

ISYS90088

Introduction to Application Development

Assignment Project Exam Help

Week 9 – Contd. from week 8 on Dictionaries;
Introduction to Functions

<https://powcoder.com>

Week 10: Contd. with functions

Add WeChat powcoder

Semester 2 , 2018

Dr Antonette Mendoza

Other Dictionary methods: **pop**

- Use the built in method called **pop**
- When executed, it outputs the value associated with the specific key and removes it from the dict.

Syntax: [Assignment Project Exam Help](https://powcoder.com)

`dictionary_name.pop(key, default)`

<https://powcoder.com>

Example:

[Add WeChat powcoder](https://powcoder.com)

```
>>> employee_record = {'name': 'kevin', 'Age': 43, 'ID': 23145, 'payrate': 24.99}
```

```
>>> employee_record.pop('name')  
'kevin'
```

```
>>> employee_record  
{ 'payrate': 24.99, 'ID': 23145, 'Age': 43 }  
>>>
```

Other Dictionary methods: **popitem**

- Use the built in method called **popitem**
- When executed, it outputs a selected key-value pair, and it removes that key-value pair from the dict. (front of the list/dict)

Syntax:

```
dictionary_name.popitem()
```

Example:

```
>>> employee_record = {'payrate': 24.99, 'name': 'kevin', 'ID': 23145, 'Age': 45}
>>> employee_record.popitem()
('payrate', 24.99)
>>> employee_record.popitem()
('name', 'kevin')
>>> employee_record.popitem()
('ID', 23145)
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Recap -Traversing: using list and dict methods

```
>>> employee_record = {'name':'kevin',  
    'Age': 43, 'ID':23145, 'payrate':24.99}
```

```
>>> list(employee_record.keys())  
['payrate', 'name', 'ID', 'Age']
```

Assignment-Project Exam Help
<https://powcoder.com>

```
>>> list(employee_record.values())  
[24.99, 'kevin', 23145, 43]
```

Add WeChat powcoder

```
>>> list(employee_record.items())  
[('payrate', 24.99), ('name', 'kevin'),  
 ('ID', 23145), ('Age', 43)]
```

Note: defaultdict when working with lists and dictionaries

- It is easy to group a sequence of key-value pairs into a dictionary of lists using `defaultdict` from a collections library:

```
>>> from collections import defaultdict
>>> s = [('yellow', 1), ('blue', 2), ('yellow', 3),
        ('blue', 4), ('red', 1)]
>>> d = defaultdict(list)
>>> for k, v in s:
...     d[k].append(v)
...
>>> d.items()
[('blue', [2, 4]), ('red', [1]), ('yellow', [1, 3])]
```

- When each key is encountered for the first time, it is not already in the mapping; so an entry is automatically created using the `default_factory` function which returns an empty list.
- The `list.append()` operation then attaches the value to the new list. When keys are encountered again, the look-up proceeds normally (returning the list for that key) and the `list.append()` operation adds another value to the list.

Example: creating a dict from a list using **append**

```
from collections import defaultdict
```

```
employee_list = [('yosh', 23, 2001), ('farah', 22,  
2010), ('matt', 34, 2000)]
```

~~#you can take a list of tuples and make it #into a
dict with key-value pairs~~

~~#start with an empty dict, start a for loop #and
append values to a key~~

```
d1 = defaultdict(list)
```

```
for key, age, start_date in employee_list:
```

```
    d1[key].append(age)
```

```
    d1[key].append(start_date)
```

```
print(d1, d1.items(), d1.values())
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Dictionary of dictionaries

- In general no matter how many dictionaries are there inside dictionary like n level dictionaries but you can always get it by the path.

For example,

`d['dictionary1']['dictionary2']['dictionary3']....['item1']`

Assignment Project Exam Help

- Inner dictionary can be accessed as: `dictA[primary key][secondary key]`

<https://powcoder.com>

Example:

```
>>> dictA = {'a': {'b': 10, 'c': 20}, 'd': {'b': 30, 'c': 40}}
>>> dictA['a']['c']
20
```

- So, `dict['a']['b']` will give you the value corresponding to inner dictionary's 'b' key.
- Nested dictionaries can be accessed just like we access the nested lists with the only difference that in lists, we use indexing whereas in dictionaries we do it via keys.

