

Simonsen.module06lab01

April 19, 2024

1 Assignment 6

1.0.1 Steven Simonsen

1.0.2 4/21/24

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.

```
[31]: import random as rd
```

```
[32]: rd.seed(10)
```

```
[33]: int_list = [(rd.randint(0, 15)) for i in range(50)]
      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]
```

```
[34]: print(int_list[10], int_list[30])
```

```
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>
```

```
[35]: import string as st
```

```
[36]: az_upper = st.ascii_uppercase
      print(az_upper)
```

ABCDEFGHIJKLMNOPQRSTUVWXYZ

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

      print(az_list)
```

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

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

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

```
{1, 2, 3, 4, 5}
```

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

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

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

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

```
[41]: 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`.
- 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 length 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?

```
[42]: from collections import defaultdict as dd
def def_value():
    return 'Not Present'
dict_1=dd(def_value)
```

```
[43]: dict_1.update(int_list=int_list, set_2=set_2, set_3=set_3)
print(dict_1)
```

```
defaultdict(<function def_value at 0x0000017FFE3BB100>, {'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}})
```

```
[44]: dict_2=dict({'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']}
```

```
[45]: print(dict_1['az_list'])
set_4=set(dict_1['az_list'])
print(len(dict_2['az_list'])-len(set_4))
```

```
Not Present
17
```

```
[46]: dict_2.update(dict_1)
print(dict_2['az_list'])
#The key:value pair weas overwritten in dict_2, thus printing "Not Present". New
↪key:value pairs were added to the dictionary.
```

```
Not Present
```