Maktab
Sharif

Python | Main course

Session 5

- Review
- Dictionary
- Exercise

by Mohammad Amin H.B. Tehrani

www.maktabsharif.ir

Review

Example 1

What’s result of code below

```
def is_primal(n):
    n = int(n)
    assert n > 0
    for i in range(2, int(n**0.5) +1):
        if not (n%i):
            return False
    return True

l = list(map(lambda x:is_primal(x), range(2, 100)))
print(any(l), all(l))
```

Example 1

Maktab Sharif

What’s result of code below

```
def is_primal(n):
    n = int(n)
    assert n > 0
    for i in range(2, int(n**0.5) +1):
        if not (n%i):
            return False
    return True

l = list(map(lambda x:is_primal(x), range(2, 100)))
print(any(l), all(l))
```

```
True False
```

Example 2

Maktab Sharif

What’s result of code below

```
s1 = list(range(10))
s2 = "Hello Ali!"
x1 = zip(s1, s2)
x2 = enumerate(s2)
print(list(x1) == list(x2))
```

Example 2

Maktab Sharif

What's result of code below

```
s1 = list(range(10))
s2 = "Hello Ali!"
x1 = zip(s1, s2)
x2 = enumerate(s2)
print(list(x1) == list(x2))
```

```
True
```

Example 3

What's result of code below

```
s = "1 - Hello World /n2-  Hello Akbar +" \
    "3 Hello Reza"
x = list(filter(lambda x: not x.isalpha(), s))
print(x)
x = list(reversed(x))
print(x)
x = [str(ord(_)) if _.isnumeric() else _ for _ in x if _ != ' ']
print(x)
x = ''.join(x)
print(x)
x = eval(x)
print(x)
x = round(x, 3)
print(x)
x = hex(int(x * 10))
print(x)
```

Example 3

Maktab Sharif

Result:

```
['1', ' ', '-', ' ', ' ', ' ', '/', '2', '-', ' ', ' ', ' ', ' ', '+', '3', ' ', ' ']
[' ', ' ', '3', '+', ' ', ' ', ' ', ' ', '-', '2', '/', ' ', ' ', ' ', '-', ' ', '1']
['51', '+', '-', '50', '/', '-', '49']
51+-50/-49
52.02040816326531
52.02
0x208
```

Chapter 11

Dictionary

Dictionary (dict)

Maktab Sharif

Dictionaries are used to store data values in KEY:VALUE pairs.

A dictionary is a collection which is **ordered***, **changeable** and **does not allow duplicates**.
As of Python version 3.7, dictionaries are *ordered*. In Python 3.6 and earlier, dictionaries are *unordered*.

Syntax:

```
my_dict = {key1:value1, key2:value2, key3:value3 , … }
```

```
my_dict = {
    'a_str_key': 'Anything can be a value in python dicts...',
    85: 32,
    True: 15.67,
    (1, 2, 3): 1,
    [1, 2, 3]: 2,
}
print(my_dict)  #???
```

Access to Dictionary Items

Maktab Sharif

1) Use access operator: ur_dict[key]

Dictionary items are presented in key:value pairs, and can be referred to by using the key name.

2) Use .get() method: ur_dict.get(key, ...)

You can call .get() method with a default value. if key not found in the dict, default value will return.

