

Reading and Research - Functions

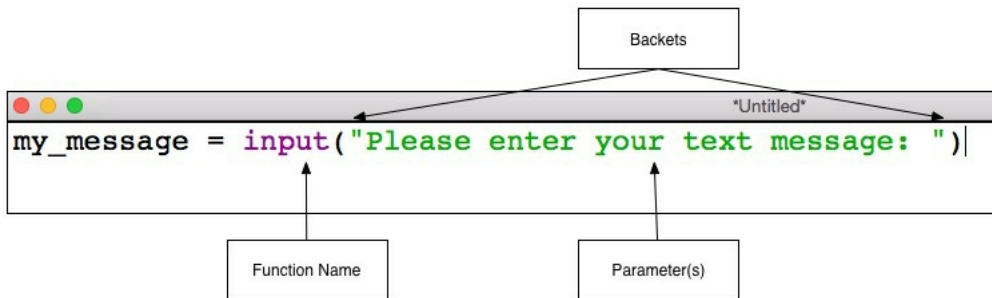
These tasks are designed to introduce you to the programming topic we will be studying in class next lesson. You **must** complete these activities prior to the lesson.

Functions

A function is a way of segmenting code into manageable pieces, so that it can be used again and again. Creating functions cuts the size of scripts and makes them easier to understand.

You have been using functions since you started learning Python. Two of the most common are `input()` and `print()`.

When a function is used in the main body of your program it is called a function call. You can recognise a function call as it has the following features:



Parameters

Most functions work on some data that is provided by the main program. The data is passed from the main program to the function by putting it between the brackets of the function call. Multiple pieces of data are separated with commas. These pieces of data are the **parameters** of the function.

A function can have **zero or more parameters**. If it has zero parameters, the brackets are empty.

Built-in functions

The Python interpreter has a number of functions and types built into it that are always available. You have already met a number of these.

Task 1

Use the Python shell to investigate the built-in functions below and complete the table.

Code	Result	Description of function
<code>round(3.87)</code>	4	Rounds float up.
<code>round(3.5)</code>	4	Rounds float up.
<code>round(4.5)</code>	4	Rounds float down.
<code>round(5.78,1)</code>	5.8	Rounds up to one decimal place.
<code>str(7)</code>	'7'	Prints 7 as a string.
<code>str(7.65)</code>	'7.65'	Prints 7.65 as a string.
<code>len("Computing")</code>	9	Prints the length of the string.
<code>type(7)</code>	class int	Defines the input as an integer.
<code>type("7")</code>	class str	Defines the input as a string.
<code>chr(97)</code>	"a"	Converts the ASCII code into its equivalent character.
<code>ord("a")</code>	97	Converts the character into its equivalent ASCII code.

String Functions

Python has a number of built-in functions that allow a programmer to manipulate text strings.

Task 2

Find out what each of the string functions in the table below does to a text string variable with the identifier `msg`. You will need to assign `msg` a value that is a string with a mix of upper and lower case letters and multiple words and then explore the effects of each of the functions.

Function	Effect it has on text string
<code>msg.upper()</code>	Capitalized all characters.
<code>msg.lower()</code>	Prints all character in lower case.
<code>msg.capitalize()</code>	Capitalizes the first letter of the first word.
<code>msg.title()</code>	Capitalizes the first letter of every word.

Task 3

1. Write a program that reads in a string and displays the number of characters in the string.
2. Write a program that displays the ASCII code for any given character.
3. Write a program that will display the character for any given ASCII code.

Paste your code and screenshots of running your programs in the space below

```
#Jordan Russell
#8/11/14
#Functions R+R Task 3 Exercise 1
```

```
message = str(input("Please input a message: "))
print("There are {0} characters in your message.".format(len(message)))
```

```
'''
Please input a message: Hello world
There are 11 characters in your message.
>>>
```

```
#Jordan Russell
#8/11/14
#Functions R+R Task 3 Exercise 2
```

```
ascii_code = ord(str(input("Please input a character to be converted into ASCII: ")))
print("The code for your given ASCII character is {0}.".format(ascii_code))
```

```
>>>
Please input a character to be converted into ASCII: ;
The code for your given ASCII character is 59.
>>>
```

```
#Jordan Russell
#8/11/14
#Functions R+R Task 3 Exercise 3
```

```
ascii_character = chr(int(input("Please input an ASCII code: ")))
print("The character for your given ASCII code is {0}.".format(ascii_character))
```

```
>>>
Please input an ASCII code: 35
The character for your given ASCII code is #.
>>>
```

Importing Modules

A module is a group of functions that programmers have packaged together for people to use. To use a module in your code you must import it into your program.

