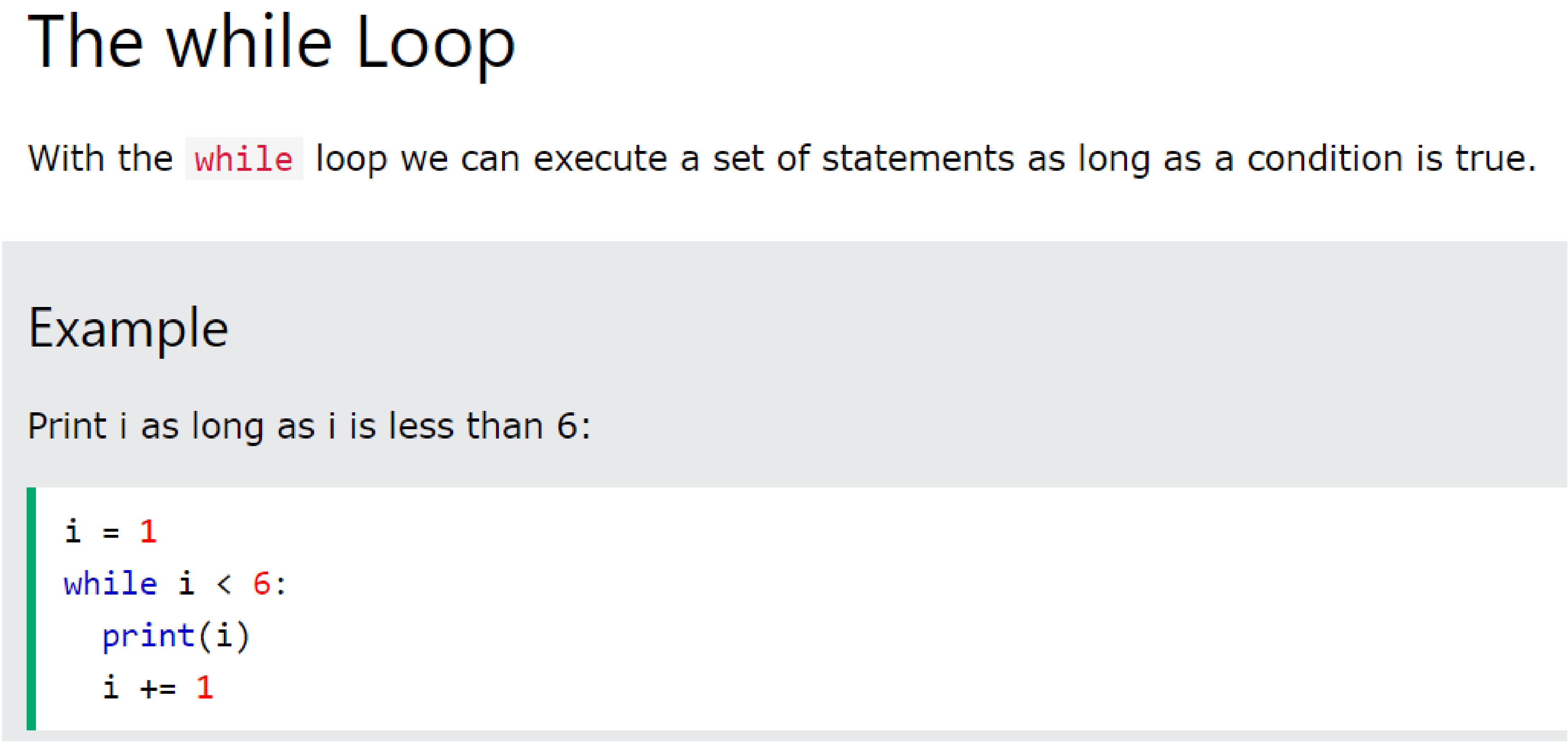


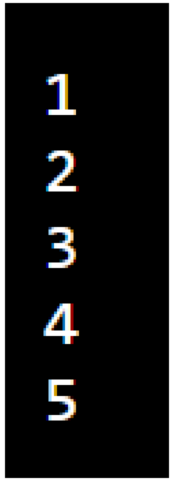
The while Loop

With the `while` loop we can execute a set of statements as long as a condition is true.

Example

Print i as long as i is less than 6:

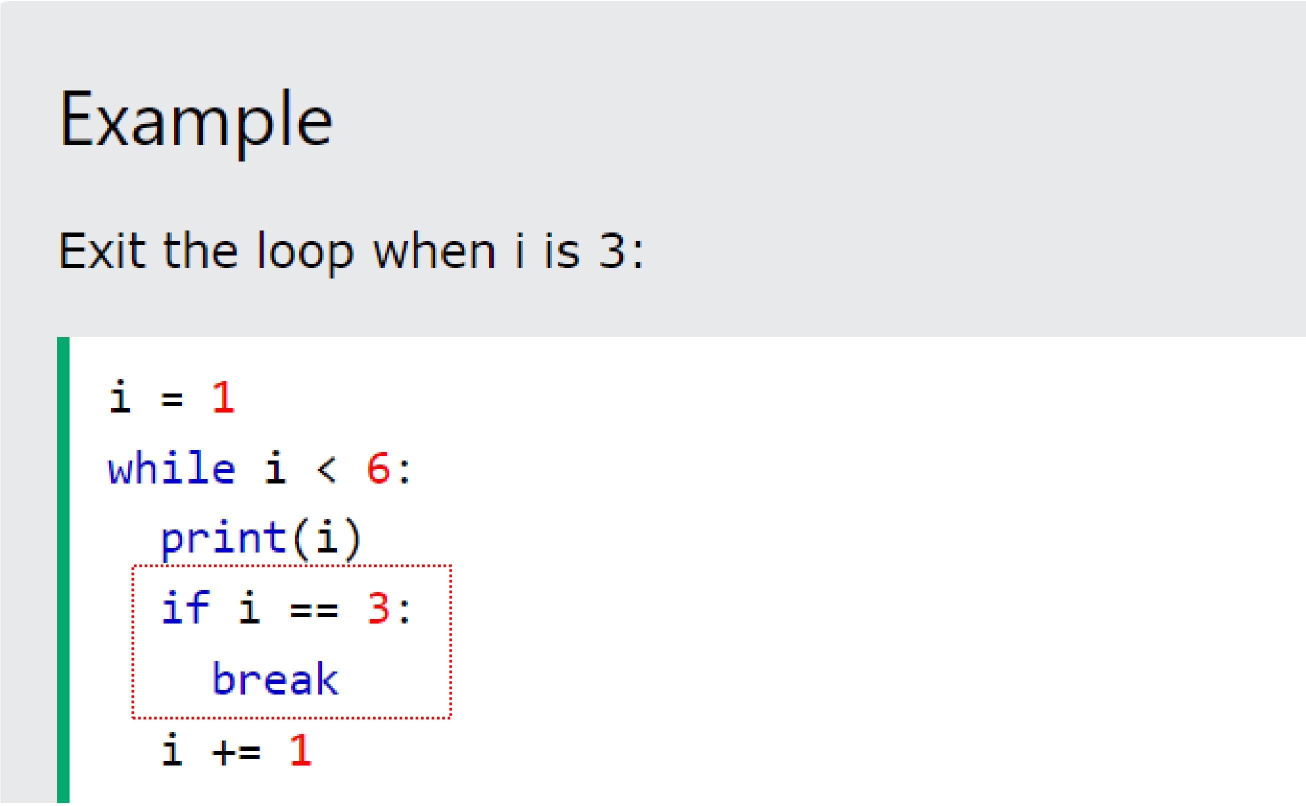
```
i = 1
while i < 6:
  print(i)
  i += 1
```



```
1
2
3
4
5
```

The break Statement

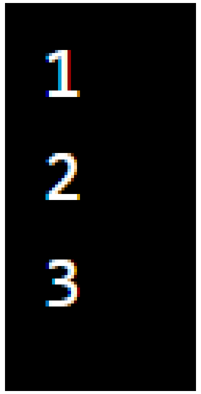
With the `break` statement we can stop the loop even if the while condition is true:



