

# Python Prime Number Finder

Time required: 30 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

## Pseudocode

1. Write pseudocode for the exercise
2. Submit with the assignment

## Prime Numbers

A prime number (or a prime) is a whole number greater than 1 that is divisible only by 1 and itself.

Prime numbers are the numbers that have only two factors, that are, 1 and the number itself. Consider an example of number 5, which has only two factors 1 and 5. This means it is a prime number. Let us take another example of the number 6, which has more than two factors, i.e., 1, 2, 3, and 6. This means 6 is not a prime number. Now, if we take the example of the number 1, we know that it has only one factor. So, it cannot be a prime number as a prime number should have exactly two factors. This means 1 is neither a prime nor a composite number, it is a unique number.

## How It Works

The idea to solve this problem is to iterate the val from start to end using a for loop and for every number, if it is greater than 1, check if it divides n. If we find any other number which divides, print that value.

## Tutorial 1 - Prime Number Finder

This is a hard coded program that finds the prime numbers or factors from 2 to 9.

```

1  """
2      Name: prime_number_finder.py
3      Author:
4      Created:
5      Purpose: Find prime numbers in a range
6      A number is prime if it is only divisible by itself and 1
7  """
8
9  # Hard coded range of numbers to check for prime numbers, 2-9
10 for n in range(2, 10):
11     is_prime = True
12     # For each number to test for prime,
13     # test all numbers from 2 to n
14     for x in range(2, n):
15
16         # If n is divisible by x, it means it's not a prime number
17         if n % x == 0:
18
19             # Not a prime number
20             # Integer division to show the factors
21             print(f"{n} equals {x} * {n//x}")
22             is_prime = False
23             break
24
25     # If the for loop runs to completion for a number
26     # n is not a prime number, go to the next number in the range
27     if is_prime == True:
28         # If n was not divisible by any x, it is a prime number
29         print(f"{n} is a prime number")

```

Example run:

```

2 is a prime number
3 is a prime number
4 equals 2 * 2
5 is a prime number
6 equals 2 * 3
7 is a prime number
8 equals 2 * 4
9 equals 3 * 3

```

## Requirements

Ask the user to enter a range of numbers to check for prime numbers. Calculate and display the prime numbers.

1. Add a creative program title
2. Get range input from user

3. Change the script style code into functions as follows
  - a. main()
  - b. print\_title()
  - c. get\_input()
  - d. calculate\_prime()
4. Calculate and print the prime numbers and factors
5. Display the results
6. Add a menu loop that allows the user to run the program again

Example run:

```
+-----+
|           Python Prime Number Finder           |
|       Find the prime numbers within a range       |
+-----+
First number: 60
Last number: 65
60 equals 2 * 30
61 is a prime number
62 equals 2 * 31
63 equals 3 * 21
64 equals 2 * 32
65 equals 5 * 13
Run again? (Y/N) y
Enter a range of numbers to test for prime
First number: 100
Last number: 105
100 equals 2 * 50
101 is a prime number
102 equals 2 * 51
103 is a prime number
104 equals 2 * 52
105 equals 3 * 35
Run again? (Y/N) n
Bye ;')
```

---

## Assignment Submission

1. Attach the pseudocode.
2. Attach the program files.
3. Attach screenshots showing the successful operation of the program.
4. Submit in Blackboard.