

10/4/23

Lists

- It's an ordered pair of elements/items.
- It's a data structure in python & enclosed with square brackets.
- Elements/items can be integers, str, char or other list

↳ nested list

- ↳ list() is a method
- ↳ new list can be created using this method & by passing arguments to it.

a) lists are mutable

- for nested list → we can use two indexing, ie inner list index & outer list index
- 1st index is the position of inner list in the outer list
- 2nd index is the position of element needed within inner list.



- index doesn't exist : Index Error
- -ve index : Counts from backwards
- any integer expression can be index value
- in operation on list gives Boolean value

b) Traversing a list

- Using for loop
- Using range() and len()
- Can also modify the values in the list

c) List Operations

- '+' takes 2 list objects and concatenates them.
- '*' takes 1 list & 1 integer repeats the list the no of times specified by the integer

d) List Slices

- similar to string
- It can also modify only certain items in a list based on index which cannot be done in string.

e) List Methods

- internally return type of most of these methods is: None
- 1. append() → add elements to the end of the list → adds value as it is
 - 2. extend() → 2 list → enhance the existing list
 - 1st is unaltered
 - 1st is added to end of 2nd ∴ 2nd is altered
 - 3. sort() → sorts in ascending order → gives TypeError if there are incompatible items in the list.
for descending order - sort(reverse=True)
 - 4. reverse() → reverse the content in the list
 - 5. count() → counts the number of occurrences of a value in a list.
 - 6. clear() → deletes elements in the list
 - 7. insert() → inserts a value before the specified index in the list
 - 8. index() → gives the index of the value
if the value is not in the list - it throws a value error.

f) Deleting Elements

when index is known → `pop()` → deletes last element in a list
index can be specified as item in that index will be removed

when item is known but, not its index → `remove()` → it will only remove the first occurrence of the specified value

3. `del()` → to del more items in the list

g) Lists and Functions

`max()`, `min()`, `len()`, `sum()`

h) Lists and Strings

`list()` → forms a list of strings

`split()` → 't' as a delimiter, used to split words in a sentence

`join()` → joins list items to form a string
here delimiter needs to be specified, usually white space is used.

i) Parsing lines

parsing → read a file and extract lines containing required pattern from a file.

j) Objects and values

→ two variables may have same value but in different memory location i.e changing one doesn't affect the other

a = 'hi'
b = 'hi'

a is b → equivalent objects
False

→ a is an alias name of b
i.e. two variables are referring to same memory object → identical objects



string literals are interned
lists are not interned

k) Aliasing

→ one variable is declared
→ another variable is referenced to 1st variable using assignment operator
object is aliased

→ both variables are referencing the same memory object
→ An aliased object is an object with more than one reference / name.

→ Strings are immutable ∴ safe

i) list arguments \Rightarrow Should know the difference between operations that modify the list and operations that create a new list.

- Shifting within a function

12/4/21

Dictionaries

\rightarrow It's a collection of unordered set of key-value pairs.
 \rightarrow They are accessed using keys

Tuple can be made as key for dictionary.

\rightarrow Actions are performed on the 'key' and not 'values'

\rightarrow The in operator on list does linear search on 'dict' it works on hashing method

hashing Technique

Hash Table = distributing keys in an one dimensional array

Hash function = predefined function that computes a value for each key.

Hash address = integer value assigned by hash function to each key

Hash Collision \rightarrow If two or more keys are hashed into the same cell on the hash table i.e. they are assigned the same value by hash function then it's called hash collision

Methods of Hash Collision Resolution

a) Open hashing / Separate chaining

\rightarrow keys are stored in linked lists attached to the cells of the hash table.

Operations on hashing

- \rightarrow Searching
- \rightarrow Insertion
- \rightarrow Deletion

b) Closed Hashing (or Open Addressing)

\rightarrow link lists are used

\rightarrow linear probing is used to prevent collision
 \hookrightarrow searching becomes a linear search & not efficient

a) Dict a set counters \Rightarrow we are creating a histogram
 \rightarrow get()

b) looping and dictionaries

c) dictionaries and Files

\rightarrow Outer loop for iterating over every line in a file
 \rightarrow Inner loop for traversing each line in a file.

d) Advanced Text Parsing

- Use string methods to convert all letters to small letters i.e lower
- Eliminate punctuations and use ' ' at its place

line.lower()

line.translate(line.maketrans(<fromstr>, <tostr>, <deletestr>))

e) Debugging

1. Scale down the input
2. Check summaries and types
3. Write Self checks
 - sanity check , avg < largest num & avg > smallest num
 - consistency check , compare the results of two different computations
4. Pretty print the output

Tuples

- They are sequence of items like lists but are immutable
- They are comparable and hashable objects
- They can be made as keys in dictionaries

for tuple with a single item include a comma after the item
→ Elements are extracted with square brackets and indices
→ Tuples can also be sliced

a) Comparing Tuples

1. $= = \rightarrow$ comparison happen lexicographically

↓
1st item → if true → 2nd → if true → 3rd ... till mismatch
if mismatch - false
if no mismatch - true

2. $>, <$ comes before or comes after

DSU → decorate, sort, undecorate

14/4/2021

- Decorate by building a list of one or more sort keys preceding the elements from the sequence
- Sort by using python's built in sort function
- Undecorate by extracting the sorted elements of the sequence

b) Assignment Operator

→ Tuples allows us to assign values to multiple variables at a time .

→ It treats LHS as a set of variables and RHS as a set of expression

→ Giving more values on RHS than available variables on the LHS will raise a Value Error

c) Dictionaries and Tuples

- dictionaries have a method called item().
- item returns a list of tuples
- Each tuple is a key-value pair

d) Multiple assignment with dictionaries

`sort()` ⇒ sorting is done on whichever value is first in the list of tuples

e) Most Common Words

→ To find most commonly used words in a text file.

Algo:

Open a file

Take a loop to iterate through every line

Remove punctuation marks and make all lower case

Take a loop to iterate through every word in a line

If Word not in dictionary :

treat it as key and initialize value to 1

else:

increment the value of the key

Now you will have a dictionary with distinct keys and their frequency in text as values

Now take a list and append each key-value pair into it

Sort in descending order

Print the first 10 elements in list to get most frequent words.

d) Summarye) debugging

→ Compound data structures are prone to shape errors.

1. Reading

2. Running

3. Ruminating → Think what kind of error it is syntax, runtime, semantic

4. Retreating

04/11/21 Regular Expressions

→ Regular expression is a library of python used to handle the task of searching and extracting

→ `search()` is used to search for a particular word and for this the module `re` must be imported

→ We can use special characters with 're' module/library that indicates specific things.

for ex ^ → caret indicates beginning of a line

a) Character matching with regular expression

→ They are meta characters that can be matched with the search string and each has a particular function.

b) Combining searching and extracting

→ Used to extract data in a particular syntax

c) Escape character

→ to include meta characters within the string and escape their function as the meta character, we can use '\' → backslash

UNIX/ LINUX

grep - Generalised regular expression parser