Assignment Project Exam Help
Functions

<https://powcoder.com>

Add WeChat powcoder

Functions

- What's a function? It is a group of statements that exist within a program for the purpose of performing a task

Assignment Project Exam Help

- (much like in Maths) functions take a set of input values, perform some calculation based on them, and optionally return a value
- you have already seen and used many functions by this stage, e.g.: `str()`, `len()`, `sqr()`, `min()`, `max()`...

Functions: usefulness

- Simpler code
 - Code re-use
 - Better testing
 - Faster development
 - Easier facilitation of teamwork
- Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

Functions: defining and calling

- The code for a function is known as a function **definition**. To execute a function, you write a statement that **calls** it. **Assignment Project Exam Help**
- In order to define a function, we need:
<https://powcoder.com>
Add WeChat powcoder
 - A function name (following same conventions as other variable names)
 - (optionally) a list of input variables
 - (optionally) a UNIQUE output object (via return)

Functions: defining and calling

Function names should be:

- Descriptive enough so that anyone reading your code can reasonably guess what the function does (prefer to use verbs)
<https://powcoder.com>
- Rules for naming:
[Add WeChat powcoder](#)
 - Cannot use python key words
 - Cannot contain spaces
 - First character must be letter; after the first character, you may use letters, digits, underscore
 - Case sensitive

Functions: defining

Basic syntax :

```
def <function_NAME>( INPUTLIST ) :  
    statement  
    statement  
    statement
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Example:

```
def message( ) :  
    print( 'this is a simple case' )
```

Functions: calling

- To execute a function, you must call the function.

`<function_name> ()`

Example: <https://powcoder.com>

```
def message():  
    print('this is a simple case')
```

```
message()
```

Functions: main() and functions

```
# a function to perform something
#the main function that might use the function
def main(): Assignment Project Exam Help
    message()
    https://powcoder.com
def message(): Add WeChat powcoder
    print('this is a simple case')

#execute main function
main()
```

The diagram consists of two orange rectangular boxes. The first box is positioned around the `def main():` block and contains the text **Assignment Project Exam Help** and the URL **https://powcoder.com**. An orange arrow points from the `message()` call inside the `main()` function to the top of this box. The second box is positioned around the `def message():` block and contains the text **Add WeChat powcoder**. An orange arrow points from the `print()` statement inside the `message()` function to the top of this box. A third orange arrow points from the bottom of the second box back to the `main()` function block, indicating a return flow.

Functions: simple examples

#This is a simple example to illustrate a function call

```
def main():  
    print("I have a message for you")  
    message()  
    print('Good bye')
```

```
def message():  
    print('this is Antonette')  
    print('can you hear me?')
```

```
#call main program  
main()
```


Example: two functions

Write two functions: one called `melbourne()` and another called `canberra()`. The functions must accept from the user the number of Small to Medium Enterprises (SME's) in these two cities. It must then print the number of SME's in each of the cities.

```
def main():
    melbourne()
    canberra()
#defining the two function
def melbourne():
    small_medium = input("the no. in melbourne are:")
    print('SMES in melbourne =', small_medium)
def canberra():
    small_medium = input("the no. in canberra are:")
    print('SMES in canberra =', small_medium)

main()
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Scope and local variables

- A variable's *scope* is the part of a program in which the variable may be accessed.
- **Local variable:** is created inside a function and cannot be accessed by statements that are outside the function.
 - Different functions within a program can have same variable names since the other functions cannot see or use each others *local variables*.
 - A local variable cannot be accessed by code that appears inside a function at a point before the variable has been created.

Example 1 (scope) – what happens here?

```
def main():  
    melbourne()  
    canberra()
```

Assignment Project Exam Help

```
#defining the functions
```

```
def melbourne():  
    birds = 1000  
    print('melbourne has', birds, 'birds')
```

```
def canberra():  
    birds = 870  
    print('canberra has', birds, 'birds')
```

```
# calling the main function  
main()
```

Examples: 2 – what's the problem here?

```
def main():  
    get_name()  
    print('hello', name) #causes an error
```

```
def get_name():  
    name = input('enter your name:')
```

Assignment Project Exam Help

```
main()
```

<https://powcoder.com>

#another scope example - issue. A local variable cannot be accessed by code that appears!!!! Why???.
Add WeChat powcoder

```
def bad_function():  
    print("the value is", val) # causes an error  
    val = 99
```

```
bad_function()
```

Functions: passing arguments to functions

- A **argument** is any piece of data that is *passed into* a function when the function is called.
- A **parameter** is a variable that *receives* an argument that is passed into a function.

