# Programming 202

## Hands-on Python Data Structures

**Kuldeep Singh Sidhu**

*kuldeepsinghsidhu.com*

# By the end of this session

- Dict

- We will practice **dict** and **list**
- We will do **hands-on exercise**

# dict

```
1.  # python dictionaries are a spacial datastructure
2.  # key:value pair (data)
3.
4.  # example an oxford dictionary word: meaning
5.  # you search the word which is the key
6.  # to get the meaning which is the value
7.
8.  # first focus on the syntax
9.  # {k:v,k:v,k:v} look at the 2 seperators : ,
10. lang = {'en':'english','fr':'french'}
11. print(lang, type(lang))
12.
13. # key is a word and value is list of synonyms
14. sameMeaning = {'pride': ['delight', 'dignity','honour','joy']}
```

# dict

```
1.  emptyDict = dict()
2.  emptyDict = {}
3.
4.  lang = {'en':'english','fr':'french'}
5.
6.  # reading
7.  print(lang['en'])
8.
9.  # adding
10. lang['tl'] = 'telugu'
11. print(lang)
12.
13. # updating >> dict are mutable
14. lang['tl'] = 'tamil'
15. print(lang)
16.
17. # why are add and update so similar
18. # if the key is not there it will add otherwise update
19. # this also means keys cannot be repeated but values can
20.
21. # delete
22. del lang['fr']
23. print(lang)
```

# dict

```
1.  # we have learned how to update data for
2.  # list, sets, dict  which are mutable
3.  # mutable data structures cannot be keys
4.  # values can be anything
5.
6.  # errorDict = {[1,2]:'list'}
7.  # errorDict = {{1,2}:'set'}
8.  # errorDict = {{'apple':'a fruit'}:'dict'}
9.
10. # you cannot ask for key that doesn't exist
11. lang = {'en':'english','fr':'french'}
12. # print(lang['pb'])
13. if 'pb' in lang:
14.     print(lang['pb'])
15. else:
16.     print('pb not present')
```

# dict

```
1.  lang = {'en':'english','fr':'french'}
2.  # using get
3.  print(lang['en'], lang.get('en'))
4.  # doesn't throw error if key doesn't exist
5.  print(lang.get('pb'))
6.  # default message
7.  print(lang.get('pb','key is not found'))
```

# dict

```
1.  # for loop in  dictionary
2.  lang = {'en':'english','fr':'french','cn':'chinese'}
3.
4.  for l in lang:
5.      print(l)
6.  # loop only prints keys
7.  # we can get the value using the key
8.  for l in lang:
9.      print(l,lang[l])
10.
11. # dictionaries in python are ordered
12. # getting keys and values
13. print(lang.keys(), lang.values())
14. print(list(lang.keys())[0])
15. print(lang[list(lang.keys())[0]])
```

# dict

```
1.  # .items()
2.  lang = {'en':'english','fr':'french','cn':'chinese'}
3.  print(lang.items())
4.  # we get key-value pairs in a list of tuples
5.  for myTuple in lang.items():
6.      print(myTuple)
7.  # unpacking the key-value pair tuple
8.  for myTuple in lang.items():
9.      key, val = myTuple
10.     print(key, val)
11. # directly unpack
12. for key, val in lang.items():
13.     print(key, val)
```

