

# Assignment 6

November 24, 2024

## 1 Assignment 6

### 1.0.1 Auriana Anderson

### 1.0.2 November 24, 2024

### 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()`. Research the documentation on how to use both functions.

```
[1]: import random
```

- 2) Use `random.seed(10)` to initialize a pseudorandom number generator.

```
[2]: random.seed(10)
```

- 3) Create a list of 50 random integers from 0 to 15. Call this list `int_list`.

```
[5]: int_list = [random.randint(0,15) for _ in range(50)] = [random.randint(0,15) for _  
↪ _ in range(50)]
```

- 4) Print the 10th and 30th elements of the list.

```
[11]: print(int_list[9])  
print(int_list[29])
```

1  
7

### 1.2 Question 2

- 1) import the string library.

```
[12]: import string
```

2) Create the string `az_upper` using `string.ascii_uppercase`. This is a single string of upper-case letters

```
[14]: az_upper = string.ascii_uppercase  
  
print(az_upper)
```

ABCDEFGHIJKLMNOPQRSTUVWXYZ

[ ]: 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` `for i in string`:  
<list operation>.

```
[20]: az_list = []  
for i in az_upper:  
    az_list.append(i)
```

[ ]: 4) Print the `list`.

```
[19]: 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`.

```
[22]: set_1 = {1,2,3,4,5}  
  
print(set_1)
```

{1, 2, 3, 4, 5}

2) Create a set from `int_list`. Call this `set_2`.

```
[23]: set_2 = set(int_list)  
print(set_2)
```

{0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}

3) Create a set by finding the `symmetric_difference()` of `set_1` and `set_2`. Call this `set_3`.

```
[26]: set_3 = set_1.symmetric_difference(set_2)  
  
print(set_3)
```

{0, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}

4) What is the length of all three sets?

```
[29]: print(len(set_1))
      print(len(set_2))
      print(len(set_3))
```

```
5
15
12
```

#### 1.4 Question 4

1) Import default dict and set the default value to 'Not Present'. Call this dict\_1.

```
[34]: from collections import defaultdict

      def def_value():
          return "Not present"

      dict_1 = defaultdict(def_value)

      print(dict_1[""])
```

```
Not present
```

2) Add int\_list, set\_2, and set\_3 to dict\_1 using the object names as the key names.

```
[38]: dict_1['int_list'] = int_list
      dict_1['set_2'] = set_2
      dict_1['set_3'] = set_3

      print(dict_1)
```

```
defaultdict(<function def_value at 0x00000210EC72FA60>, {'': 'Not present',
'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], 'set_2': {0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12,
13, 14, 15}, 'set_3': {0, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}})
```

3) Create a new dictionary, dict\_2, using curly bracket notation with set\_1 and az\_list as the keys and values.

```
[42]: dict_2 = {'set_1':set_1, 'az_list':az_list}

      print(dict_2)
```

```
{'set_1': {1, 2, 3, 4, 5}, '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']}
```

- 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 length of the difference between `dict_2['az_list']` and `'set_4'`?

```
[53]: print(dict_1['az_list'])

set_4 = set(dict_1['az_list'])

az_set = set(dict_2['az_list'])

difference_sets = az_set - set_4

print(len(difference_sets))
```

Not present  
25

- 5) Update `dict_2` with `dict_1`. Print the value of the key `az_list` from `dict_2`. What happened?

```
[55]: dict_2.update(dict_1)

print(dict_2['az_list'])
```

Not present

The original values of `az_list` was overwritten to "Not present"