Skip to main content

Side panel

UWA-CSSE

-
- 0

  ### Notifications

  You have no notifications
  See all

- Zehua Zhu
  Dashboard

  Profile Grades Messages Preferences

  Log out

- CITS1401 2022S2
- Participants
- Grades
- General
- Labs

- Home
- Dashboard
- Calendar
- Private files
- My courses
- CITS1401 2022S2

CITS1401 Computational Thinking with Python (2022S2)

1. Home
2. My courses
3. CITS1401 2022S2
4. Labs
5. Lab 02. Functions in Python

Information

Flag question

Information text

Functions

The process of decomposing a larger problem into a set of smaller ones, each of which may be decomposed into even smaller ones, and so on, is a key part of human problem solving. It's at the heart of computational thinking (programming) too, which after all is just a form of problem solving. In programming, the solutions for solving the subproblems are called *functions.* (Revise the function usage and defining functions in Lectures).

To learn a bit about how functions work in Python, type each of the inputs on the left hand side of the table below into the Python shell, observing what happens and reading the notes. Note that some of the table cells in the left column involve typing multiple lines.

Input Line	**Notes**

`def square(num):`
`    """Returns a given number squared"""`
`    return num * num`
	After you enter the first line, Python will respond with cursor blinking at next line at a tab space indentation. Subsequent lines within the function (called the body of the function) must be indented, usually with the Tab key. The purpose of the prompt is to indicate that Python knows you haven't finished and it's expecting more input, which in this case will be an indented block. After typing the third line you'll hit Enter key twice to complete the entry of this def statement.
	The first line defines the function's name ("square" in this example) and any pieces of data the function requires to operate which are called *parameters* (in this example, there is one parameter called `num`). The names of functions should obey similar naming conventions as variable names so they can be easily read.
	The second line is a docstring, a line of text that describes the point of the function that starts and ends with """ (three sets of double quotes). This is important part of the good style of coding. The details are discussed in the lectures.
	The third line says that the function *returns* the value of the parameter multiplied by itself, i.e. "squared". This is the output of the function.
	This block of code defines the function but does not

execute (run) it. We will execute the function by calling it by its name as discussed below.

```
square(3)
```

Square is a function (defined above) that for any given num, returns the value num^2. num is called a parameter to the function. The value given to the parameter, 3 in this case, is called the *argument*.

```
square(3.0)
```

The type returned by square is the same as the type of the argument: float in this case, integer (or just int) in the previous case.

```
def gonk(x):
    """A mystery function"""
    return x * x
```

What do you think this does?

```
gonk(5)
```

The functions square and gonk are identical in effect, but the choice of name makes the second almost meaningless to the reader.

```
def to_celsius(temp_f):
    """Converts a given fahrenheit temperature to celsius"""
    return (temp_f - 32) * 5 / 9
```

You should be able to figure out what this function will do ...

```
to_celsius(212)
```

... and predict the values of these two expressions ...

```
to_celsius(32)
```

... before you even type them.

```
def celsius(temp_f):
    """Converts a given fahrenheit temperature to celsius"""
    freezing = 32
    factor = 5 / 9
    ans = (temp_f - freezing) * factor
    return ans
```

Here's a longer version of an equivalent function using local variables (variables belonging just to the function) and a multiline function body. Note that all lines of the function body must be indented by the same amount.

```
celsius(212)
celsius(32)
```

Just to prove this one

	behaves like the last one.
`abs(-9)`	Python has lots of built-in functions. Can you work out what this one does?

`round(3.8)`
`round(3.5)`
`round(3.3)`
	And this one?

`int(3.8)`
`int(3.5)`
`int(3.3)`
	And this?

Question 1

Not complete
Marked out of 3.00
Flag question

Question text

Fill in the blanks in the following.

The Python code below defines a function named Answer distance_travelled that has two parameters, the first being Answer duration and the second being Answer speed. If the function is "called" via the expression `distance_travelled(10, 50)`, the value of the arguments are Answer 10 and Answer 50 respectively, and the numeric value returned by the function will be Answer 500.

```
def distance_travelled(duration, speed):
    """Returns the distance travelled given duration and the speed"""
    return speed * duration
```

Check

Question 2

Not complete
Marked out of 1.00
Flag question

Question text

Functions should also have clear and concise naming, so that users (such as yourself and your colleagues) will be able to understand what it does without having to go over the function in detail. Of course, they should also be syntactically correct. Select **ALL** functions names that are syntactically correct AND are of good style (i.e. readability).

Select one or more:
☑

find_largest_value()

□

function()

□

a_t_z_cal()

☑
evaluate_triangle_area()
□

convert_currency[]

☑

remove_white_spaces()

□

find minimum()

Check
Next page

Network connection lost. (Autosave failed).

Make a note of any responses entered on this page in the last few minutes, then try to re-connect.

Once connection has been re-established, your responses should be saved and this message will disappear.

◄ Lab 01. Python basics (for late submission or approved special considerations)
Jump to... Jump to...
Lab 02. Functions in Python (for late submission or approved special considerations) ►
Skip <span id="mod_quiz_navblock_title">Quiz navigation</span>

Quiz navigation

Information i This page Question 1 This page Question 2 This page Question 3 Question 4 Question 5
Question 6 Question 7 Question 8 Question 9 Question 10
Finish attempt ...
Time left
You are logged in as Zehua Zhu (Log out)
CITS1401 2022S2