Assignment Project Exam Help

<https://powcoder.com>

- Many times we send across pieces of information (data) into a function and tasks are performed within the function.
- And many times information is passed back from a function to the main that called this function using a **return** statement .

Another way of writing the previous example : passing argument and check scope!

```
def melbourne(a):  
    print('SMES in melbourne =', a)
```

a and b are parameters

```
def canberra(b):  
    print('SMES in canberra =', b)
```

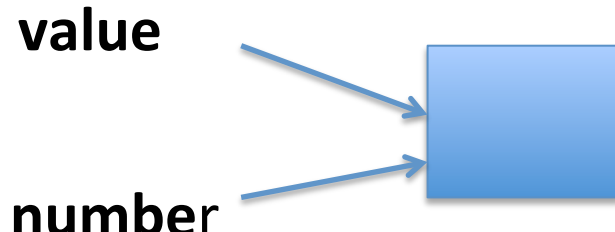
```
small_medium_melb = input("the no. in melbourne are:")  
small_medium_canb = input("the no. in canberra are:")  
melbourne(small_medium_melb)  
canberra(small_medium_canb)
```

This is an argument that is passed to the function

Example: how to pass values

```
def main():  
    value = int(input('enter a number:'))  
    show_double(value)  
  
def show_double(number):  
    result = number * 2  
    print(result)  
  
main()
```

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder



The diagram illustrates the flow of data between the `main()` function and the `show_double` function. Two blue arrows originate from the left: the top arrow starts at the label `value` and points to the top edge of a blue square box; the bottom arrow starts at the label `number` and points to the bottom edge of the same blue square box. This represents the arguments being passed to the `show_double` function.

Functions: simple examples

write a function to print the number of digits in a number

```
def print_digits(n):  
    s = str(n)  
    print(len(s) - ('.' in s))
```

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

write a function to convert from Celsius to Fahrenheit:

```
def C2F(n):  
    print (9*n/5 + 32)
```


Functions: simple examples

Print the number of digits in a number

```
def print_digits(n):  
    s = str(abs(n))  
    print (len(s) - ('.' in s))  
  
def main():  
    v = float(input('Enter a value:'))  
    print_digits(v)  
  
main()
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Convert from Celsius to Fahrenheit:

```
def C2F(n):  
    print(9*n/5 + 32)
```

Now write the main function that calls this function?

Functions: simple examples

#Convert from Celsius to Fahrenheit:

```
def C2F(n):
```

```
    print(9*n/5 + 32)
```

Assignment Project Exam Help

```
def main(): https://powcoder.com
```

```
    cels = <fill in>
```

Add WeChat powcoder

```
    C2F(cels)
```

```
<fill in>
```

Functions and return statement

A value-returning function has a **return statement** that returns a value back to the part of the program that called it.

Assignment Project Exam Help

Syntax:

```
def <function_name>():  
    statement  
    statement  
    statement  
    return <expression>
```

<https://powcoder.com>

Add WeChat powcoder

Functions and return statement

- The value of the expression that follows the key word return will be sent back to part of the program that called this function. This can be any value, expression, or variable that has a value.

Assignment Project Exam Help

<https://powcoder.com>

- A return statement can also send back:
 - Strings
 - boolean values
 - multiple values

Add WeChat powcoder

Examples 1: return statement

```
# write a program that converts  
cenlius to farenheit
```

Assignment Project Exam Help

```
def C2F(n):
```

```
    return 9*n/5 + 32
```

<https://powcoder.com>

Add WeChat powcoder

```
def main():
```

```
    cels = int(input('enter a value in celcius:'))
```

```
    f = C2F(cels)
```

```
    print(f)
```

```
main()
```

Examples 2: return statement

```
def main():  
    # get the user's age  
    first_age = int(input('enter your age:'))  
    # get the users best friends age  
    second_age = int(input('enter your best friends age:'))  
    # get the sum of both ages  
    total = sum (first_age, second_age)  
    # display the total age  
    print('their total age is:', total, 'years old')  
  
#sum function accepts two int arguments & returns sum of those  
arguments  
def sum(num1, num2):  
    result = num1 + num2  
    return result  
  
#call main function  
main()
```

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

Variables and “Scope” :

- Each function (call) defines its own local variable “scope”. Its variables are not accessible from outside the function (call) – so what happens in this example?

