Functions Part 1

can code communities

Functions

A function is a block of code which runs when it is called.

We pass parameters into functions.

Functions can perform actions on objects, return an object, and/or print.

Just like with conditionals and loops, everything inside the body of a function must be indented.

Calling a Function

A function is not active until it is called.

We call a function by entering its name with arguments included in parentheses.

When a function is called, the argument passed in, it is assigned to the function parameter which is used within the function for processing.

You must provide the same number of arguments as parameters listed in the function call.

Writing a Function

When we write our own functions there are a few key things that we need to think about:

- What to name the function?
- What do we need to get from the user? (Parameters/Arguments)
- What do we return?
- What do we want the function to do?

Exercise

Write a function that will loop through a string and print whether a character is or is not a vowel.

Function Purposes

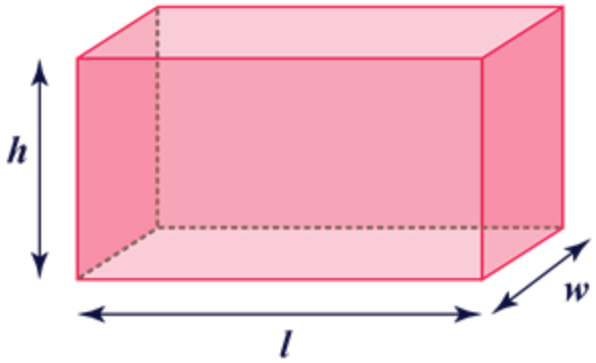
There are 3 main types of functions:

1. Modify one or more of its parameters
2. Return something
3. Print something

Most functions will only do **one** of these things. Sometimes functions can do more than one (for example, modify a list and return a value that indicates whether it was successful or not).

When you're writing your function, think about which of these categories it falls into.

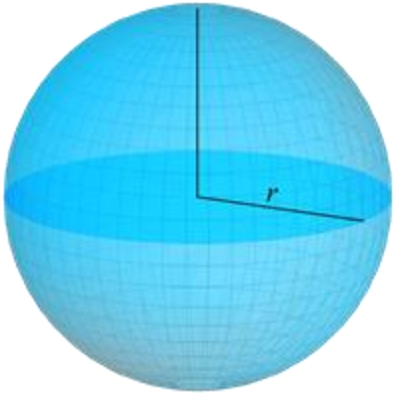
Example



Write a function that returns the surface area of a box (rectangular prism).

Surface Area = width*2 + length*2 + height*2

Exercise



Write a function that returns the surface area of a sphere.

Surface Area = 4 * pi * radius^2

Return?

You can have a function that returns something, or a function that returns nothing.

The return statement goes at the end of a function.

If a function returns something, that is the result of the function call. You can set it to a variable. A function can return any data type.

If a function returns nothing, there is no return statement, but you can think of it like a return statement at the end saying `return None`, or a return statement at the end saying `return`

If you try to use the return value of a function that returns nothing, this will cause issues.

Exercise

What is the output of this code?

```
def get_vowels(word):
    out = ""
    for letter in word:
        if letter in "aeiou":
            out += letter
    return out

my_word = "strawberries"

get_vowels(my_word)
print(get_vowels(my_word))
```

What about this code?

```
def print_vowels(word):
    out = ""
    for letter in word:
        if letter in "aeiou":
            out += letter
    print(out)

my_word = "strawberries"

print_vowels(my_word)
print(print_vowels(my_word))
```

Are they the same or different? Why?

Exercise

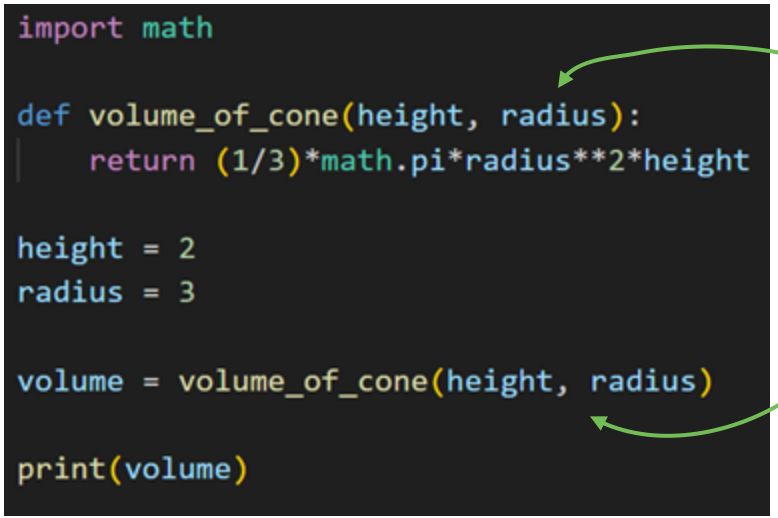
Write a function that takes a list and a value, and removes the value until it no longer exists in the list.

Return how many times the value was removed.

Parameters vs. Arguments

Parameters are in the function definition. **Arguments** are in your function call. When you call the function, the parameters get assigned to the values of the arguments.

However, the terms can often be used interchangeably.



In this example, the parameters and arguments have the same names, but they can have different names. This is why order is important.

Exercise

Suppose you work for a bank, and you have a list of transactions with the following information for each one: customer ID, transaction amount, and transaction type (deposit or withdrawal).

Write a function that takes in the list of customer transactions and returns a dictionary where the keys are the customer IDs and the values are the total transaction amounts for each customer.

Example:

```
transactions = [{'id': 'a', 'amount': 500, 'type': 'deposit'},\
                {'id': 'b', 'amount': 350, 'type': 'deposit'},\
                {'id': 'a', 'amount': 450, 'type': 'withdrawal'}]
```

Output: `{'a': 50, 'b': 350}`

Discussion: Why are functions useful?