# indexing

```
1.  # indexing in lists, tuples, strings
2.  name='ronaldo'
3.  teams = ('rm','ju','mu')
4.  player = [name,teams]
5.  print(player, player[0])
6.  print('latest team', teams[-1])
7.
8.  print('name start with',name[0])
9.  print('reverse of the name is',name[::-1])
```

```python
#1 What is 7 raise to the power 4
print(7**4)

#2 Using string formatting print: diameter of earth is 12734
planet = 'earth'
diameter = 12734
print(f'diameter of {planet} is {diameter}')

#3 splitting strings into a list
s = 'this is python programming'
print(s.split()) # by default space is used as a delimiter

#4 get all the words in the string that contain *
# hint split and then use a loop
giftBox = 'bread iph*one butter ip*ad milk'
words = giftBox.split()
for w in words:
    if '*' in w:
        print(w)
```

```python
giftBox = 'bread iph*o**ne butter ip*a*d milk'
words = giftBox.split()
# print(words)
for w in words:
    # print(w)
    for c in w:
        if c=='*':
            print(w)
            break
```

```
1.  #5 filter colors starting with vowel a, e, i, o, u
2.  colors = 'red orange blue indigo green'
3.  c_list = colors.split()
4.  for color in c_list:
5.      print(color, '-->', color[0])
6.      if color[0] in 'aeiou':
7.          print(color, 'starts with vowel')
```



```
orange starts with vowel
indigo starts with vowel
```

```
# 6 Given a nested list, grab and print 'hello'
lst = [1,2,[3,4],[5,[100,200,[True,'hello'], False],'hi'],10,11]

print(lst[3][1][2][-1])

# 7 Given a nested dict, grab and print 'hello'
# use indexes, dict keys and splitting of strings
dct = {'here':[1,2,3,{'target':(False,{'color':'black','greetings':['hello world']})}]}

print(dct['here'][3]['target'][1]['greetings'][0][:5])
```

```
# 8 Guess the number
# generate a random between 1-10 and store it
# then ask the user to enter a number
# if the user choice is greater than computer choice say 'too high'
# if the user choice is smaller than computer choice say 'too low'
# if the user choice is equal to computer choice say 'correct guess'
# BONUS points: allow user a max of 3 guesses
# will need: use google to learn how to generate a random number between 1-10
```

```
1.  import random
2.  computer_choice=random.randint(1,10)
3.  attempts = 0
4.  while attempts<3:
5.      user_choice = int(input('Guess the number: '))
6.      if user_choice==computer_choice:
7.          print('correct guess')
8.          break
9.      elif user_choice>computer_choice:
10.         print('too high')
11.     else:
12.         print('too low')
13.     attempts+=1
14. if attempts>=3:
15.     print('Failed 3 times')
16. print('GAME OVER')
```

# range(start,stop,step=1)

```
1.  # range(start,stop,step=1)
2.  # stop is not included
3.  for i in range(5):
4.      print(i)
5.
6.  print ('even numbers from 1 to 15')
7.  for i in range(2,15,2):
8.      print(i)
9.
10. print ('numbers  5,4,3,2,1')
11. for i in range(5,0,-1):
12.     print(i)
```

```
1.  #9 coding number guessing game using for loop and range()
2.  import random
3.  computer_choice=random.randint(1,10)
4.  for i in range(3):
5.      user_choice = int(input('Guess the number: '))
6.      if user_choice==computer_choice:
7.          print('correct guess')
8.          break
9.      elif user_choice>computer_choice:
10.         print('too high')
11.     else:
12.         print('too low')
13. else: # this runs only when loop completes all cycles
14.     print('Failed 3 times')
15. print('GAME OVER')
```

# Python modules

```
1.  # what is random
2.  import random
3.  print(type(random))
4.  print(random)
```

# Arithmetic operations

```
1.  # Arithmetic operations
2.
3.  # +
4.  print(5+2)
5.
6.  # -
7.  print(5-2)
8.
9.  # *
10. print(5*2)
11.
12. # /
13. print(5/2)
```

```
1.  ## special operators
2.
3.  # power: raised to power >> **
4.  print(5**2)
5.
6.  # get floor division and get an integer result
7.  # largest integer smaller or equal to the
    division value
8.  # (discarding any fractional result) >> //
9.  print(int(5/2))
10. print(5//2)
11.
12. # modulo: get remainder >> %
13. print(5%2)
```

```
1.  #10 check if an integer input by the user is even OR odd
2.
3.  num= int(input('Enter your number: '))
4.
5.  # using floor division
6.  if num//2==num/2:
7.      print('even')
8.  else:
9.      print('odd')
10.
11. # using modulo
12. if num%2==0:
13.     print('even')
14. else:
15.     print('odd')
```


https://www.hackerrank.com/domains/python

# Practice

This is just the beginning