Prime Number Checker in Python

Author: Pratyush Saha
Date: 14.07.25
Class: 9

Project Type: Python code to check wether a given number is a prime no. or not

Virtually Published on Github

Index

Table of Contents

Introduction

I would like to start the paper with an introduction to my project. This project is about checking wether the given number(or any number in a variable), is a prime number or not.
Prime numbers are a very fundamental part of mathematics. These are very important in studying about number theory, mathematics algorithms or any kind of problems.

This paper presents yet efficient way of function for finding out a prime number. Instead of trial calculations, we can directly divide the given number by factors ranging from 2 to √n. We will only use the whole number of √n, like int(√n).
We don’t have to try dividing the number by so many guessed numbers. In this program, my main aim is to check wether the given number is prime or not using the same algorithm. We will use math module to accomplish the code.

Logic

```
1. When the user puts a number < 1:
        it will give a false output of not possible
2. If the number is not 1 or n<1:
        it will check the number by finding out the factors upto √n by using for loop
        and
         during the looping iteration
        if the factor is completely divisible by one of the factors from 2 to √n;
                then the loop will break and say its a composite number.
```

Code and Explanation

This is the code for the logic we’ve discussed already.

```
import math
#code to check wether the number given is a prime
number or not
#using the trial method
#aim: experiment with math module

def primecheck(number):
if number<=1:
print("Its not possible!")
return False

else:
set_prime = True
number_range = int(math.sqrt(number))
for s in range(2, number_range+1):
if number % s ==0:
print("its not prime")
break
print("It is a prime number")
primecheck(int(input("Enter a number: "))) #you can
change the values
#here i have given input
```

In line 1., I have imported the math module. We are going to use the sqrt() funtion in our code

In this line `def primecheck(number):`, we defined a function primecheck with a parameter “number”. We would also been have directly used without function. Defining a function makes this code cleaner.

```
if number<=1:
return False
```

Here, we created a conditon so that any number which is = or < 1 then its not valid

Then we passed this to else.

```
else:
set_prime = True
```

We set set_prime variable to true unless and untill something happens in the loop

Now the main action scene comes.
According to our logic, any no. From 2 to √n, if becomes divisible by the number n then the number will be composite and the vice versa will be the prime. NOTE: √

```
number_range = int(math.sqrt(number))
print("It is a prime number")
for s in range(2, number_range+1):
if number % s ==0:
print("its not prime")
break
```

We assigned a variable number_range where we are storing the whole number part of√n as we are not going to take the decimal of the rooted number.

```
for s in range(2, number_range+1):
```

Then we looped from 2 to √number as we've already discussed the logic.

