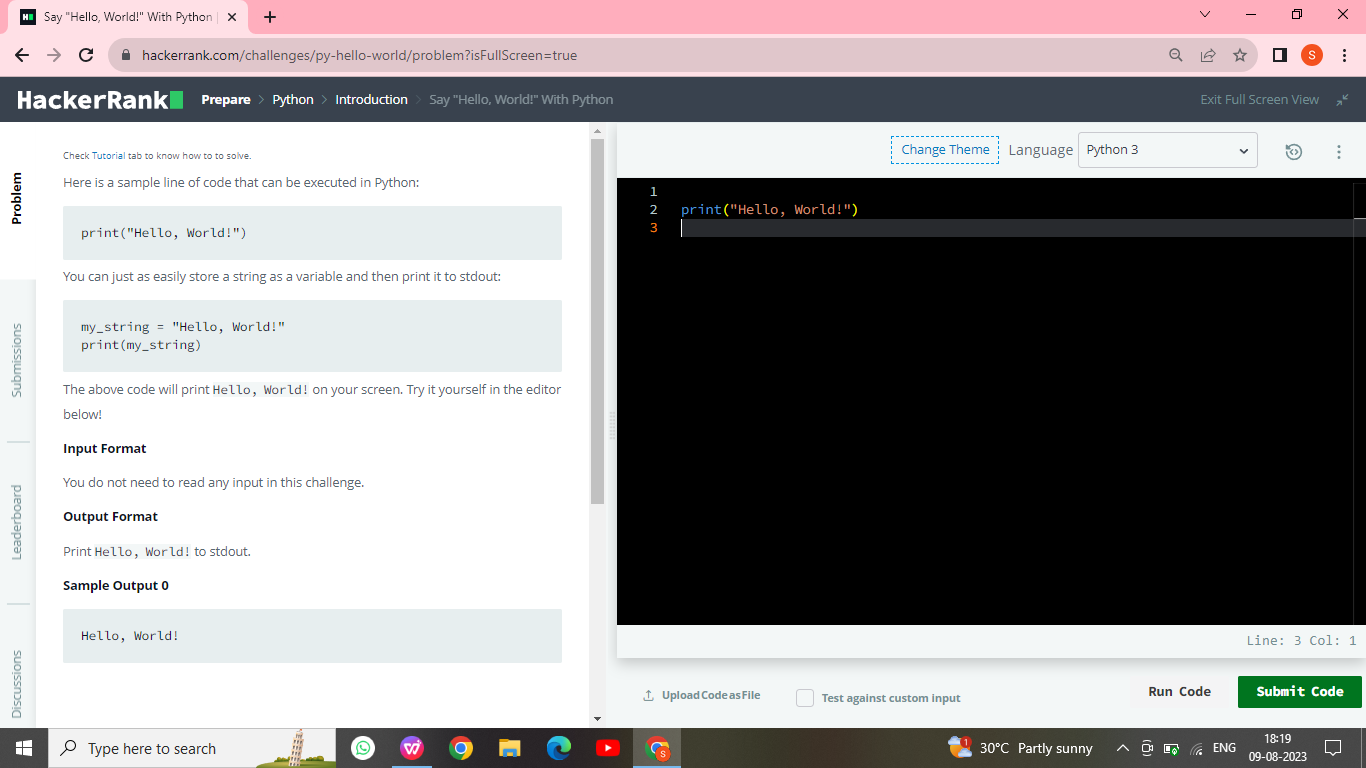
HACKER RANK PYTHON PROBLEM SOLVING

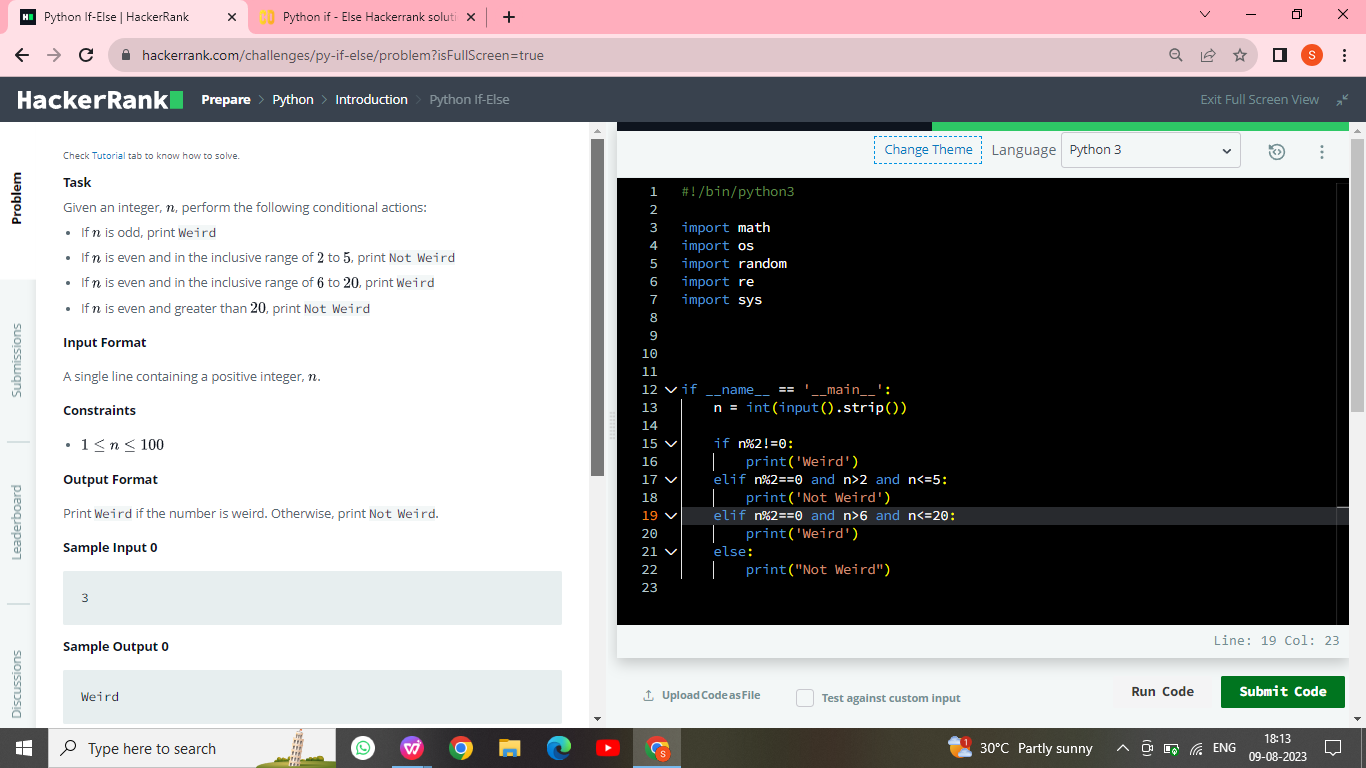
From Today Onwards

09-08-2023

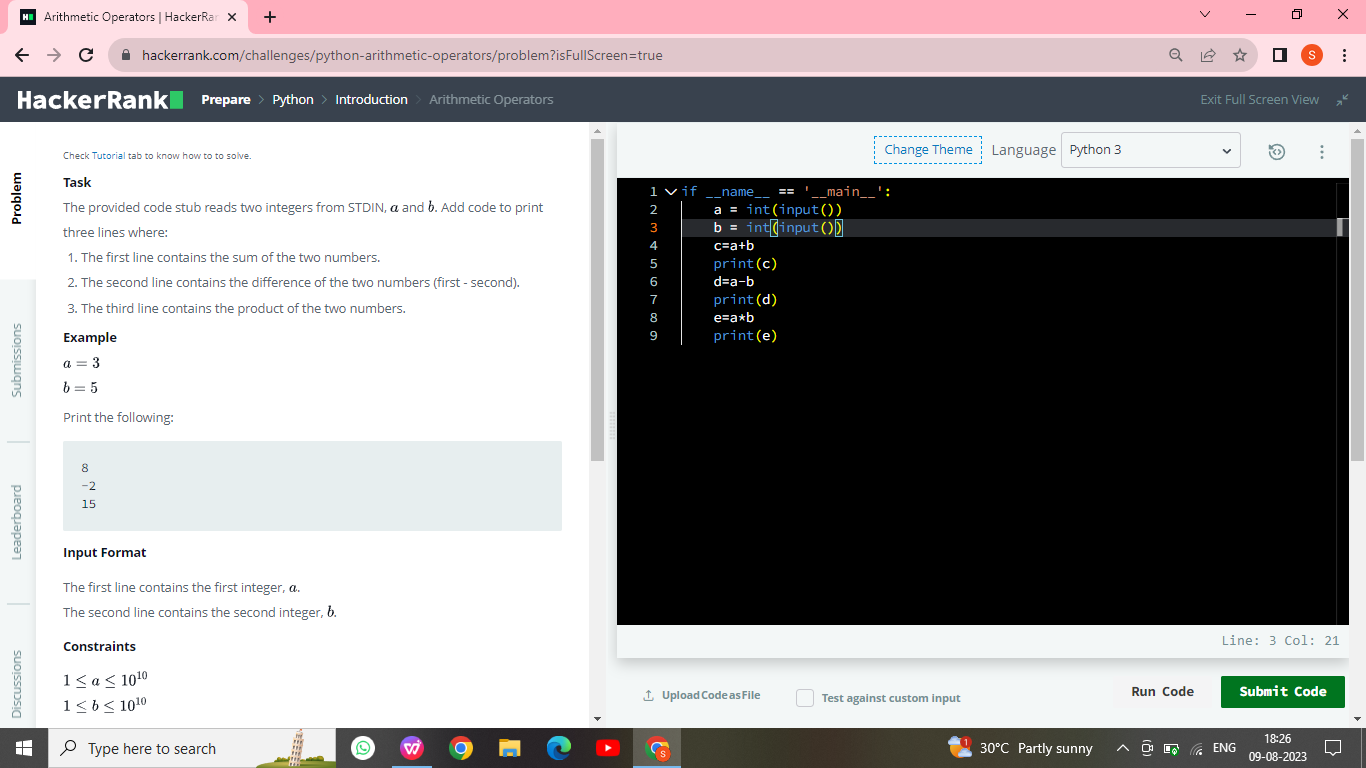
1. Python



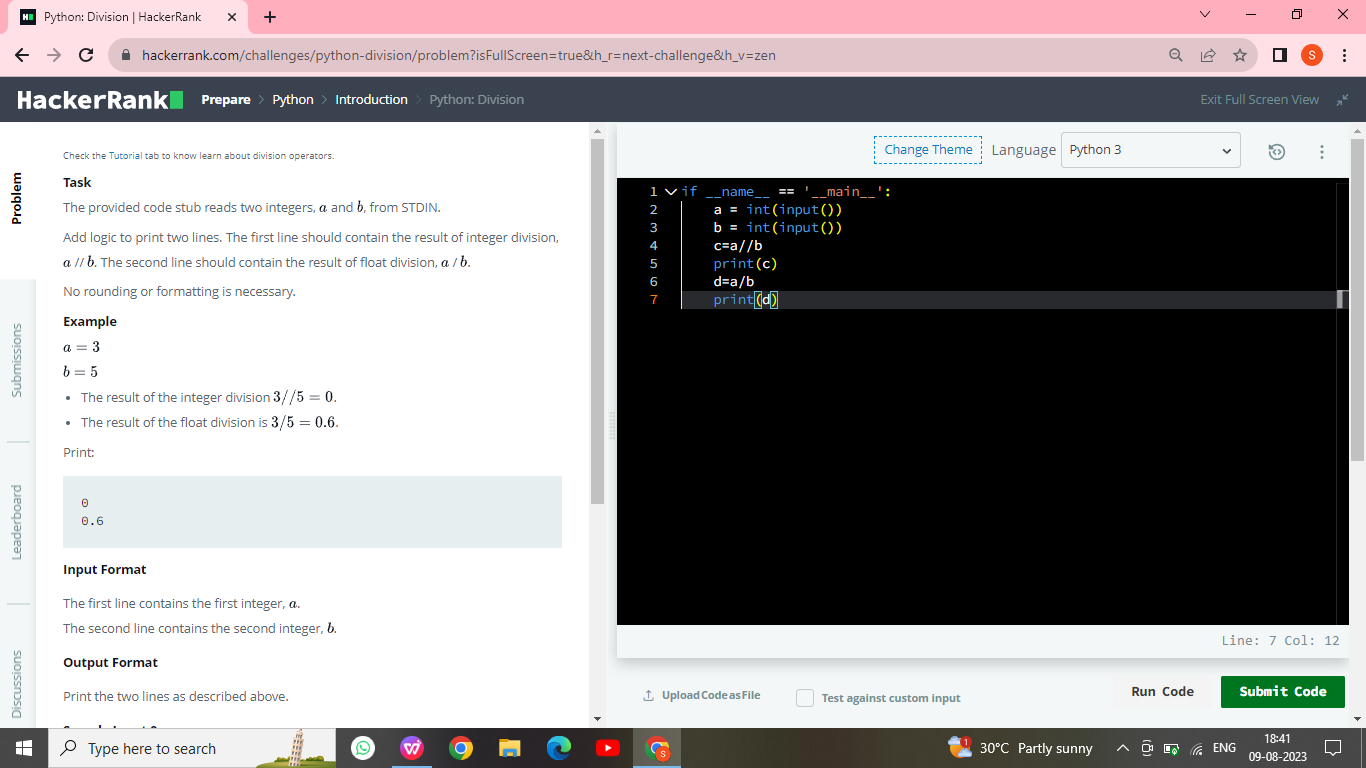
2 . Python If-else



3 . Arithmetic Operators



1. Python Division



1. ****Task****  
   The provided code stub reads and integer, , from STDIN. For all non-negative integers , print .

****Example****

The list of non-negative integers that are less than  is . Print the square of each number on a separate line.