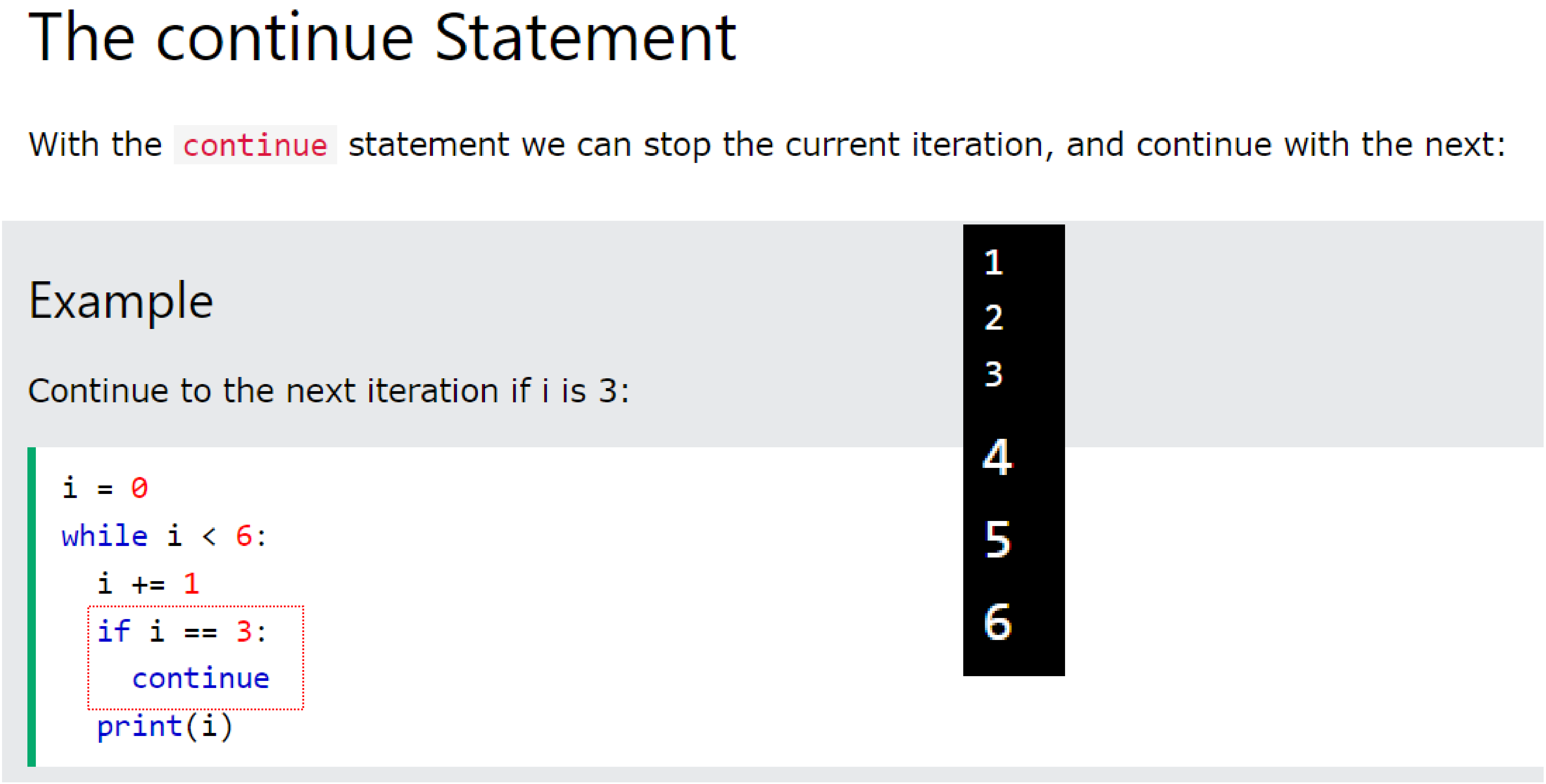
Example

Exit the loop when i is 3:

```
i = 1
while i < 6:
  print(i)
  if i == 3:
    break
  i += 1
```



```
1
2
3
```



The continue Statement

With the `continue` statement we can stop the current iteration, and continue with the next:

Example

Continue to the next iteration if i is 3:

```
i = 0
while i < 6:
  i += 1
  if i == 3:
    continue
  print(i)
```

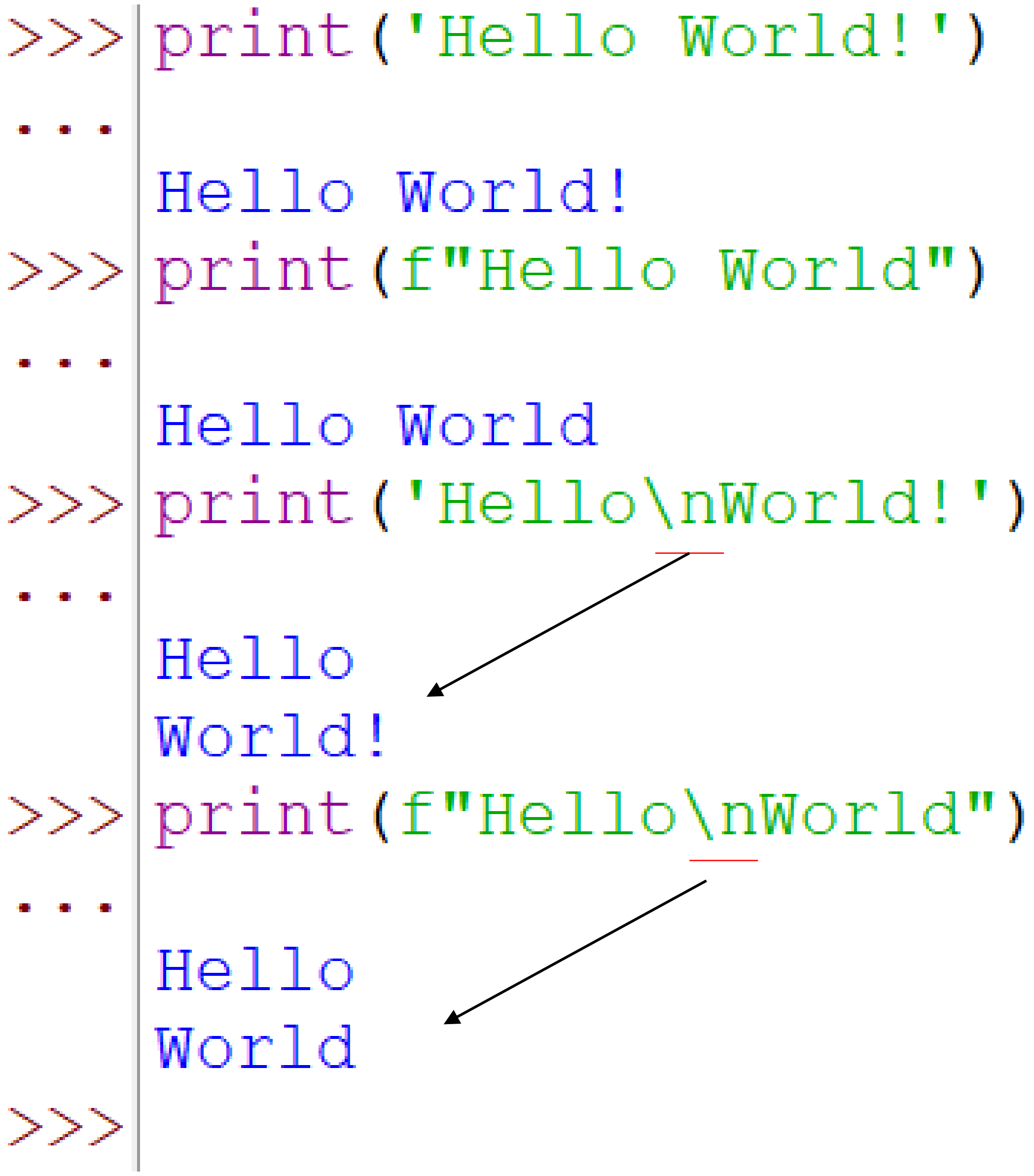
```
1
2
3
4
5
6
```

Python New Line

\n

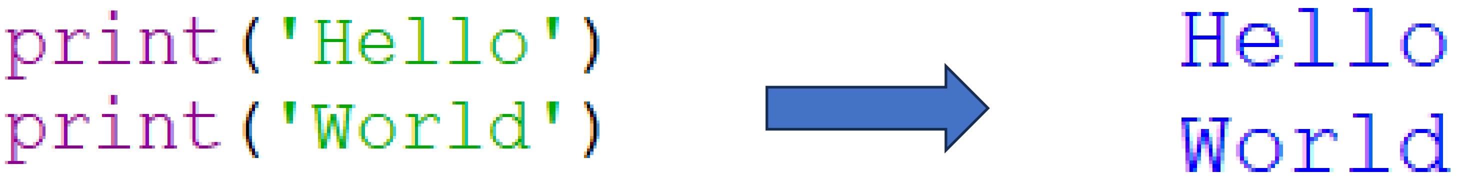
It is made of two characters:

- A backslash.
- The letter n.