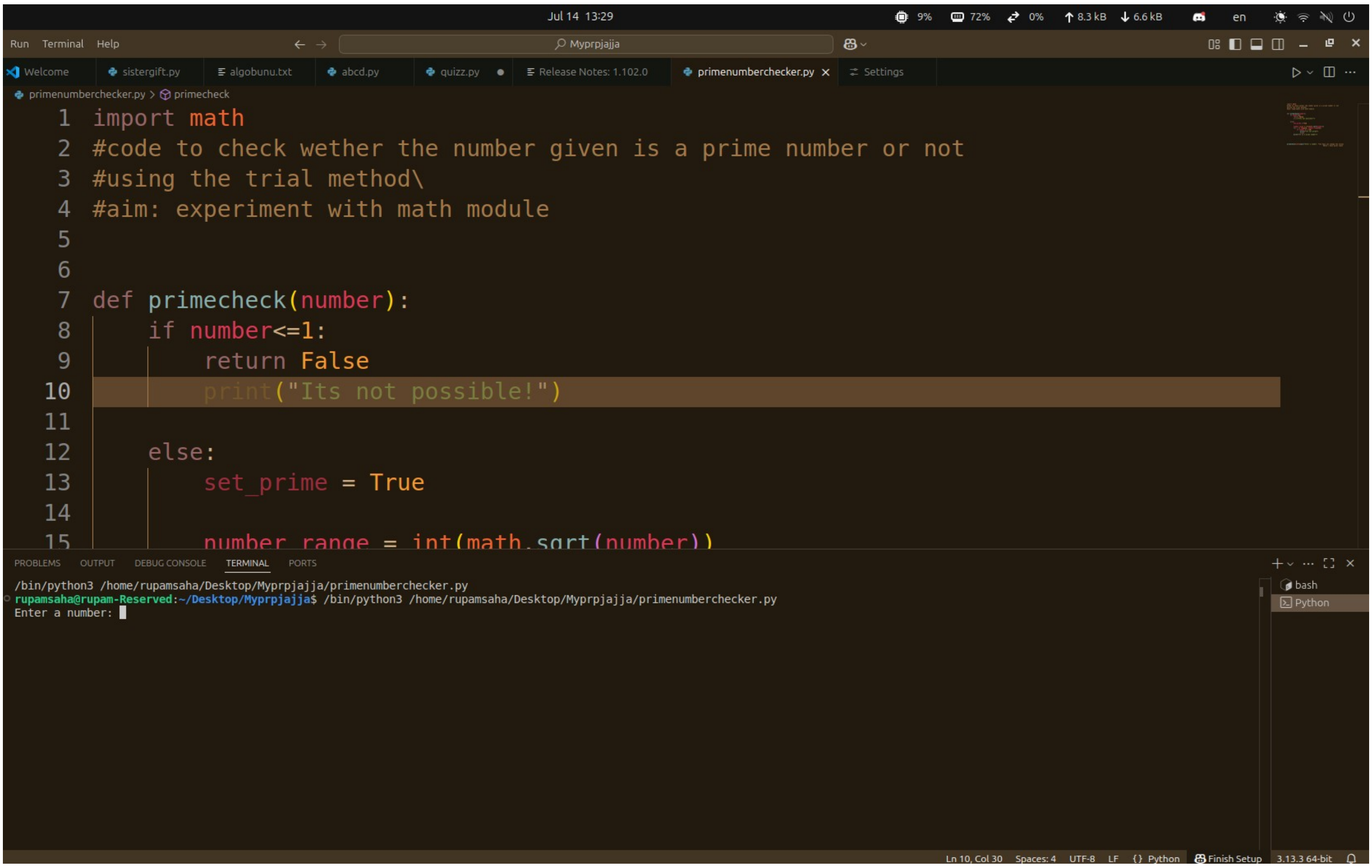
Now,

```
if number % s ==0:
print("its not prime")
break
```

As we know, the number will be divided by the numbers from 2 to √number so that we can check the number as prime or composite.
I used if conditional statement to check IF the numbers(the iterated numbers) are divisible by the number, then the process stops here. Hence the number is a composite numbers.

Samples with Photo

I have now talked about the heart of the entire process. Now we are going to see some examples to understand the code better.