0

1

4

****Input Format****

The first and only line contains the integer, .

****Constraints****

****Output Format****

Print  lines, one corresponding to each .

****Sample Input 0****

5

****Sample Output 0****

014916

**METHOD 1**

if \_\_name\_\_ == '\_\_main\_\_':

    n = int(input())

    i=0

while i<n:

    print(i\*\*2)

    i=i+1

**METHOD 2**

n = int(input())

[print(n \* n) for n in range(0, n)]

METHOD 3

import math as m if \_\_name\_\_ == '\_\_main\_\_':

n = int(input())

for i in range(n):

print(int(m.pow(i,2)))

10-08-2023

1. Leap year or not using function in Python

****Task****

Given a year, determine whether it is a leap year. If it is a leap year, return the Boolean True, otherwise return False.

Note that the code stub provided reads from STDIN and passes arguments to the is\_leap function. It is only necessary to complete the is\_leap function.

****Input Format****

Read , the year to test.

****Constraints****

****Output Format****

The function must return a Boolean value (True/False). Output is handled by the provided code stub.

****Sample Input 0****

1990

****Sample Output 0****

False

****Explanation 0****

1990 is not a multiple of 4 hence it's not a leap year.

**PROGRAM**

**def** is\_leap(year):

**if** 1900<=year<=10\*\*5:

**if**(year%400==0 **and** year%100==0) **or** (year%4==0 **and** year%100!=0):

**return** **True**

**else**:

**return** **False**

**else**:

**return** **False**

year = **int**(**input**())

**print**(is\_leap(year))