Assignment 6

December 1, 2023

1 Assignment 6

1.0.1 Wanida Ruangsiriluk

1.0.2 November 30, 2023

1.1 Question 1

1) import the random library.
2) Use `random.seed(10)` to initialize a pseudorandom number generator.
3) Create a list of 50 random integers from 0 to 15. Call this list `int_list`.
4) Print the 10th and 30th elements of the list.

You will need to use list comprehension to do this. The syntax for list comprehension is: = `[<expression> for <item> in <iterable>]`. For this question your expression will be a randint generator from the random library and your iterable will be `range()`. Researh the documentation on how to use both functions.

[2]:
```
# 1 - 3
import random
random.seed(10)

int_list = []
n = 50
for i in range(n):
 int_list.append(random.randint(0,15))
print(int_list)
```

```
[1, 13, 15, 0, 6, 14, 15, 8, 5, 1, 15, 10, 2, 7, 11, 1, 13, 4, 11, 12, 13, 9, 8,
14, 5, 9, 11, 4, 14, 7, 14, 12, 1, 0, 7, 4, 6, 9, 11, 7, 10, 14, 13, 15, 2, 10,
5, 7, 13, 7]
```

[3]:
```
# 4

for i in [10, 30]:
    print(int_list[i])
```

```
15
14
```

1.2 Question 2

1) import the string library.
2) Create the string `az_upper` using `string.ascii_uppercase`. This is a single string of uppercase letters
3) Create a list of each individual letter from the string. To do this you will need to iterate over the string and append each letter to the an empty list. Call this list `az_list`.
4) Print the list.

You will need to use a for-loop for this. The syntax for this for-loop should be:

```
for i in string>:    <list operation>
```

[4]:
```
import string
az_upper = string.ascii_uppercase


az_list = []

for i in az_upper:
    az_list.append(i)

print(az_list)
```

```
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
```

1.3 Question 3

1) Create a set from 1 to 5. Call this `set_1`.
2) Create a set from int_list. Call this `set_2`.
3) Create a set by finding the `symmetric_difference()` of `set_1` and `set_2`. Call this `set_3`.
4) What is the length of all three sets?

[8]:
```
set_1 = {1,2,3,4,5}
set_2 = set(int_list)
set_3 = set_1.symmetric_difference(set_2)

#4. The length of set_1 is 5, set_2 is 15, set_3 is 12
len(set_1)
```

[8]: 5

[9]:
```
len(set_2)
```

[9]: 15

[10]:
```
len(set_3)
```

[10]: 12

1.4 Question 4

1) Import default dict and set the default value to 'Not Present'. Call this dict_1.
2) Add `int_list`, `set_2`, and `set_3` to `dict_1` using the object names as the key names.
3) Create a new dictionary, `dict_2`, using curly bracket notation with `set_1` and `az_list` as the keys and values.
4) Invoke the default value of `dict_1` by trying to access the key `az_list`. Create a new set named `set_4` from the value of `dict_1['az_list']`. What is the lenght of the difference between `dict_2['az_list']` and 'set_4'?
5) Update `dict_2` with `dict_1`. Print the value of the key `az_list` from `dict_2`. What happened?

[14]:
```
#1-3

from collections import defaultdict
def def_value():
    return 'Not Present'

dict_1 = defaultdict(def_value)

dict_1['int_list'] = int_list
dict_1['set_1'] = set_1
dict_1['set_2'] = set_2
dict_1['set_3'] = set_3

dict_2 = {'set_1': set_1, 'az_list': az_list}


#4
print(dict_1['az_list'])
```

```
Not Present
```

[15]:
```
#4 continues

set_4 = set(dict_1['az_list'],)

len(set_4)
```

[15]: 9

[16]:
```
#4 continues
len(dict_2['az_list'])

#The length of set_4 is 9 and length of len(dict_2['az_list']) is 26.  The␣
 ↪differences is 17
```

[16]: 26

[17]:
```
#5
dict_2.update(dict_1)

print(dict_2['az_list'])
# dictionary 2 retrieved the values of 'az_list' key, as a value "Not Present",␣
 ↪so the 'az_list' key of dict_2 was
# overwritten.
```

```
Not Present
```

[ ]: