp = { ‘one’ : ‘uno’ , ‘two’: ‘dos’} |  |
| “in” operator | Used to check if a word is present as a key of the dictionary  If ‘word’ in dict:  do blah |  |
| Keys / values | * **dict.keys()** -> Returns keys list * **dict.values()** -> Return values list * **dict.items()** -> Return (key, value) pair as a list of tuples |  |
| Iterating a dictionary | * for **key** in **dict.keys**():   value = dict[key]   * for **key, value** in **dict.items**():   print key , value   * for **key, value** in **dict.iteritems**(): | dict.items() -> Returns a list of key,value pairs  **dict.iteritems -> Produces one key value pair in each iteration – Use this** |
| **Strings** | Convert list into string | delimiter\_string.join([item1, item2, item3, ...]) |  |
| Convert string into list | list(orig\_string) |  |
| lstrip | 'AAA Car Insurance'.lstrip('A') |  |
| rstrip | 'AAA Car Insurance'.rstrip('e') |  |
| String split | "A told B, and B told C, I'll race you to the top, of the coconut tree".split(' ') |  |
| Format | *# The {} spaces are filled in successively by the parameters of the format function.*  greeting = 'Hello Citizen {} ID {}'.format(name, ID) | **This is very helpful while writing he formula in dmatrices in Modelling** |
|  |  |  |  |
| **Files** | Open file | fp = open('Intro\_4\_data/CarParts.csv', 'r') |  |
| Read line | first\_line = fp.readline()  line = fp.readline()  **while** line **is** **not** '':  *# the line can have some trailing characters, like 'newline'*  line = line.rstrip()  *# read in a new line*  line = fp.readline() |  |
| **Regular Expressions** | '[0-9]' | Any character between 0 and 9 | **import** **re** |
|  |  | **[...]** means match any character within the square brackets  **[^...]** means match anything *except* the characters within the brackets |  |
|  | *[abcd]* | means match any **one** of 'a', 'b', 'c', or 'd' |  |
|  | *[a-zA-Z]* | means match any lower-case or upper-case character |  |
|  | Shorthand | **\w** is shorthand for [a-zA-Z0-9\_]  **\d** is shorthand for [0-9]  **\s** is shorthand for a space or a tab, often used as delimiter |  |
|  | [/.] | Matches a full stop |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |