1) Say "Hello, World!" With Python

In [1]:
```
print("Hello, World!")
```

```
Hello, World!
```

2) Python - if - else Task.

Given an integer,n, perform the following conditional actions:

If is odd, print Weird.
If is even and in the inclusive range of 2 to 5, print Not Weird.
If is even and in the inclusive range of 6 to 20, print Weird.
If is even and greater than 20, print Not Weird.

In [2]:
```
import math
import os
import random
import re
import sys

n = int(input())
if n % 2 != 0:
    print("Weird")
elif n % 2 == 0 and 2 <= n <= 5:
    print("Not Weird")
elif n % 2 == 0 and 6 <= n <= 20:
    print("Weird")
else:
    print("Not Weird")
```

```
20
Weird
```

3) Arithematic Operators Task.

The provided code stub reads two integers from STDIN, a and b. Add code to print three lines where:

- The first line contains the sum of the two numbers.
- The second line contains the difference of the two numbers (first - second).
- The third line contains the product of the two numbers.

In [4]:
```
if __name__ == '__main__':
    a = int(input())
    b = int(input())
    print("Addition",a+b)
    print("Subtraction",a-b)
    print("Multiplication",a*b)
```

```
25
50
Addition 75
Subtraction -25
Multiplication 1250
```

4) Python : Division Task.

The provided code stub reads two integers, a and b, from STDIN.

Add logic to print two lines. The first line should contain the result of integer division, a // b. The second line should contain the result of float division, a / b.

No rounding or formatting is necessary.

In [5]:
```
if __name__ == '__main__':
    a = int(input())
    b = int(input())
    print("Floor Division",a//b)
    print("Division",a/b)
```

```
5
10
Floor Division 0
Division 0.5
```

5) Loops

The provided code stub reads and integer, n, from STDIN. For all non-negative integers i < n, print i^2.

In [6]:
```
if __name__ == '__main__':
    n = int(input())
    for i in range(n):
        print(i*i)
```

```
5
0
1
4
9
16
```

6) Write a Function Task.

An extra day is added to the calendar almost every four years as February 29, and the day is called a leap day. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap day.

In the Gregorian calendar, three conditions are used to identify leap years:

- The year can be evenly divided by 4, is a leap year, unless:
  - The year can be evenly divided by 100, it is NOT a leap year, unless:
    - The year is also evenly divisible by 400. Then it is a leap year.

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.

In [8]:
```
def is_leap(year):
    leap = False
#The year can be evenly divided by 4, is a leap year, unless:
#The year can be evenly divided by 100, it is NOT a leap year, unless:
#The year is also evenly divisible by 400. Then it is a leap year.
#This means that in the Gregorian calendar, the years 2000 and 2400 are leap
    # Write your logic here
    if year%4 == 0:
        if year%100 == 0:
            if year%400 == 0:
                leap = True
            else:
                leap = False
        else:
            leap = True
    else:
        leap = False
    return leap
year = int(input())
print(is_leap(year))
```

```
1990
False
```

7) Print a function.

The included code stub will read an integer, n, from STDIN.

Without using any string methods, try to print the following: 123...n
Note that "..." represents the consecutive values in between.

In [9]:
```
if __name__ == '__main__':
    n = int(input())
    for i in range(n):
        print(i+1, end = '')
```

```
5
12345
```