

1. When to write a function

Functions are usually written to help better organize code, or to help reduce the amount of code that is being written. If a program contains multiple sections of code that look identical, or nearly identical, that program should probably be rewritten with functions to take the place of those repeated sections.

2. What to think about when writing a function

The two important things to consider when writing a function are: what information do I need to send *into* this function (as parameters) for it to be able to work? and what information does this function need to send back (*return*) to the main program?

3. Basic function definition

```
def <function_name>(<optional_parameters>):  
    <statement>  
    <statement>  
    <statement>  
    <optional return> <optional_values>
```

2. Basic function used to perform simple instructions

No input via *parameters* to the function is needed, and no *return* value from the function is expected.

```
def print_instructions():  
    print("This program will think of a number ")  
    print("between 1 and 10, and you'll have ")  
    print("three tries to guess it. Good luck! ")
```

3. Function that uses input via *parameters*

Function requires information from the caller to be able to work correctly. No information needs to be sent back to the caller, however.

Values from the main program are used in the function call, and substituted in for the parameters in the function definition.

```
def old_macdonald(animal, noise):  
    print("Old MacDonald had a farm, ee-aye-ee-aye-oh,")  
    print("And on that farm he had some " + animal + "s, ee-aye-ee-aye-oh.")  
    print("With a " + noise + ", " + noise + " here")  
    print("and a " + noise + ", " + noise + " there")  
    print("Here a " + noise + ", there a " + noise)  
    print("Everywhere a " + noise + ", " + noise)  
    print("Old MacDonald had a farm, ee-aye-ee-aye-oh,")
```

These lines in the main program, then, would be able to call the function after it had been defined.

```
def main():  
    old_macdonald("pig", "oink")  
    old_macdonald("geese", "honk")  
    old_macdonald("dog", "woof")  
    old_macdonald("horse", "neigh")  
    old_macdonald("sheep", "baa")
```

4. **Function that takes input via parameters and returns values to another part of the program**

```
def calculate_area(length, width):  
    """This function calculates the area of a rectangle."""  
    area = length * width  
    return area
```

These lines in the main program, then, would be able to call the function after it had been defined.

```
def main():  
    rect_length = 5.6  
    rect_width = 4  
    rect_area = calculate_area(rect_length, rect_width)  
    print("Area of the rectangle is", rect_area)
```

5. **Immutable parameters (numbers, strings) will not have their global values altered by a function.**

```
def double_my_money(balance):  
    """The variable balance stores the amount of money in my account, and  
    this function shows you that it won't work."""  
    balance = balance * 2  
    return # optional
```

These lines in the main program, show how this doesn't work.

```
def main():  
    balance = 100  
    double_my_money(balance)  
    print("I have", balance, "dollars") # prints "I have 100 dollars"
```

If you want to write the `double_my_money` function so that it does affect the outside program, you should write it so that it returns a value that the main program can then use.

```
def double_my_money(balance):  
    """This function shows you how to get it to work."""  
    new_balance = balance * 2  
    return new_balance
```

And the main program would include this:

```
def main():  
    balance = 100  
    balance = double_my_money(balance) # gets the new value and saves it
```

6. **Mutable parameters (lists, objects) may have their values changed by a function.**

```
def remove_odd_numbers_from_list(num_list):  
    for num in num_list:  
        if num % 2 == 1:  
            num_list.remove(num)
```

And an example of what the function call might look like in the main program:

```
def main():  
    my_nums = [0,1,2,3,4,5,6,7,8,9]  
    remove_odd_numbers_from_list(my_nums)  
    print(my_nums) # now has the values [0,2,4,6,8]  
                  # The list has been mutated
```