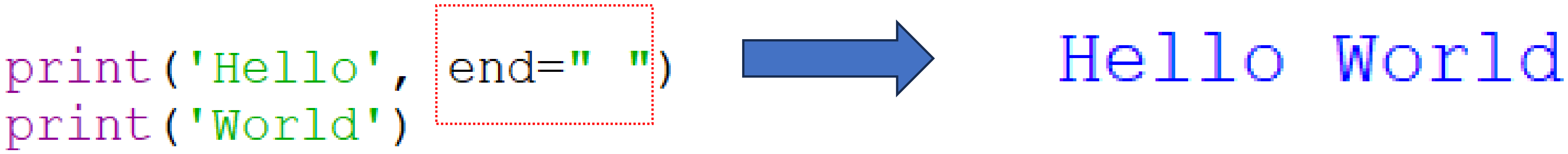
```
>>> print('Hello World!')
...
    Hello World!
>>> print(f"Hello World")
...
    Hello World
>>> print('Hello\nWorld!')
...
    Hello
    World!
>>> print(f"Hello\nWorld")
...
    Hello
    World
>>>
```

How To Python Print Without A Newline



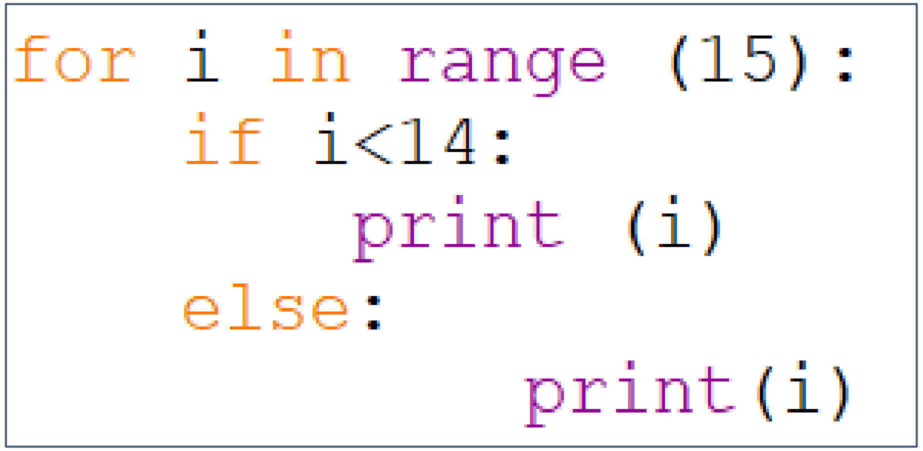
```
print('Hello')
print('World')
```

```
Hello
World
```

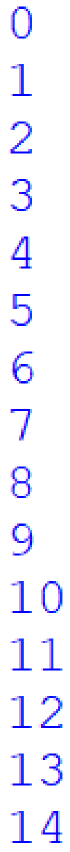


```
print('Hello', end=" ")
print('World')
```

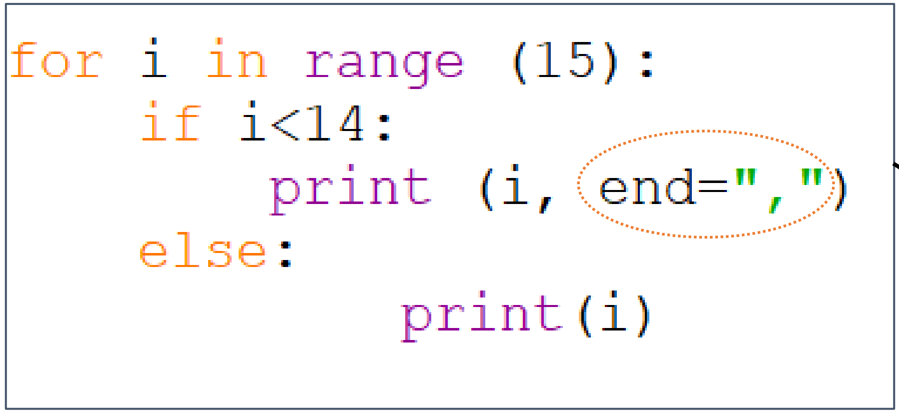
```
Hello World
```



```
for i in range (15):
    if i<14:
        print (i)
    else:
            print(i)
```

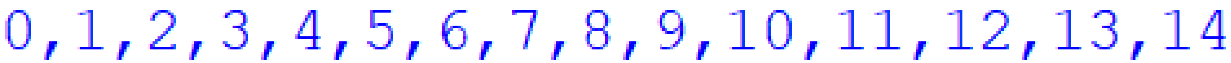


```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
```



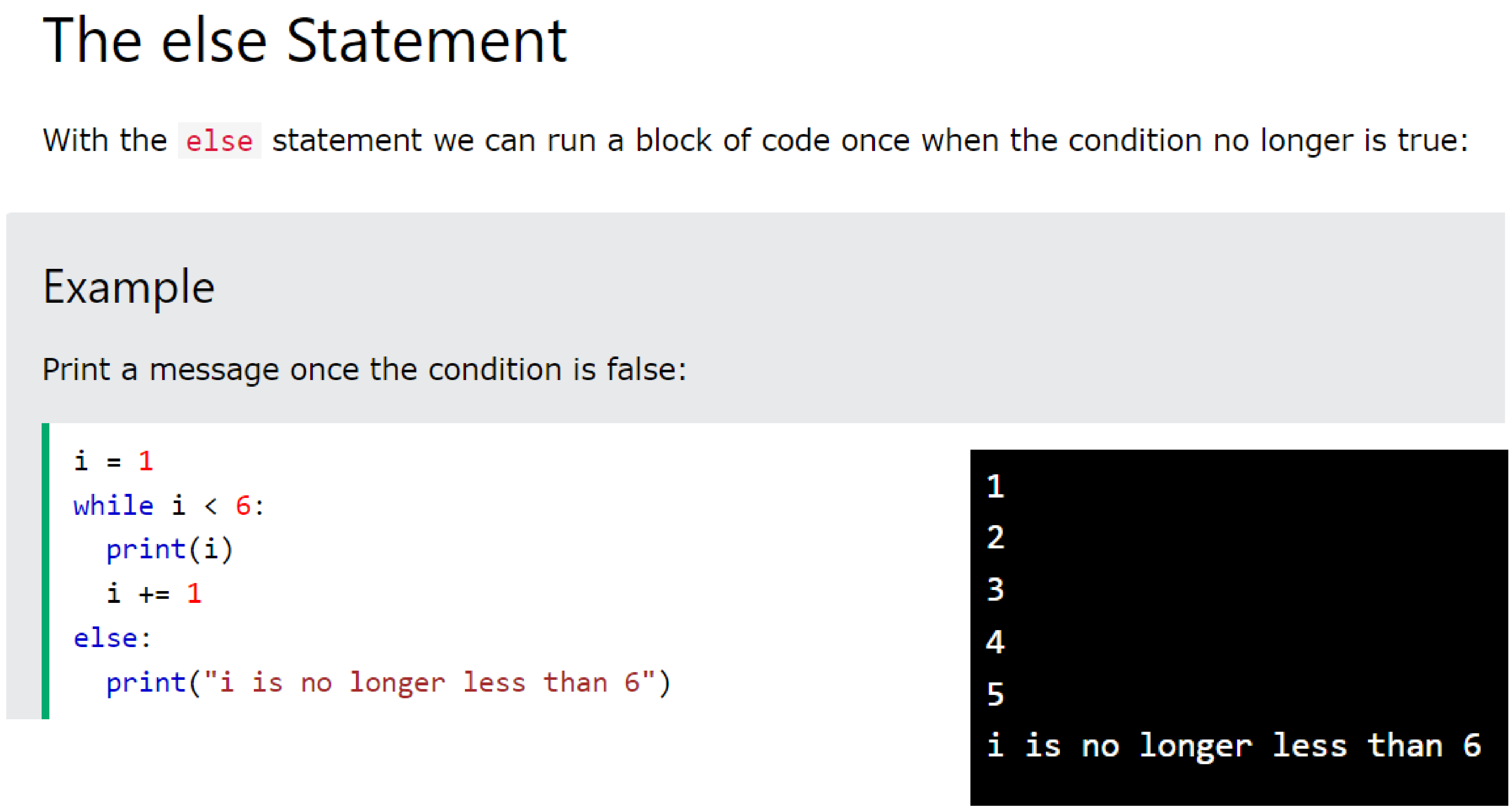
```
for i in range (15):
    if i<14:
        print (i, end=",")
    else:
            print(i)
```

end = “ , ”



```
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
```

While else



The else Statement

With the `else` statement we can run a block of code once when the condition no longer is true:

Example

Print a message once the condition is false:

```
i = 1
while i < 6:
  print(i)
  i += 1
else:
  print("i is no longer less than 6")
```

```
1
2
3
4
5
i is no longer less than 6
```

1. Write a Python program using the while loop that will produce the computational output as below:

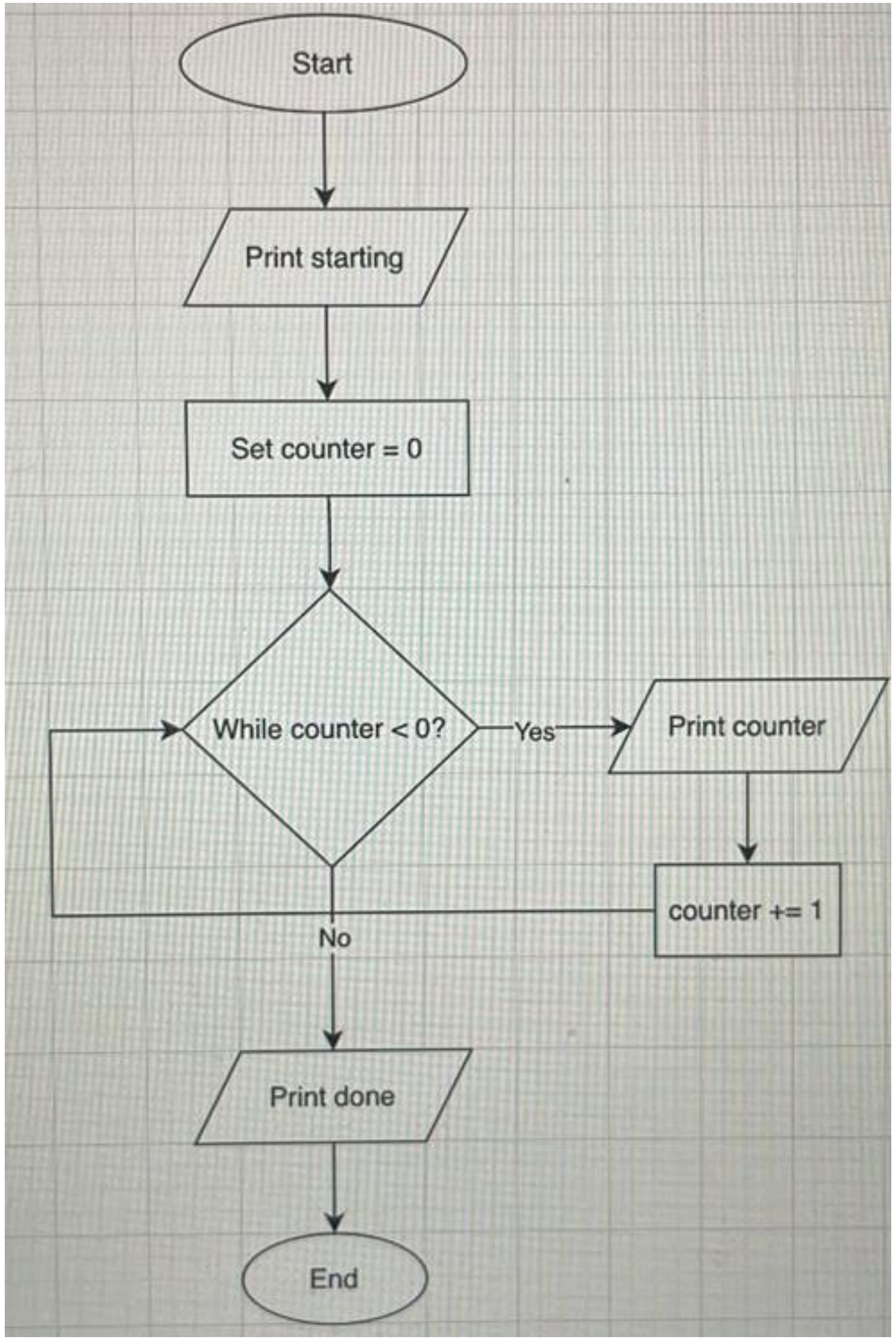
```
Starting
0 1 2 3 4 5 6 7 8 9
Done
```

Descriptions:

- Set a counter to 0
- Print “Starting”
- While counter is less than 10
- Print counter
- Increment counter to 1
- End the loop

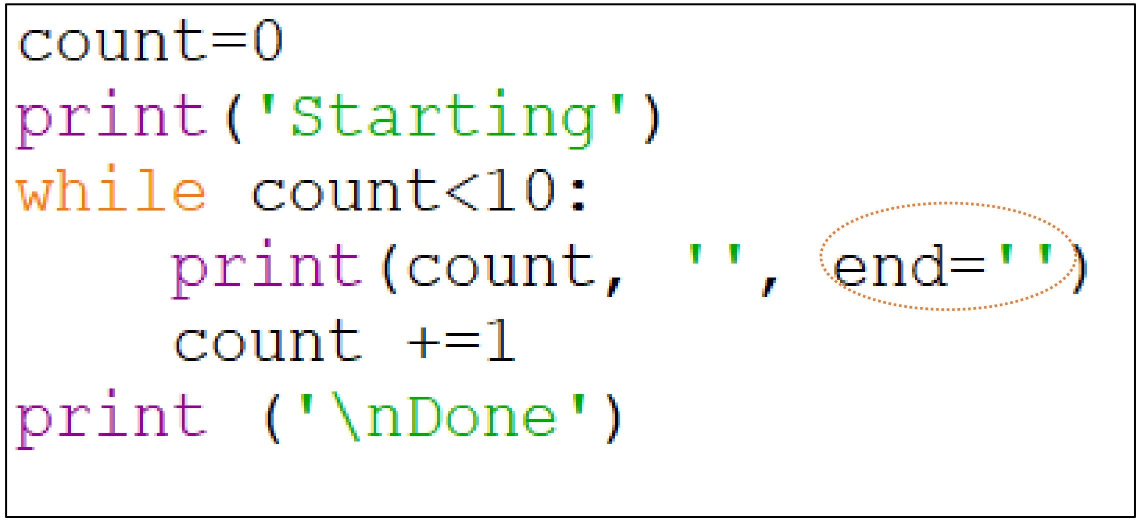
Flowchart

- Start
- Print "Starting“
- Set counter = 0
- While counter < 10
- Print counter
- Increment counter by 1
- End While
- Print "Done“
- End

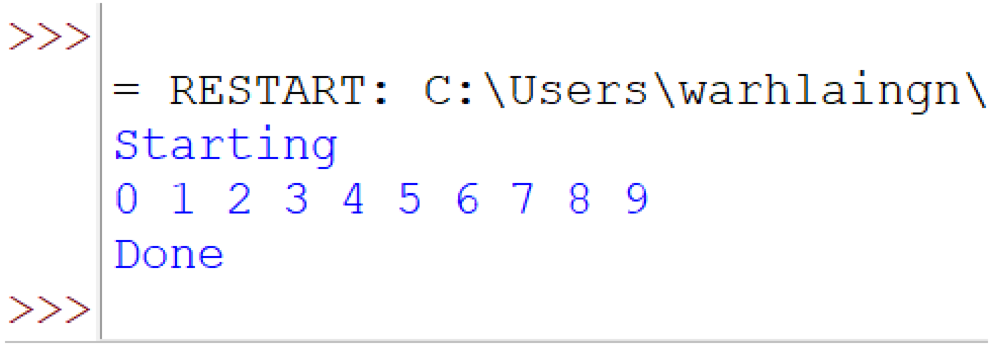


1. Write a Python program using the while loop that will produce the computational output as below:

```
Starting
0 1 2 3 4 5 6 7 8 9
Done
```



```
count=0
print('Starting')
while count<10:
    print(count, '', end='')
    count +=1
print ('\nDone')
```



```
>>>
= RESTART: C:\Users\warhlaingn\
Starting
0 1 2 3 4 5 6 7 8 9
Done
>>>
```

2. Write a Python program using the while loop that will produce the computational output as below:

```
Starting
-20 -18 -16 -14 -12 -10 -8 -6 -4 -2
Done
```

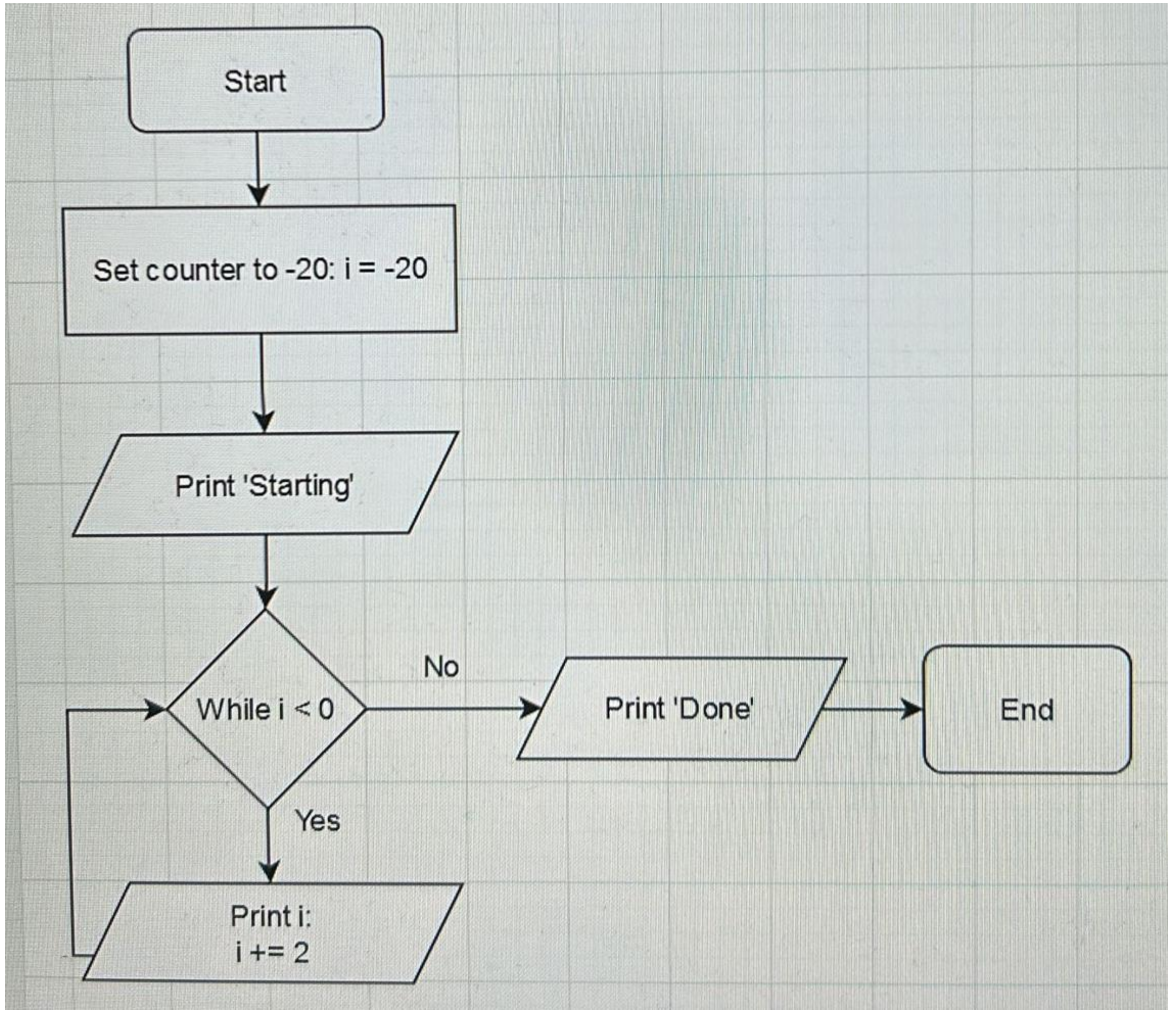
Descriptions:

- Set a counter to ???
- Print “Starting”
- While counter is less than 0
- Print counter
- Increment counter to ???
- End the loop

Flowchart

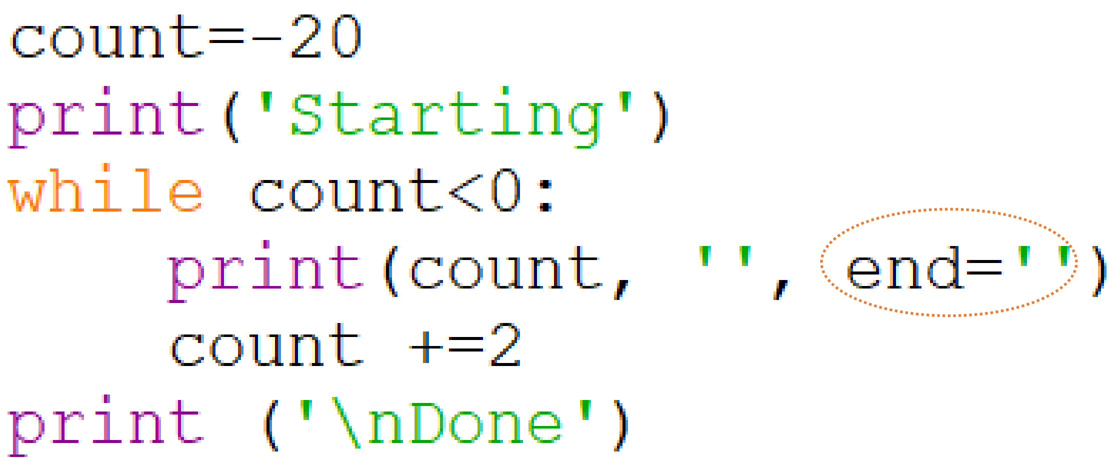
- Start
- Print "Starting"
- Set counter = -20
- While counter < 0
- Print counter
- Increment counter by 2
- End While
- Print "Done"
- End

```
Starting
-20 -18 -16 -14 -12 -10 -8 -6 -4 -2
Done
```

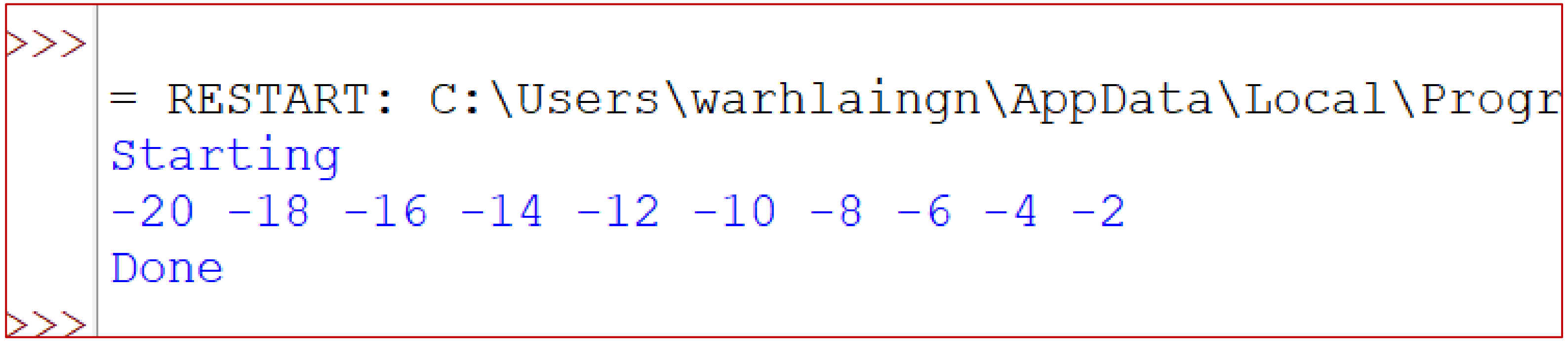


2. Write a Python program using the while loop that will produce the computational output as below:

```
Starting
-20 -18 -16 -14 -12 -10 -8 -6 -4 -2
Done
```



```
count=-20
print('Starting')
while count<0:
    print(count, '', end='')
    count +=2
print ('\nDone')
```



```
>>>
= RESTART: C:\Users\warhlaingn\AppData\Local\Progr
Starting
-20 -18 -16 -14 -12 -10 -8 -6 -4 -2
Done
>>>
```