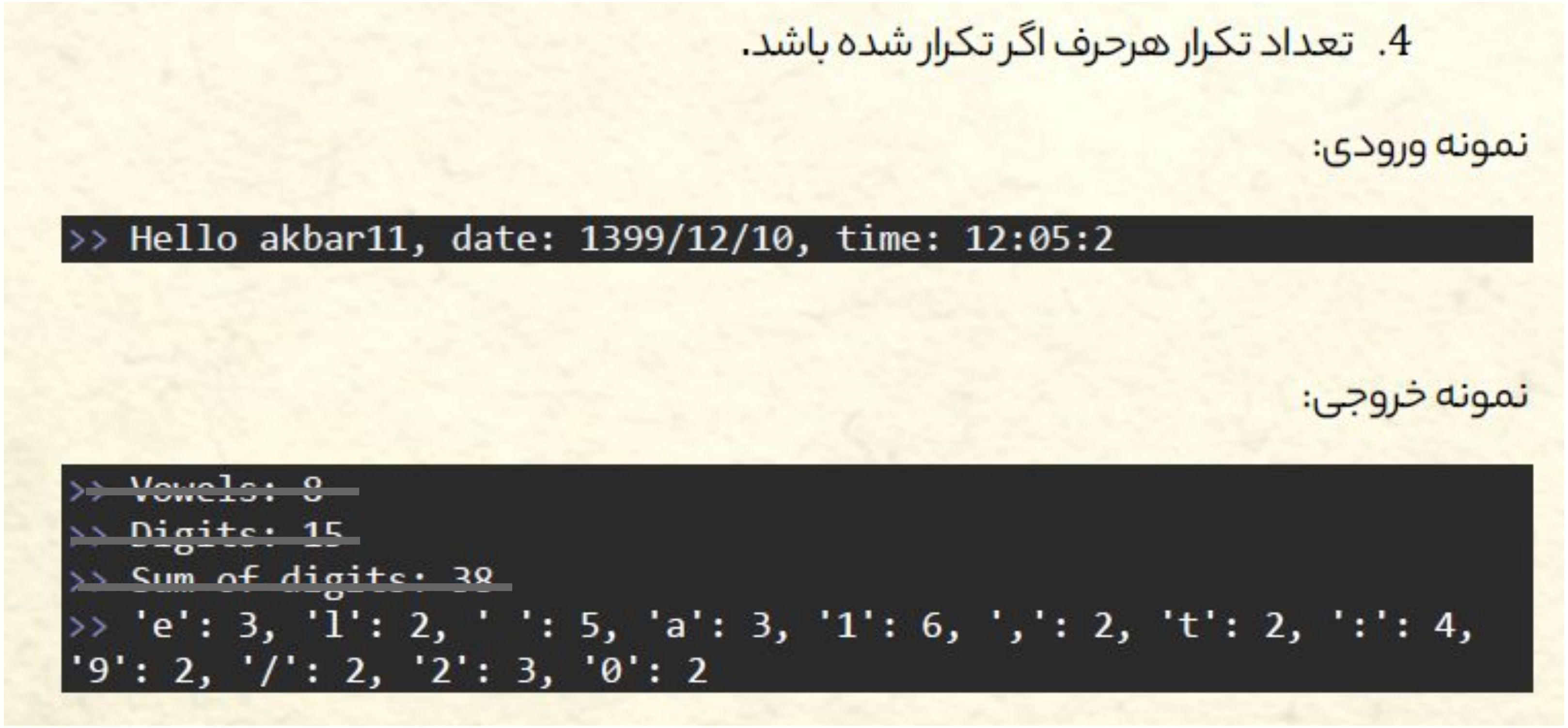
```
print(my_dict[85])
print(my_dict['a_str_key'])
print(my_dict[True])
print(my_dict[(1, 2, 3)])
print(my_dict['Name'])

print(my_dict.get(85))
print(my_dict.get('a_str_key'))
print(my_dict.get('Name', 'Akbar'))
```

Some Dictionary methods

Maktab Sharif

clear()	Removes all the elements from the dictionary
copy()	Returns a copy of the dictionary
fromkeys()	Returns a dictionary with the specified keys and value
get()	Returns the value of the specified key
items()	Returns a list containing a tuple for each key value pair
keys()	Returns a list containing the dictionary's keys
pop()	Removes the element with the specified key
popitem()	Removes the last inserted key-value pair
update()	Updates the dictionary with the specified key-value pairs
values()	Returns a list of all the values in the dictionary

Example



4. تعداد تکرار هرحرف اگر تکرار شده باشد.

نمونه ورودی:

```
>> Hello akbar11, date: 1399/12/10, time: 12:05:2
```

نمونه خروجی:

```
>> ~~Vowels: 8~~
>> ~~Digits: 15~~
>> ~~Sum of digits: 38~~
>> 'e': 3, 'l': 2, ' ': 5, 'a': 3, '1': 6, ',': 2, 't': 2, ':': 4,
'9': 2, '/': 2, '2': 3, '0': 2
```

Example: Code

Maktab Sharif

```
s = "Hello akbar11, date: 1399/12/10, time: 12:05:2"

d = {}
for _ in s:
    d[_] = d.get(_, 0) + 1

# Dict Comprehension
print({key: value for key, value in d.items() if value > 1})
```

Example: *args , **kwargs

Maktab Sharif

What's result of code below

```
def save_information(first_name, last_name, phone, *marks, **extra_info):
    print(first_name)
    print(last_name)
    print(phone)
    print(type(marks), marks)
    print(type(extra_info), extra_info)
    ...


save_information('Reza',
                 'Bahadori',
                 '099999999999',
                 20, 12, 3, 5,
                 email='reza@bahadori...', username='R.BAHADOR'
                 )
```

Exercises

Exercise: Guess the number

Maktab Sharif

Guess the number!

Write a console game, that tries to guess the user's number.

