

# Python Language and its Applications

Prof. (Dr.) K.R. Chowdhary  
*Email: kr.chowdhary@jietjodhpur.ac.in*

Campus Director,  
JIET College of Engineering, Jodhpur

Wednesday 7<sup>th</sup> June, 2017

```
# hello-you.py
```

```
person = input('enter your name: ')\nprint('Hello', person)
```

```
$ python hello-you.py\nenter your name: 'krc'\n('hello', 'krc')
```

```
# birthday1.py  
  
print("Happy Birthday to you!")  
print("Happy Birthday to you!")  
print("Happy Birthday, dear Emily.")  
print("Happy Birthday to you!")
```

```
$ python birthday1.py  
Happy Birthday to you!  
Happy Birthday to you!  
Happy Birthday, dear Emily.  
Happy Birthday to you!
```

```
# birthday2.py
```

```
def happyBirthdayEmily():  
    print("Happy Birthday to you!")  
    print("Happy Birthday to you!")  
    print("Happy Birthday to you!")
```

Run in Python by:

```
>>> from birthday2 import happyBirthdayEmily  
>>> happyBirthdayEmily()  
Happy Birthday to you!  
Happy Birthday to you!  
Happy Birthday to you!
```

```
# textproc2.py

def plural(word):
    if word.endswith('y'):
        return word[:-1] + 'ies'
    elif word[-1] in 'sx' or word[-2:] in ['sh', 'ch']:
        return word + 'es'
    elif word.endswith('an'):
        return word[:-2] + 'en'
    else:
        return word + 's'

>>> plural('fairy')
'fairies'
>>> plural('woman')
'women'
```

Save your function(s) in a file called (say) **textproc.py**. Now, you can access your work simply by importing it from the file:

```
>>> from textproc import plural
>>> plural('wish')
wishes
>>> plural('fan')
fen
```

A collection of variable and function definitions in a file is called a [Python module](#). A collection of related modules is called a [package](#).

```
# sump.py

def sumProblem(x, y):
    sum = x + y
    print('The sum of ', x, ' and ', y, ' is ', sum, '.')

def main():
    sumProblem(2, 3)
    sumProblem(1234567890123, 535790269358)
    a = int(input("Enter an integer: "))
    b = int(input("Enter another integer: "))
    sumProblem(a, b)
```

```
# summ.py

from sump import sumProblem
from sump import main

sumProblem(5, 10)
main()
```

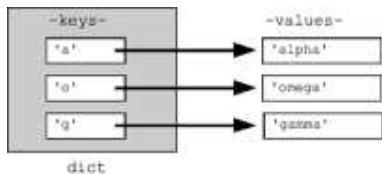


# Returned function values

```
def f(x):  
    return x*x  
  
print(f(3))  
print(f(3) + f(4))
```

# Dictionary

A dictionary is an unordered set of key: value pairs, with the requirement that the keys are unique (within one dictionary)



```
>>> pos = {}
>>> pos
```

```
{
>>> pos['colorless'] = 'ADJ'
>>> pos
{'colorless': 'ADJ'}
>>> pos['ideas'] = 'N'
>>> pos['sleep'] = 'V'
>>> pos['furiously'] = 'ADV'
>>> pos
{'furiously': 'ADV', 'ideas': 'N',
'colorless': 'ADJ', 'sleep': 'V'}
>>> pos['ideas']
'N'
>>>
```

To just find the keys, we can either convert the dictionary to a list or use the dictionary in a context where a list is expected, as the parameter of `sorted()`, or in a for loop.

```
>>> list(pos)
['ideas', 'furiously', 'colorless', 'sleep']
>>> sorted(pos)
['colorless', 'furiously', 'ideas', 'sleep']
>>> [w for w in pos if w.endswith('s')]
['colorless', 'ideas']
```

```
>>> for word in sorted(pos):
    print word + ":", pos[word]

...
colorless: ADJ
furiously: ADV
sleep: V
ideas: N
>>> pos.keys()
['colorless', 'furiously', 'sleep', 'ideas']
>>> pos.values()
['ADJ', 'ADV', 'V', 'N']
>>> pos.items()
[('colorless', 'ADJ'), ('furiously', 'ADV'), ('sleep', 'V'),
 ('ideas', 'N')]
>>> for key, val in sorted(pos.items()):
    print key + ":", val
colorless: ADJ
```

# Definition and Use of Dictionaries

```
# English to Hindi dictory
# ehdict.py
# Returns a tiny english-
# to hindi dictionary
def createDictionary():
#creates an empty dict.
    hindi = dict()
    hindi['who']='kon'
    hindi['yes']='haan'
    hindi['one']='eik'
    hindi['two']='tho'
    hindi['three']='teen'
    hindi['red']='lal'
```

```
hindi['black']='kala'
hindi['green']='hara'
hindi['blue']='aasmani'
return hindi
```

```
def main():
    dictionary = createDictionary()
    print(dictionary['two'])
    print(dictionary['red'])
    print(dictionary['black'])

#main() :to be called
```