5otrg2gsd

October 25, 2023

CSC1001 TUTORIAL 7 - LISTS

Frederick Khasanto - 122040014 - 26 Oct 2023

---

1 List

- A sequence of elements, can include any objects
- Ordered
- Mutable object
- Operations on lists are similar to string

[1]:

```
# Defining an empty list
lst1 = []
lst2 = list()
# Defining a list with elements
lst3 = [1, 'a', 'hello', print, input]
```

[2]:

```
# Operations on list

my_list1 = [1, 2]
my_list2 = [3, 4]
my_list3 = [3, 4]

# Addition
print(my_list1 + my_list2)

# Multiplication (copies)
print(my_list1 * 3)
print(3 * my_list1)
print()
print(1 in my_list1)
print(4 not in my_list2)
print()

# Comparison
print(my_list1 > my_list2)
print(my_list2 == my_list3)
```

```python
print(my_list2 is my_list3)
print()

# For loop
for e in my_list1:
  print(e)
```

```
[1, 2, 3, 4]
[1, 2, 1, 2, 1, 2]
[1, 2, 1, 2, 1, 2]

True
False

False
True
False

1
2
```

[3]:

```python
# Methods on lists

his_list = ['a', 'b', 56]
print(len(his_list))  # List length

her_list = [345, 789, 10]
print(min(her_list))  # minimum value
print(max(her_list))  # maximum value
print(sum(her_list))  # sum of values
print()

his_list.append(100)  # add value at the end of the list
print(his_list)
his_list.pop()  # remove the last element in the list
print(his_list)
his_list.remove('a')  # remove the first occurence of the element in the list
print(his_list)
print()

his_list.extend(her_list)  # combine two lists
print(his_list)
print()

her_list.insert(1, 200)  # insert element at specified index
print(her_list)
```

```
her_list.sort(reverse=False)  # sort in ascending order (reverse=True for␣
 ↪descending)
print(her_list)
her_list.reverse()  # reverse the list
print(her_list)
print(her_list.count(200))  # count the number of occurences
```

```
3
10
789
1144

['a', 'b', 56, 100]
['a', 'b', 56]
['b', 56]

['b', 56, 345, 789, 10]

[345, 200, 789, 10]
[10, 200, 345, 789]
[789, 345, 200, 10]
1
```

1.1 Slicing

apply to lists and strings

`lst[initial : end : step]`

–initial: (optional) starting index (default 0)

–end: (required) stopping index (excluded)

–step: (optional) incrementation (default 1)

(Source: https://stackoverflow.com)

[4]:

```
lst = ['a', 'b', 'c', 'd', 'e', 'f']
word = 'abcdef'

# lst[6]  # IndexError
# word[6]

print(lst[:])  # equals the whole list
print(lst[6:])
print()

print(lst[:-3])  # equivalent to lst[0:-3:1]
print(word[:-3])
print()
print(lst[1:-1])
print(word[1:-1])
print()

print(lst[::2])  # step=2
print(word[::2])
print()

print(lst[-3:-1])
print(lst[-1:-3])
print()

print(lst[:-3:-1])
print(lst[-1:-3:-1])
print()

print(lst[:3])
print(lst[3:])
print()
```

```
['a', 'b', 'c', 'd', 'e', 'f']
[]

['a', 'b', 'c']
abc

['b', 'c', 'd', 'e']
bcde

['a', 'c', 'e']
ace

['d', 'e']
[]

['f', 'e']
['f', 'e']

['a', 'b', 'c']
['d', 'e', 'f']
```

[5]:

```
# String strip
str1 = '  abc   abc  '
str2 = ' abc    abc '
str3 = '#  #abc   abc  '
```

```

#remove all leading and tailing whitespace
print(str1.strip())
print(str2.strip())

#remove all leading and tailing character '#'
print(str3.strip('#'))
```

```
abc   abc
abc    abc
  #abc   abc
```

2 Dictionary

- (key, value) pair sequence
- not ordered

[6]:
```
# Define an empty dictionary
dict1 = {}
dict2 = dict()

# Define a dictionary with elements
dict3 = {'key1':'val1', 'key2':'val2'}
```

[7]:
```
# Adding elements to dictionary
purse = dict()
purse['money'] = 12
purse['candy'] = 3
purse['tissues'] = 75

print(purse)
print(purse['candy'])
# print(purse['pencils'])  # KeyError since the key does not exist

purse['candy'] += 2
print(purse)
```

```
{'money': 12, 'candy': 3, 'tissues': 75}
3
{'money': 12, 'candy': 5, 'tissues': 75}
```