You do this by adding a line at the top of your program e.g.

```
import math
```

When calling a function that is part of a module you need to prefix the function name with the module identifier, using dot notation e.g.

```
math.trunc(my_float)
```

Task 4

1. Use the [Python Documentation](#) to find out about the functions in the table below

Math Function	Description
<code>math.trunc(3.14159)</code>	Returns the real value of x (in this case 3.14159) truncated to an integer.
<code>math.pi</code>	Outputs the mathematical constant of pi to available precision.
<code>round(math.pi, 3)</code>	Rounds pi to three decimal places.

1. Using the Python documentation, identify two other useful functions in the math module, briefly describe what each does and give a brief example of the purpose of a program that might use this function.

Math Function	Description	Program that would use it
<code>math.cos()</code>	Returns the cosine of x radians.	A program used to calculate the length of sides or angles on a triangle using trigonometry.
<code>math.degrees()</code>	Converts angle x from radians to degrees.	A program calculating and converting the sizes of angles.

Using Built-in Functions and Modules

Let's consider some code that uses some of the built-in functions that you have been investigating.

Task 5

Without entering the code into Python, try to explain what the code below does. You will need to explain each of the different

sections of the program by **adding comments** to the program.

```
import math

number_to_change = float(input("Please enter a number to 3 or more decimal places: ")) # User inputs a number to 3 or more decimal pla
number_one = math.trunc(number_to_change) #Rounds the input number down to an integer e.g 5.
number_two = round(number_to_change,2) #Rounds the input number to two decimal places, e.g .36.

print("The integer part of your number is {0}.".format(number_one)) #Prints the integer e.g 5
print("The number to two decimal places is {0}.".format(number_two)) #Prints the integer and then the two decimal places, e.g 5.36.
```

Structured programming

Read **pages 84 and 85** of AS Computing by Bond and Langfield which describe structure tables and hierarchy charts.

Task 6

Complete the following statements about structure tables

1. A structure table shows how a problem can be ... broken down into sub-problems and expressed as a numbered list in structured English.
2. The steps involved in solving a problem are written as ... numbered lists of sub-problems written in structured English displayed using indentation,

Consider this problem:

Roasting Time For Chickens

A program is needed that works out the cooking time for roasting a chicken. The time required is 30 minutes for every 500g plus 30 minutes. If the chicken is stuffed, it needs an extra 30 minutes. The user enters the weight of the chicken in grams and whether it is stuffed, and the program calculates the approximate time it should be roasted for in hours and minutes.

The problem above can be divided into simpler sub-problems that make writing the program easier:

- Get weight of chicken from user.
- Get whether chicken is stuffed or not from user.
- Calculate total roasting time.
- Display roasting time.

The third sub-problem, calculating the total roasting time, involves some sub-problems itself:

- Calculate roasting time for weight in minutes.
- Add additional 30 minutes required.
- Add time if chicken is stuffed (further 30 minutes).
- Convert minutes cooking time to hours and minutes.

Task 7

Produce a structure table for the **Roasting time for chickens** problem.

0 CALCULATE roasting time

0.1 INPUT weight of chicken

0.2 INPUT whether or not chicken is stuffed

0.3 CALCULATE roasting time

0.3.1 CALCULATE roasting time for weight in minutes.

0.3.2 ADD additional 30 minutes

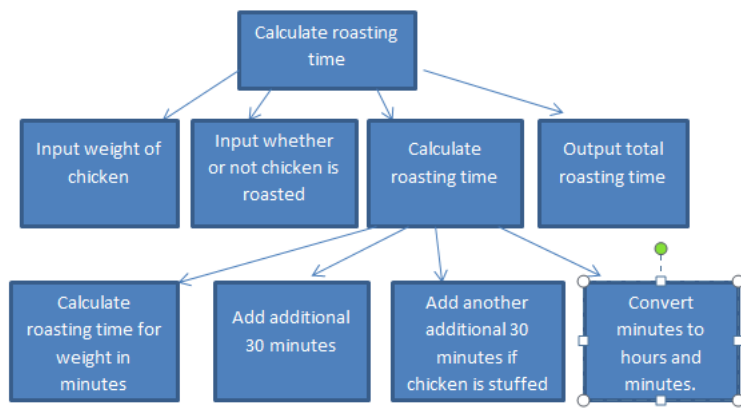
0.3.3 ADD 30 minutes if chicken is stuffed

0.3.4 CONVERT minutes to hours and minutes

0.4 OUTPUT total roasting time

Task 8

Draw a hierarchy chart for the 'Roasting time for chickens' problem.



Summary

In this R&R you have investigated built-in functions and importing modules. You have seen the basic structure of a function call and how parameters can be passed to a function. You have also seen how to import modules so that their functions are available in your program. You have investigated functions in the random module and the math module.

You have also explored structure tables and hierarchy charts. These are approaches to program design that help in the breakdown of more complex problems into sub-problems. This makes it easier to develop programs to address more complex problems.

Please make sure you have completed this R&R fully before your next programming lesson as it will form the basis of the initial classroom discussion and starter tasks.