3. Write a Python program using the while loop that prompts the user to enter two numbers 'a' and 'b' and **calculate 'a' to the power of 'b'.**

- Initialize result to 1 (to multiple with a for b times)
- Initialize counter to 1

$$a^b = \overbrace{a \times a \times a}^{b}$$

if $b = 1$

$$a^1 = a^1$$

if $b = 2$

$$a^2 = a^1 \times a^1$$

if $b = 3$

$$a^3 = a^1 \times a^1 \times a^1$$

$$= a^{1+1+1}$$

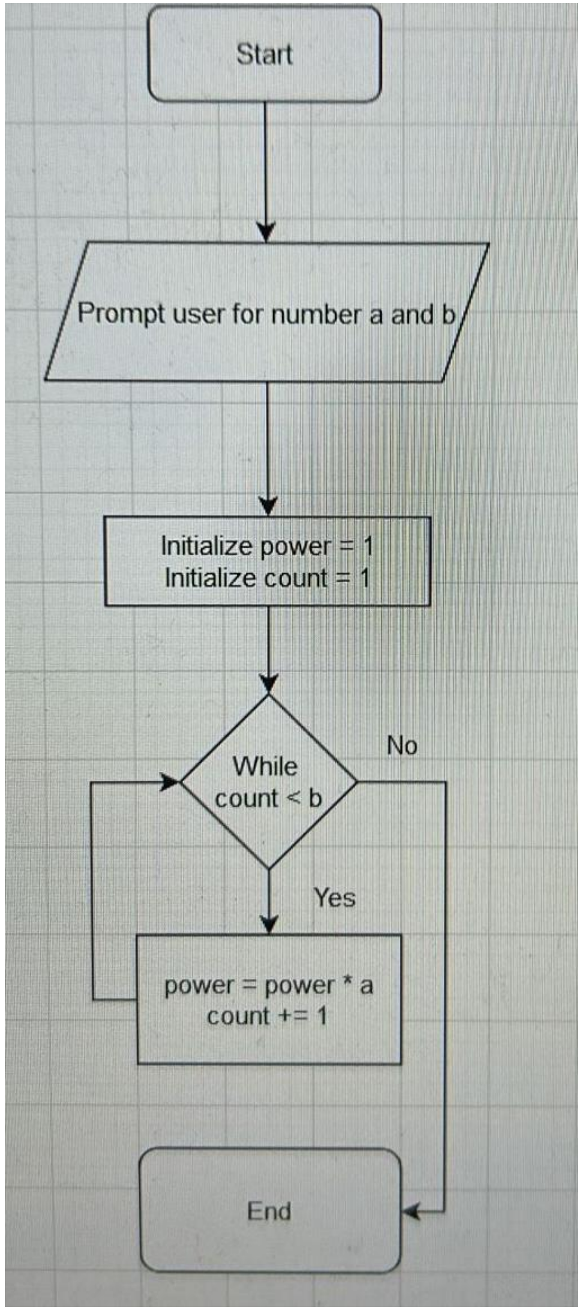
$$= a^3$$

3. Write a Python program using the while loop that prompts the user to enter two numbers 'a' and 'b' and **calculate 'a' to the power of 'b'.**

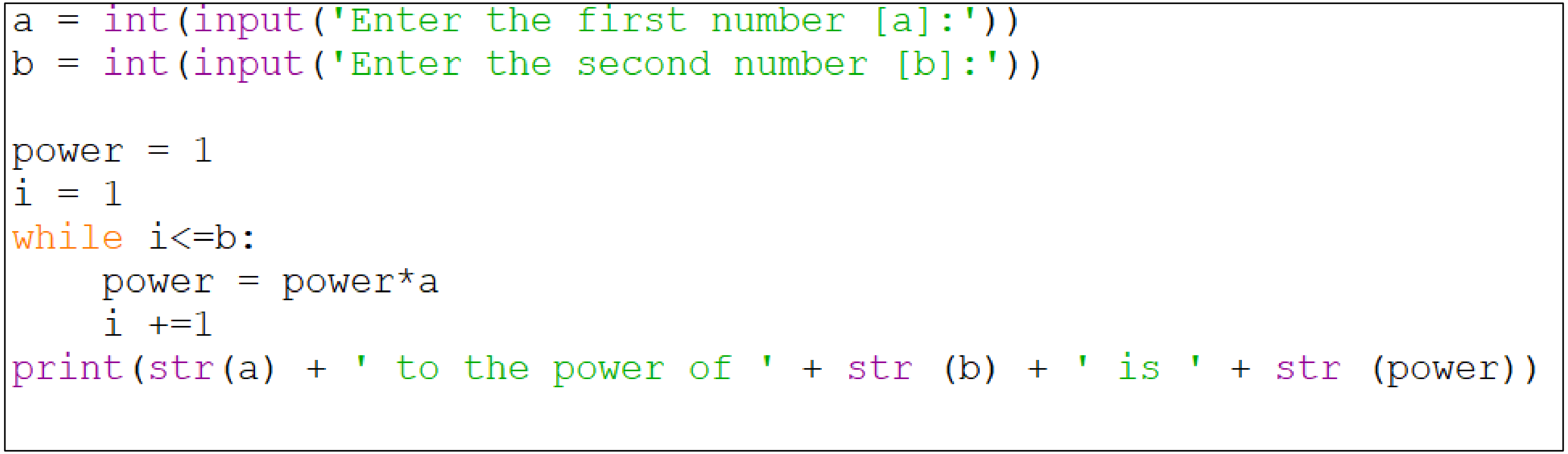
Descriptions:

- Prompt user for input of 'a’ and Prompt user for input of 'b'
- Initialize result to 1 (to multiple with a for b times)
- Initialize counter to 1
- While counter < b: Multiply result by a
- Increment counter
- End of Loop
- Print result

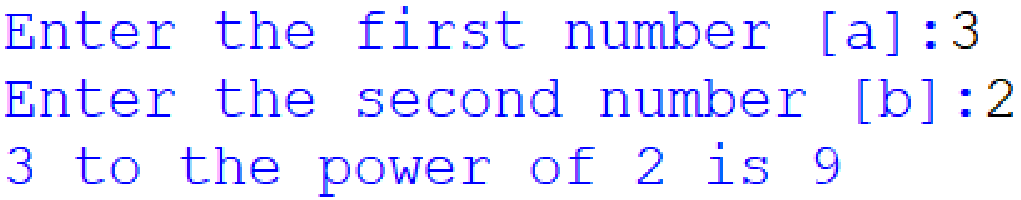
Flowchart



3. Write a Python program using the **while loop** that prompts the user to enter two numbers 'a' and 'b' and calculate 'a' to the power of 'b'.



