

Functions in Python

This notebook will help you learn how to use **functions** in Python.

What is a Function?

- A function is a block of code that runs only when you call it.
- Functions help us **reuse code** instead of writing the same thing many times.

Understanding a Python Function (Step by Step)

A **function** is like a small machine that does a job for us.

We give it a name, sometimes give it inputs, and it can give us back an answer.

Example: A Very Simple Function

```
def say_hello():  
    print("Hello!")  
  
# Call the function  
say_hello()
```

Hello!

def

- This is a keyword in Python.
- It tells Python: "Hey, I'm about to **define** a function."

say_hello

- This is the **name** of the function.
- We choose the name (it can be anything, but it should describe the job).
- Here, the job is to "say hello," so the name is `say_hello`.

()

- Functions can take **inputs** (also called parameters).
- In this example, the parentheses are empty, meaning the function does **not** need any input.
- Later, we will see functions that have something **inside** these parentheses.

:

- The colon tells Python:
"Everything that comes after this (and is indented) belongs to the function."

Indentation (TAB or 4 spaces)

- In Python, indentation (the space before a line) is **VERY important**.
- It tells Python that this code is *inside* the function.
- Without indentation, Python will give an error.

```
# Wrong
def say_hello_again():
print("Hello!")    # Not indented error <--
```

```
File "/var/folders/5n/lp3_lp252c17c644s2777gmr0000gn/T/ipykernel_81997/2417206209.py", line 3
```

```
    print("Hello!")    # Not indented error <--
    ^
```

```
IndentationError: expected an indented block after function definition on line 2
```

```
# Correct
def say_hello_again():
    print("Hello!") # Indented
```

The Body (Code Block)

- The indented area **is the body of the function**.
- The body is where the actual work happens.
- Think of it as the inside of the machine.

Here's the function again, with the **body highlighted**:

```
def say_hello():          # function header
    print("Hello!")       # <-- THIS PART is the body (code block)
```

Functions with Parameters

Functions can take information (parameters) and use it.

```
def greet(name):  
    print("Hello", name)
```

```
# Examples  
greet("Ana")  
greet("Luis")
```

```
Hello Ana  
Hello Luis
```

Functions with Math

```
def add_numbers(a, b):  
    result = a + b  
    print("The result is:", result)
```

```
# Examples  
add_numbers(3, 4)  
add_numbers(10, 20)
```

```
The result is: 7  
The result is: 30
```

Returning Values

Sometimes we want a function to **give back** an answer.

```
def multiply(a, b):  
    return a * b
```

```
# Example  
answer = multiply(6, 7)  
print("6 times 7 is", answer)
```

```
6 times 7 is 42
```

Example: Why Use Functions?

Let's imagine we want to calculate the **monthly payment** for a loan.

The formula (don't worry too much about it right now) is:

$$\text{payment} = P \times \frac{i(1+i)^n}{(1+i)^n - 1}$$

- (P) = principal (the amount of the loan)
- (i) = monthly interest rate (for example, 5% yearly → 0.05/12 per month)
- (n) = number of months (loan length)

Without a Function

If we don't use a function, we have to write the full formula every time.

```
# Example: $10,000 loan, 5% yearly interest, 12 months
P = 10000
i = 0.05 / 12
n = 12

payment1 = P * (i * (1 + i)**n) / ((1 + i)**n - 1)
print("Monthly payment 1:", payment1)

# Another loan, different values
P = 5000
i = 0.07 / 12
n = 24

payment2 = P * (i * (1 + i)**n) / ((1 + i)**n - 1)
print("Monthly payment 2:", payment2)
```

```
Monthly payment 1: 856.0748178846745
Monthly payment 2: 223.86289551572642
```

With a Function

If we put the formula into a function, we only need to **define it once**.

```
def calculate_monthly_payment(P, i, n):
    return P * (i * (1 + i)**n) / ((1 + i)**n - 1)

# Now we can reuse it easily:
print("Monthly payment 1:", calculate_monthly_payment(10000, 0.05/12, 12))
print("Monthly payment 2:", calculate_monthly_payment(5000, 0.07/12, 24))
# We can also be more explicit by naming each parameter:
print("Monthly payment 3:", calculate_monthly_payment(P=15000, i=0.04/12, n=36))
```

```
Monthly payment 1: 856.0748178846745
Monthly payment 2: 223.86289551572642
Monthly payment 3: 442.85977510264564
```

A Note on Naming

In the 3rd example we wrote:

```
calculate_monthly_payment(P=15000, i=0.04/12, n=36)
```

The names P, i, and n only make sense if you already know the formula.

For a beginner (or someone reading your code later), P could mean anything, and i looks like just a letter, not "interest rate."

This is why choosing clear names is so important in functions.

A better version would be:

```
def get_monthly_payment(principal, monthly_interest, months):  
    return (principal *  
            (monthly_interest * (1 + monthly_interest)**months)  
            /  
            ((1 + monthly_interest)**months - 1))  
  
print(get_monthly_payment(principal=15000, monthly_interest=0.04/12, months=36))  
#Same as :  
print(get_monthly_payment(15000, 0.04/12, 36))
```

442.85977510264564

442.85977510264564

Why This is Better

- **Less typing:** we don't repeat the formula every time we need a monthly payment.
- **Fewer mistakes:** the formula only lives in one place, inside the function. If we fix it once, it works everywhere.
- **Reusability:** we can call the function for any loan, with any numbers.
- **Clear names:** when we choose clear names for the inputs, it helps us (and anyone reading the code) understand what the function needs without having to look back at the formula.

Practice Exercises

1. Make a function that subtracts two numbers and prints the result.
2. Make a function that returns the area of a rectangle (base × height).
3. Make a function that asks for a number and prints if it is even or odd.
4. Make a function that converts Celsius to Fahrenheit.

Note: I am only asking to the function, and I'm also saying what it should do. All functions SHOULD BE TESTED

Problem 1

Problem 2

$$A = \text{base} * \text{height}$$

Problem 3

Problem 4

Formulas:

$$F = \frac{9}{5}C + 32$$

$$C = \frac{5}{9}(F - 32)$$