Dictionaries Continued and Anonymous Functions (Lambdas) and Higher Order Functions

February 2, 2023

[ ]:
```
# dict.keys(), dict.values(), dict.items()
# if we want to set all the keys in the dict
# to a single value
# fromkeys()
```

[5]:
```
lst = [10, 20, 30, 40]
d = {}.fromkeys(lst, 5)
print(d)
```

```
{10: 5, 20: 5, 30: 5, 40: 5}
```

[6]:
```
string = 'hello'
d = {}.fromkeys(string, 1)
print(d)
```

```
{'h': 1, 'e': 1, 'l': 1, 'o': 1}
```

[7]:
```
d = {'h': 1, 'e': 1, 'l': 1, 'o': 1}
d.clear()
print(d)
```

```
{}
```

[8]:
```
# dict.get()
# get the corresponding value if key is present
# get the default value (None)
d = {'h': 1, 'e': 1, 'l': 1, 'o': 1}
print(d.get('h'))
```

```
1
```

[9]:
```
# dict.get()
# get the corresponding value if key is present
# get the default value (None)
d = {'h': 1, 'e': 1, 'l': 1, 'o': 1}
print(d.get('z'))
```

```
None
```

[10]:
```
# dict.get()
# get the corresponding value if key is present
# get the default value (None)
d = {'h': 1, 'e': 1, 'l': 1, 'o': 1}
print(d.get('z', 'xyz'))
```

```
xyz
```

[11]:
```
# dict.get()
# get the corresponding value if key is present
# get the default value (None)
d = {'h': 1, 'e': 1, 'l': 1, 'o': 1}
print(d.get('h', -1))
```

```
1
```

[12]:
```
string = 'abcdabbcccdddef'
d = {}
for i in string:
    if i in d.keys():
        d[i] += 1
    else:
        d[i] = 1
print(d)
```

```
{'a': 2, 'b': 3, 'c': 4, 'd': 4, 'e': 1, 'f': 1}
```

[13]:
```
string = 'abcdabbcccdddef' #
d = {} # {'a': 1}
for i in string: # 'a'
    d[i] = d.get(i, 0) + 1 # d['a'] = 2
print(d)
```

```
{'a': 2, 'b': 3, 'c': 4, 'd': 4, 'e': 1, 'f': 1}
```

[15]:
```
d = {}
d['a'] = 1
d['a'] = 2
print(d)
```

```
{'a': 2}
```

1 Anonymous Functions (lambda)

- When we require a simple function for a specific amount of time

[16]:
```
def cube(n):
    return n * n * n
print(cube(10))
```

```
1000
```

[17]:
```
# lambda arguments: expression
(lambda n: n * n * n)(8)
```

[17]:
```
512
```

[18]:
```
x = lambda n: n * n * n
print(x(8))
```

```
512
```

[19]:
```
(lambda a, b: a + b)(10, 20)
```

[19]:
```
30
```

1.1 Higher Order Functions

- map()
- map(func, iterable)
- Apply the given function on each member of the iterable and returns a map object.

[21]:
```
lst = ['abc', 'defgh', 'ijklmn']
# [3, 5, 6]
m = list(map(len, lst)) # map(len, ['abc', 'defgh', 'ijklmn'])
print(m)
```

```
[3, 5, 6]
```

[27]:
```
print(''.join(list(map(chr, range(97, 123)))))
```

```
abcdefghijklmnopqrstuvwxyz
```

[28]:
```
# User defined functions
def cube(n):
    return n * n * n
nums = [10, 20, 30, 40]
print(list(map(cube, nums)))
```

```
[1000, 8000, 27000, 64000]
```

[34]:
```
# User defined functions
nums = [10, 20, 30, 40]
print(list(map(lambda n: n * n * n, nums)))
```

```
[1000, 8000, 27000, 64000]
```

[29]:
```
import random
marks = [[random.randint(25, 95) for i in range(3)] for j in range(5)]
marks
```

[29]:
```
[[62, 47, 75], [67, 91, 29], [74, 80, 49], [38, 27, 62], [95, 70, 66]]
```

[35]:
```
marks = [[62, 47, 75],
         [67, 91, 29],
         [74, 80, 49],
         [38, 27, 62],
         [95, 70, 66]]
summary = list(map(lambda l: [min(l), max(l), sum(l)], marks))
summary
```

[35]:
```
[[47, 75, 184], [29, 91, 187], [49, 80, 203], [27, 62, 127], [66, 95, 231]]
```

[32]:
```
def get_details(lst):
    return [min(lst), max(lst), sum(lst)]
```

[32]:
```
[49, 80, 203]
```

[36]:
```
import math
lst = [5, 6, 7, 3]
fact = list(map(math.factorial, lst))
print(fact)
```

```
[120, 720, 5040, 6]
```

[39]:
```
lst = [10, 20, 30]
print(list(map(len, lst)))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_26680\3038015942.py in <cell line: 2>()
      1 lst = [10, 20, 30]
----> 2 print(list(map(len, lst)))

TypeError: object of type 'int' has no len()
```

[37]:
```
len(10)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_26680\3338265758.py in <cell line: 1>()
----> 1 len(10)

TypeError: object of type 'int' has no len()
```