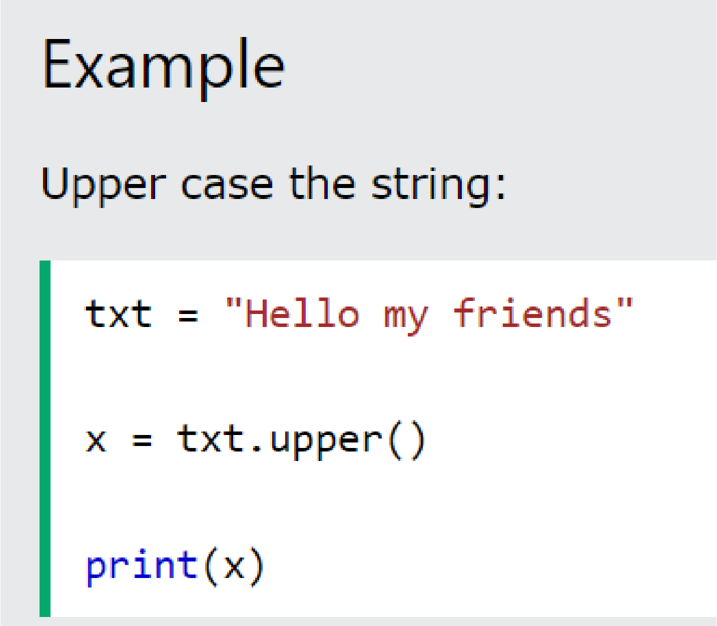
```python
a = int(input('Enter the first number [a]:'))
b = int(input('Enter the second number [b]:'))

power = 1
i = 1
while i<=b:
    power = power*a
    i +=1
print(str(a) + ' to the power of ' + str (b) + ' is ' + str (power))
```



```
Enter the first number [a]:3
Enter the second number [b]:2
3 to the power of 2 is 9
```

Python String upper() Method

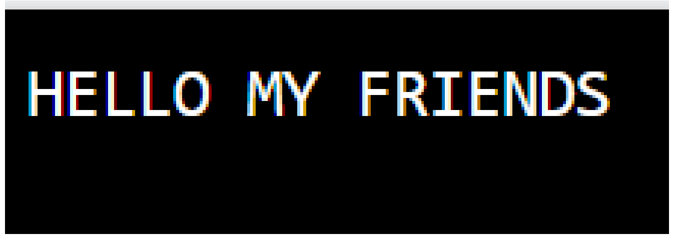
upper()   Converts a string into upper case



Example

Upper case the string:

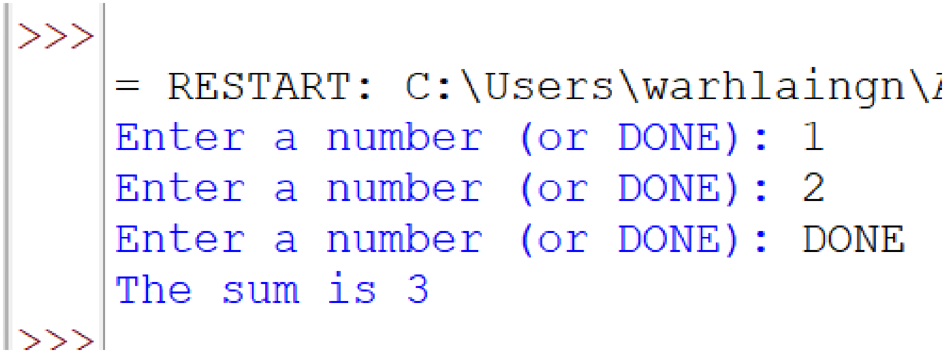
```
txt = "Hello my friends"

x = txt.upper()

print(x)
```



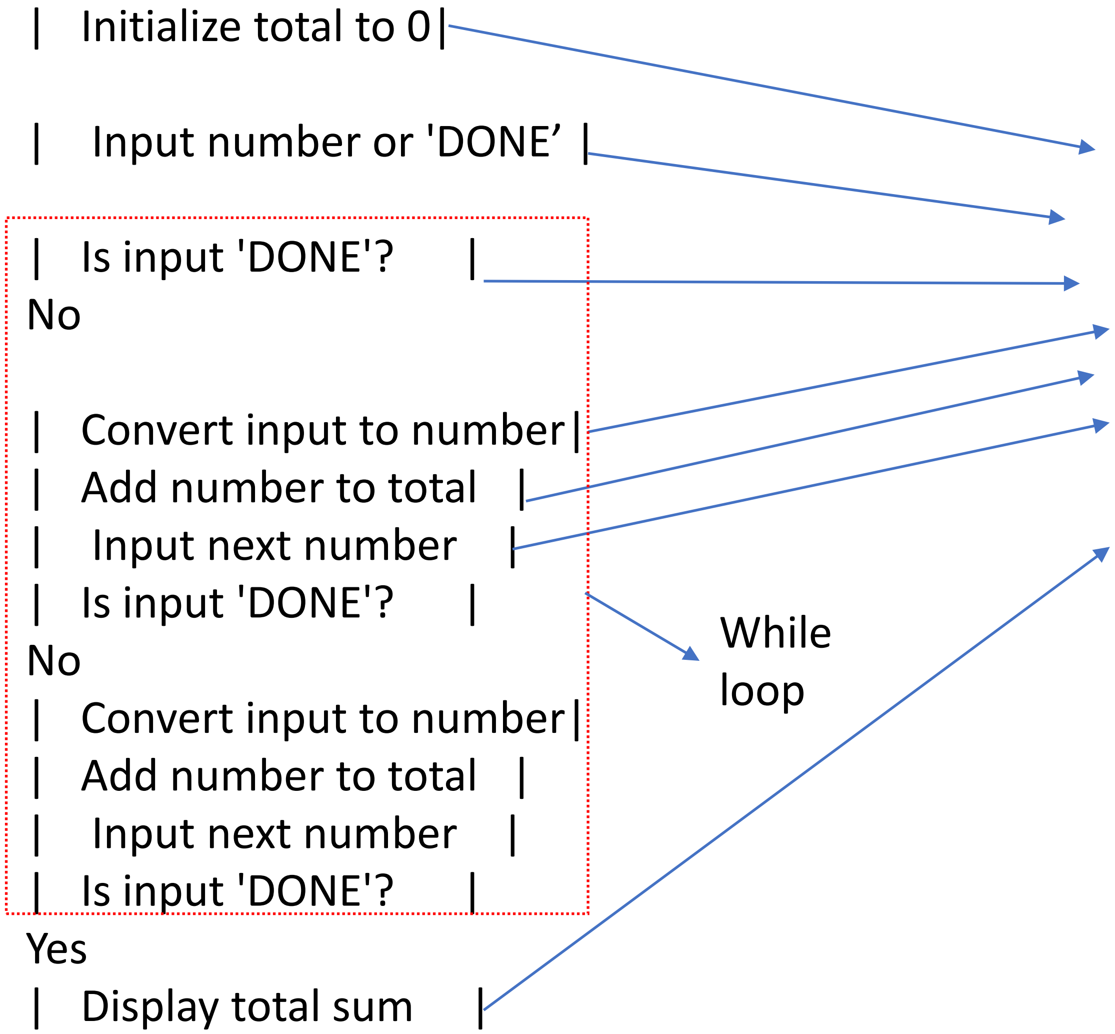
```
HELLO MY FRIENDS
```

4. Write a Python program using the **while loop** that performs summing an unknown number of numbers **until the user types 'DONE', then the program end.**

- Initialize a variable total to store the sum of numbers:
  Set total to 0.
- Input Number: Prompt the user to input a number or 'DONE’.
- Check if the input is 'DONE’.
  If input is 'DONE', Display the total sum.
  If input is a number, Add the input number to the total.
- Input Next Number: to input the next number.
- Display the total sum.

