Chapter Overview

In this Chapter

- Functions
- Function parameters
- Arguments to a Function
- Function documentation
- Returning data from a function

Functions

- Any task which is usually repeated should be in a Function
  - MakeTea
  - Add 2 numbers
  - Multiplybyx
- Think of function like a unit of responsibility
  - Call the unit by its name

Methods Vs Function

Functions that belong to a class are called Methods

That's It.

Function Definition

```
def function_name():
    # Code block



def hello():
    print('Hi!')
```

```
def hello():
    print('Hello World!')

hello()
```

```
Hello World!
```

```
hello()


def hello():
    print('Hello World!')



Traceback (most recent call last):
  File ''hello.py'', line 1, in <module>
    hello()
NameError: name 'hello' is not defined

```

```
def hello(uname):
    print('Hello {}!'.format(uname))

hello('Pankaj')
hello('Guys')
```

```
Hello Pankaj!
Hello Guys!
```

```
def hello(uname):
    print('Hello {}!'.format(name))


hello()


File ''hello.py'', line 4, in <module>
    hello()
TypeError: hello()  missing 1 required
positional argument  'name'
```

```
def hello(name ='Anonymous'):
    print('Hello{}!'.format(name))
hello()
hello('Cedar')
```

```
Hello Anonymous!
Hello Cedar!
```

```
def hello(firstname, lastname):
    print('Hello {} {}!'.format(firstname,
    lastname))

hello('John', 'Smith')
```

```
Hello John Smith!
```

```
def hello(first, last):
    print('Hello {} {}!'.format(first,
    last))


hello(first = 'John', last= 'Smith')
hello(last = 'Doe', first= 'John')

>
```

```
def hello(first, last='Smith'):
    print('Hello {} {}!'.format(first,
    last))

hello('John')
hello('John', 'Smith')
```

```
def hello(first, last='Doe'):
    """Say hello."""
    print('Hi {} {}!'.format(first, last))


help(hello)

Help on function hello in module __main__:

hello(first, last='Doe')
    Say hello.
```

```
def odd or even(numb er):
    ''''D'e'termine if a number is odd or even.''''''
    if number % 2 == 0:
        return 'Even'
    else:
        return 'Odd'

odd or even string = odd or even(7)
print(odd or even string)
```

```
Odd
```

```
def is odd(number):
    ''''D'e'termine if a number is odd.'""
    if number % 2 == 0:
        return False
    else:
        return True


print(is odd(7))
```

```
True
```

```
def get_name():
    name = input('What is your name? ')
    return name


def say_name(name):
    print('Your name is {}.'.format(name))

def get_and_say_name():
    ''''G'e't and display name''''''
    name = get_name()
    say_name(name)


get_and_say_name()
```

What is your name? Tom

Your name is Tom.

Section Summary

Summary

- A function is a block of reusable code that performs an action and can optionally return data.
- A function must be defined before it is called.

Summary

- **The basic syntax for defining a function is:**

```
def function_name(parameter_name):
```

- **A function can accept parameters. To make a parameter optional supply a default value for that parameter.**

Summary

- You can supply a docstring as the first line of your function.
- The return statement exits the function and passes back what follows return.
- Use the built-in help() function to get help with an object. When supplying a function to help() the docstring contained within the function is displayed.