Your program can ask the user 3 types of questions:

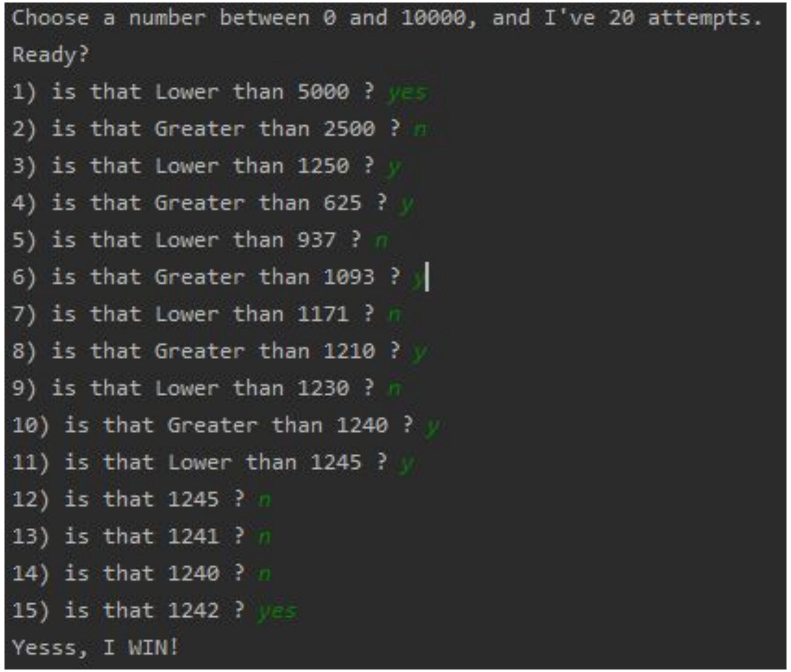
1) Is that Greater than x? (x > guess ?)
2) Is that Less than x? (x < guess ?)
3) Is that x? (x == guess ?)

Bound = 0 - 10,000

Attempts = 20

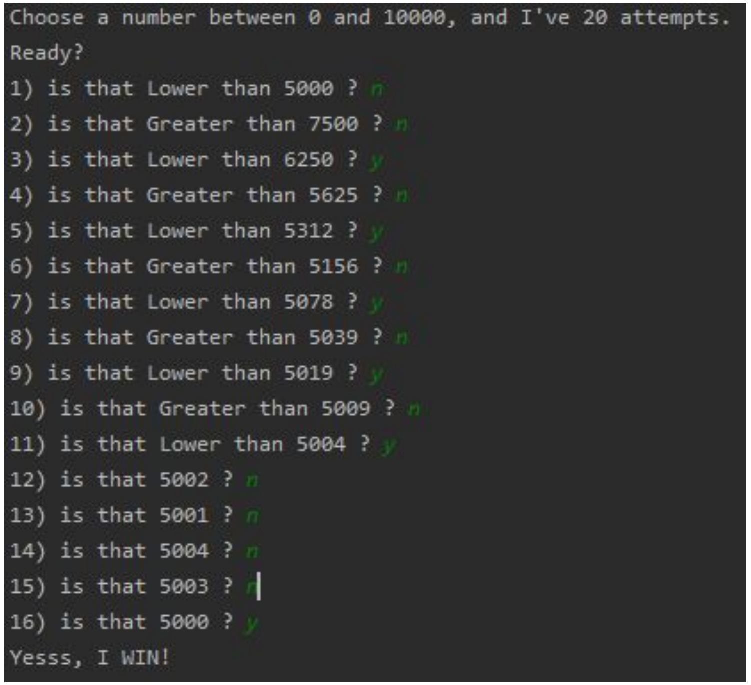
Exercise: Guess the number

Number: 6759



```
Choose a number between 0 and 10000, and I've 20 attempts.
Ready?
1) is that Lower than 5000 ? yes
2) is that Greater than 2500 ? n
3) is that Lower than 1250 ? y
4) is that Greater than 625 ? y
5) is that Lower than 937 ? n
6) is that Greater than 1093 ? y
7) is that Lower than 1171 ? n
8) is that Greater than 1210 ? y
9) is that Lower than 1230 ? n
10) is that Greater than 1240 ? y
11) is that Lower than 1245 ? y
12) is that 1245 ? n
13) is that 1241 ? n
14) is that 1240 ? n
15) is that 1242 ? yes
Yesss, I WIN!
```

Number: 5000



```
Choose a number between 0 and 10000, and I've 20 attempts.
Ready?
1) is that Lower than 5000 ? n
2) is that Greater than 7500 ? n
3) is that Lower than 6250 ? y
4) is that Greater than 5625 ? n
5) is that Lower than 5312 ? y
6) is that Greater than 5156 ? n
7) is that Lower than 5078 ? y
8) is that Greater than 5039 ? n
9) is that Lower than 5019 ? y
10) is that Greater than 5009 ? n
11) is that Lower than 5004 ? y
12) is that 5002 ? n
13) is that 5001 ? n
14) is that 5004 ? n
15) is that 5003 ? n
16) is that 5000 ? y
Yesss, I WIN!
```