



**Python Software Developer**  
**Task: Recursion**

[www.hyperiondev.com](http://www.hyperiondev.com)



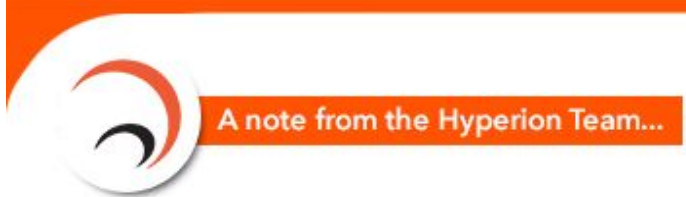
# Introduction

**Welcome to the Recursion Task!**

**Overview:**

Welcome to the next task and congratulations for making it this far. Now that you are comfortable with Python, we can start working on some applications and interesting fields. We will now dwell into one of the most interesting and important concepts in programming; Recursion and its applications.

-The Hyperion Team



## **An introduction to Recursion**

Recursion is the process of solving a problem by breaking it up into smaller sub-problems, and solving these. This may sound confusing at first, but is an important technique and there are some problems (listed later as additional exercises) which can only be solved by thinking recursively! Often when you are faced with a large task, you accomplish it by breaking the task into smaller, more manageable components and then completing these in order to complete the overall task. Recursion applies this idea to programming .

## **Practical Applications of Recursion**

Often, recursion is studied at an advanced computer science level. Recursion is usually used to solve complex problems that can be broken down into smaller, identical problems. Recursion isn't required to solve a problem. Problems that can be solved with recursion, most likely can be solved with loops. Also, a loop may be more efficient than a recursive function. Recursive functions require more memory, and resources than loops, making them less efficient in a lot of cases. This usage requirement is sometimes referred to as overhead. You might be thinking, "Well why bother with recursion. I'll just use a loop. I already know how to loop and this is a lot more work." This thought is understandable, but not entirely ideal. When solving complex problems, a recursive function is sometimes easier, faster, and simpler to build and code.

Think of these two "rules":

- If I can solve the problem now, without recursion, the function simply returns a value.
- If I cannot solve the problem now without recursion, the function reduces the problem to something smaller and similar, and calls itself to solve the problem.

Let's apply this using a common mathematical concept, factorials. If you are unfamiliar with factorials in mathematics, please refer to the following reading.

## **Factorials**

The factorial of a number  $n$ , is denoted as  $n!$ .

Here are some basic rules of factorials:

If  $n = 0$  then  $n! = 1$

If  $n > 0$  then  $n! = 1 \times 2 \times 3 \times \dots \times n$

For example, let's look at the factorial of the number 9:

$$9! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9$$



## Instructions

First read **example.py**, open it using Notepad++ (Right click the file and select 'Edit with Notepad++').

- example.py should help you understand some more Python. Every task will have example code to help you get started. Make sure you read all of example.py and try your best to understand.
- You may run example.py to see the output. The instructions on how to do this are inside the file. Feel free to write and run your own example code before doing this task to become more comfortable with Python.
- You are not required to read the entirety of Additional Reading.pdf, it is purely for extra reference.

**Note:** You have reached a milestone in your Python learning, as you're now beginning to create some genuinely useful programs!

# Compulsory Task 1

**Follow these steps:**

- Write all of the following recursive functions in a file called “recursion.py”:
  - Reverse a string
  - Find the factorial of an integer
  - Calculate the nth Fibonacci number (Fibonacci numbers are a sequence where each number is the sum of the previous two - 0 1 1 2 3 5 8...)

# Optional Task 1

**Follow these steps:**

- Print out the first n Fibonacci numbers (using the previous Fibonacci function is useful)
- Implement a search/replace function recursively:

Sample I/O:

Enter String:

Hello world

Enter word to find:

Llo

Enter word to replace:

@@

he@@ world

### Things to look out for:

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Python** or **Notepad++** may not be installed correctly.
2. If you are not using Windows, please ask your tutor for alternative instructions.

### Still need help?

Just write your queries in your comments.txt file and your tutor will respond. Alternatively you can email us on [help@hyperiondev.com](mailto:help@hyperiondev.com).

## Task Statistics

Last update to task: 20/05/2016

Author: Riaz Moola

Main tutor: Sobane Motlomelo

Task Feedback link: [Hyperion Development Feedback](#).