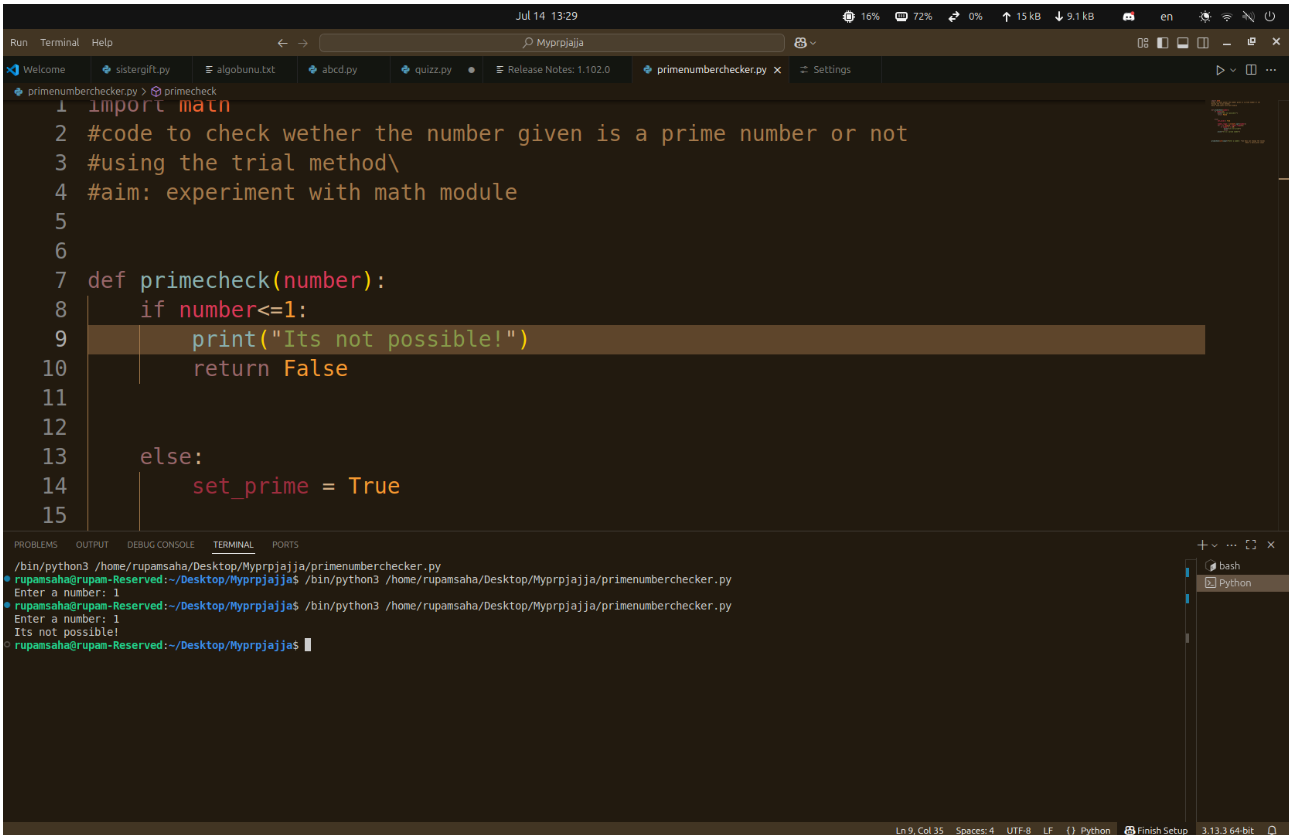
```python
import math
#code to check wether the number given is a prime number or not
#using the trial method\
#aim: experiment with math module


def primecheck(number):
    if number<=1:
        return False
        print("Its not possible!")

    else:
        set_prime = True

        number_range = int(math.sqrt(number))
```

```
/bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
rupamsaha@rupam-Reserved:~/Desktop/Myprpjajja$ /bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
Enter a number: 
```