```
def subtract_one(k):  
    k = k - 1  
    return k  
  
i = 0  
n = subtract_one(i)  
print(i)  
print(n)  
print(k)
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Variables and “Scope” II

- Are the semantics different to the previous slide?

```
def subtract_one(i):  
    i = i - 1  
    return i
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
i = 0  
n = subtract_one(i)  
print (i)  
print (n)  
print (k)
```


Try at home!

- What is printed to the screen here?

```
>>>def bloodify1(word):  
    return word[:3] + '-bloody-' + word[3:]
```

Assignment Project Exam Help

```
>>>print(bloodify1('fantastic'))
```

<https://powcoder.com>
Add WeChat powcoder

- What is printed to the screen here?

```
def bloodify2(word):  
    return word[:3] + '-bloody-' + word[3:]
```

```
print(bloodify2('mendoza'))
```

Returning strings

```
def get_name():  
    name = input('enter your name:')  
    return name  
  
def main():  
    print('this example prints a name  
given by user:')  
    user_name = get_name()  
    print('my name is' , user_name)  
  
main()
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Returning boolean values

- You receive an integer from the user. Write a function that checks whether or not this integer is even or odd and returns a boolean.

Assignment Project Exam Help

<https://powcoder.com>

- What are the tasks in this program?
 - accept value from user
 - checks whether the value is even or odd and returns a boolean
 - print back a response

Add WeChat powcoder

Example: Returning boolean values

```
def main():  
    number = int(input('enter a number:'))  
    if XXX:  
        print('the number is even')  
    else:  
        print('the number is odd')  
  
# how do you check if a number is even or odd  
  
def is_even(num1):  
    if (XXX) == 0:  
        status = True  
    else:  
        status = False  
    return XXX  
  
main()
```

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

Returning multiple values

Syntax:

```
return expr1, expr2, expr..etc...
```

Example:

#a function to accept first and last name and then print out your full name

```
def get_name():  
    first = input('enter your first name:')  
    last = input('enter your last name:')  
    return first, last
```

```
def main():  
    first_name, last_name = get_name()  
    print('My name is:', first_name, last_name)
```

```
main()
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Returning multiple values: tuples

```
def checking_tuple(x):
```

```
    sum = x + 1
```

```
    mult = x * 3
```

```
    exp = x ** 3
```

```
    return (sum, mult, exp)
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
def main():
```

```
    num = 5
```

```
    (a, b, c) = checking_tuple(num)
```

```
    print(a,b,c)
```

```
main()
```

- Break – continue to week 10 if time does not permit

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Key arguments

While generally , arguments are passed by position to parameter variables in functions, you can also specify which parameter variable the argument should be passed to.

Assignment Project Exam Help

Syntax:

<https://powcoder.com>

`parameter_name = value`

Add WeChat powcoder

In this format, `parameter_name` is the parameter and `value` is the value being passed to that parameter. A \

An argument written in this format is called a **key argument**.

Key arguments: examples

```
def main():  
    show_interest(rate = 0.01, periods = 10,  
principal = 100000.0)  
  
def show_interest(principal, rate, periods):  
    interest = principal * rate * periods  
    print('The simple interest will be $%.2f' %  
interest)  
  
main()
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Mixing Key arguments with positional arguments : examples

#mixing keyword arguments with positional arguments. Positional arguments must come first followed by keyword arguments.

```
def main():  
    show_interest(10000.0, periods = 20, rate = 0.01)
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
def show_interest(principal, rate, periods):  
    interest = principal * rate * periods  
    print('The simple interest will be $%.2f' %  
interest)
```

```
main()
```