`get()` method

`get(key, default_value)`

To get the value of the wanted key, or give a default value if the key is not found

[8]:
```
# get method
count = {'a':1, 'b':2, 'c':3}
```

```
print(count.get('a', 0))
print(count.get('e', 5))

count['f'] = count.get('f', 6)
print(count)
```

```
1
5
{'a': 1, 'b': 2, 'c': 3, 'f': 6}
```

[9]:

```
numList = [3, 44, 45, 11, 5, 11]
countDict = dict()  # dictionary to count occurences
for num in numList:
  countDict[num] = countDict.get(num,0) + 1

print(countDict)
```

```
{3: 1, 44: 1, 45: 1, 11: 2, 5: 1}
```

Retrieving keys and values in dictionary

[10]:

```
# Viewing elements in dictionary
print(list(purse))  # list of keys
print(list(purse.keys()))  # list of keys
print(list(purse.values()))  # list of values
print(list(purse.items()))  # list of (key, value) pairs
print()

for key, value in purse.items():
  print(key, value)
print()
for key in purse:
  print(key, purse[key])
```

```
['money', 'candy', 'tissues']
['money', 'candy', 'tissues']
[12, 5, 75]
[('money', 12), ('candy', 5), ('tissues', 75)]

money 12
candy 5
tissues 75

money 12
candy 5
tissues 75
```

3 Practice Questions

Please try by yourself before looking into the answers~

3.1 Q1: Count Occurences

[ ]:

```python
integerLine=input("Enter several integers in the interval [1,9](separated by space):\n")
integerList=integerLine.split()
countDict=dict()

for integer in integerList:
    countDict[integer]=countDict.get(integer,0)+1

for key,value in countDict.items():
    if value>1:
        print("%s occurs %d times."%(key,value))
    else:
        print("%s occurs %d time."%(key,value))
```

3.2 Q2: Display distinct numbers

[ ]:

```python
numberLine=input("Enter ten numbers(separated by space):\n")
numberList=numberLine.split()
distinctnumber=[]

for number in numberList:
    if number not in distinctnumber:
        distinctnumber.append(number)

print("The distinct numbers are:",end='')
for number in distinctnumber:
    print(number,end=' ')
```

3.3 Q3: Compute mean and deviation

[ ]:

```python
def mean(x):
    return sum(x)/len(x)

def deviation(x):
    deviationList=[(xi-mean(x))**2/(len(x)-1) for xi in x]
    deviation=sum(deviationList)**0.5
    return deviation

def main():
    numberLine=input("Enter numbers whose mean and deviation you want to calculate(separated by space):\n")
    numberList=numberLine.split()
##  A quick way to change a list [x1,x2,x3,...] to a new list
##  [f(x1),f(x2),f(x3),...] where f is some function or mapping.
    numberList=[eval(x) for x in numberList]
    print("The mean is",mean(numberList))
    print("The standard deviation is",deviation(numberList))

main()
```

3.4 Q4: Test sorted list

[ ]:

```
def isSorted(lst):
    for i in range(len(lst)-1):
        if lst[i]>lst[i+1]:
            return False
    return True

def main():
    lst=input("Enter elements in a list separated by space:\n").split()
    lst=[eval(x) for x in lst]
    if isSorted(lst):
        print("The list is already sorted.")
    else:
        print("The list is not sorted.")

main()
```

3.5 Q5: Word recitation

[ ]:

```
import random

file=open("Dictionary.txt",'r')
words={}

for line in file.readlines():
    word=line.split(':')
    words[word[1].strip()]=word[0].strip()

k=list(words.keys())
total=len(k)
count=0  ##Use "count" to count the correct guesses

##Notice that bool([])->False
while k:
    i=random.randint(0,len(k)-1)
##  Print how many words you have guessed
```

```
    print('(%d/%d)'%(total-len(k)+1,total),k[i])
    answer=input('Please guess the word:')

    if answer==words[k[i]]:
        print("√√√\nYour answer is correct!\n")
        count+=1
    else:
        print("×××\nThe correct answer should be \"%s\".\n"%words[k[i]])
##  Remove the word that you have just guessed
    k.remove(k[i])

print('Finished!You have correctly guess %d out of %d words!'%(count,total))

file.close()
```