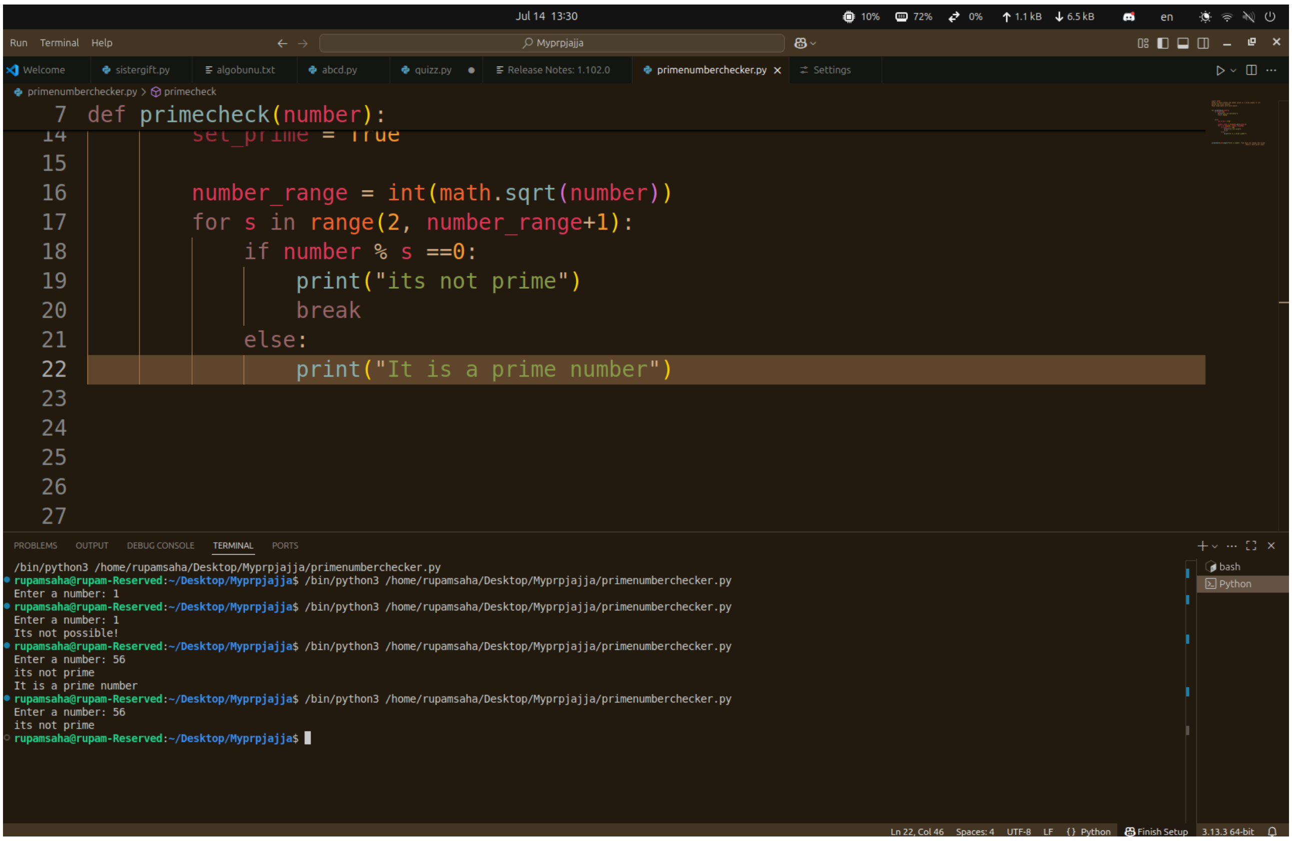
```python
import math
#code to check wether the number given is a prime number or not
#using the trial method\
#aim: experiment with math module


def primecheck(number):
    if number<=1:
        print("Its not possible!")
        return False


    else:
        set_prime = True

```

```
/bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
rupamsaha@rupam-Reserved:~/Desktop/Myprpjajja$ /bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
Enter a number: 1
rupamsaha@rupam-Reserved:~/Desktop/Myprpjajja$ /bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
Enter a number: 1
Its not possible!
rupamsaha@rupam-Reserved:~/Desktop/Myprpjajja$ 
```



```python
def primecheck(number):
        set_prime = True

        number_range = int(math.sqrt(number))
        for s in range(2, number_range+1):
            if number % s ==0:
                print("its not prime")
                break
            else:
                print("It is a prime number")





```

```
/bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
rupamsaha@rupam-Reserved:~/Desktop/Myprpjajja$ /bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
Enter a number: 1
rupamsaha@rupam-Reserved:~/Desktop/Myprpjajja$ /bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
Enter a number: 1
Its not possible!
rupamsaha@rupam-Reserved:~/Desktop/Myprpjajja$ /bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
Enter a number: 56
its not prime
It is a prime number
rupamsaha@rupam-Reserved:~/Desktop/Myprpjajja$ /bin/python3 /home/rupamsaha/Desktop/Myprpjajja/primenumberchecker.py
Enter a number: 56
its not prime
rupamsaha@rupam-Reserved:~/Desktop/Myprpjajja$ 
```

These are some samples I did with my code. You can check out the github.

Conclusion

We have now finished the project and now we are coming to the conclusion.

We saw that we used loops, functions and math to create this program.

I will be working on more projects on solving small math problems.

References:

Images: My own laptop

OS used: Ubuntu 25.04 Plucky Puffin(Linux Distro)

Modules: Math

Prime Number Theory: Wikipedia