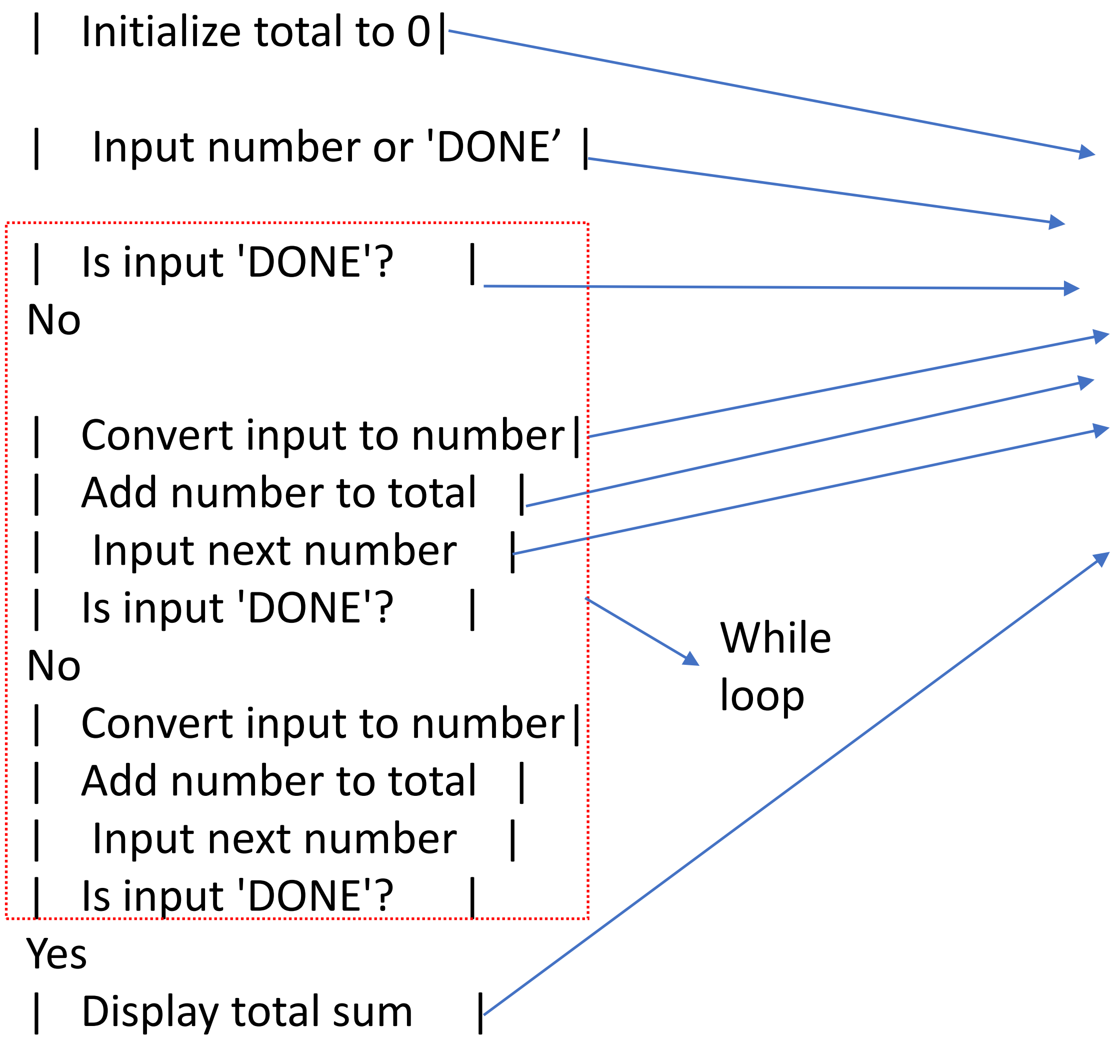


```
>>>
    = RESTART: C:\Users\warhlaingn\
    Enter a number (or DONE): 1
    Enter a number (or DONE): 2
    Enter a number (or DONE): DONE
    The sum is 3
>>>
```

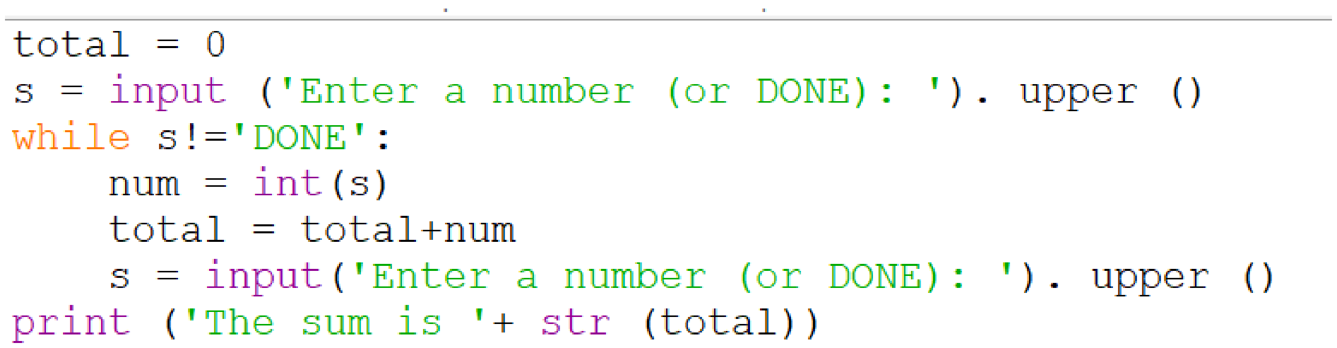


```
|   Initialize total to 0|

|   Input number or 'DONE’ |

|   Is input 'DONE'?       |
No

|   Convert input to number|
|   Add number to total   |
|   Input next number     |
|   Is input 'DONE'?       |
No
|   Convert input to number|
|   Add number to total   |
|   Input next number     |
|   Is input 'DONE'?       |
Yes
|   Display total sum      |
```

Draw
Flowchart

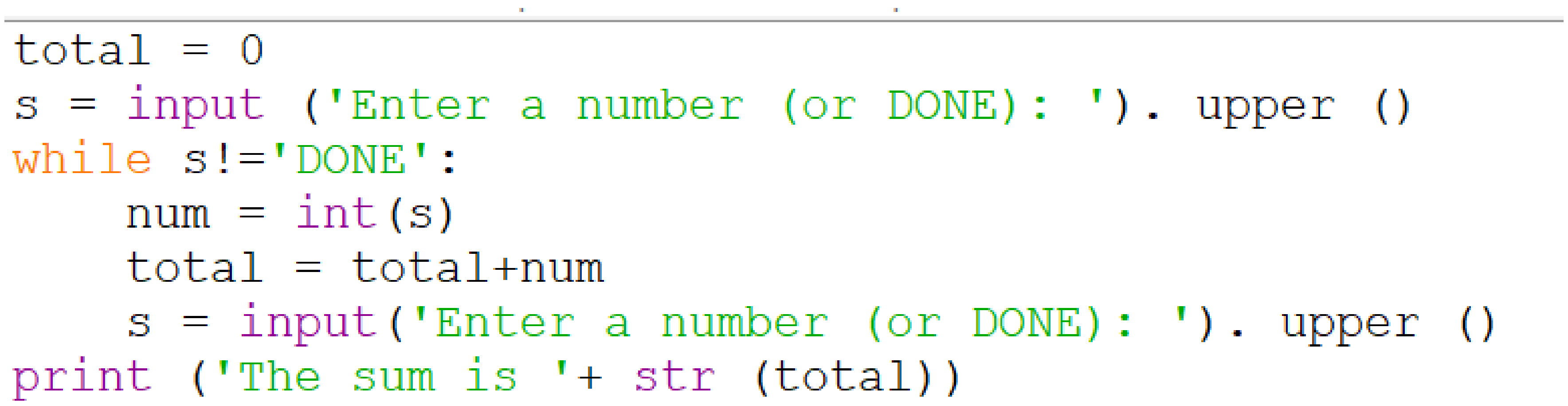


```
|   Initialize total to 0|

|   Input number or 'DONE’ |

|   Is input 'DONE'?      |
No

|   Convert input to number|
|   Add number to total   |
|   Input next number     |
|   Is input 'DONE'?      |
No
|   Convert input to number|
|   Add number to total   |
|   Input next number     |
|   Is input 'DONE'?      |
Yes
|   Display total sum     |
```

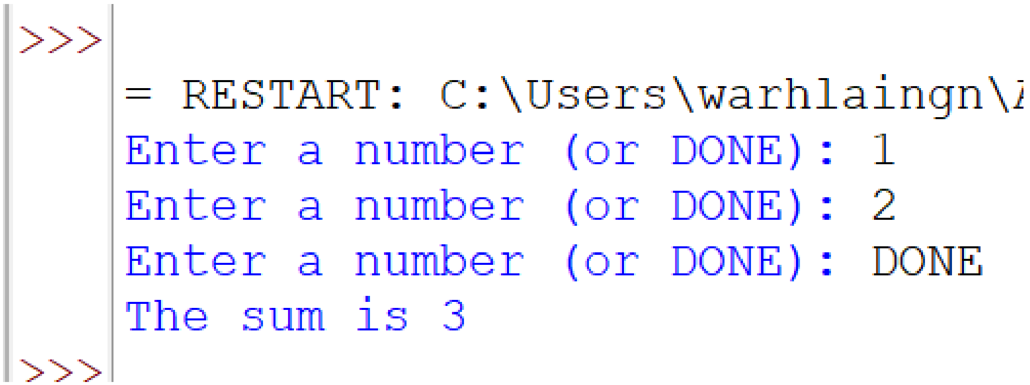


```
total = 0
s = input ('Enter a number (or DONE): '). upper ()
while s!='DONE':
    num = int(s)
    total = total+num
    s = input('Enter a number (or DONE): '). upper ()
print ('The sum is '+ str (total))
```

4. Write a Python program using the **while loop** that performs summing an unknown number of numbers **until the user types 'DONE', then the program end.**

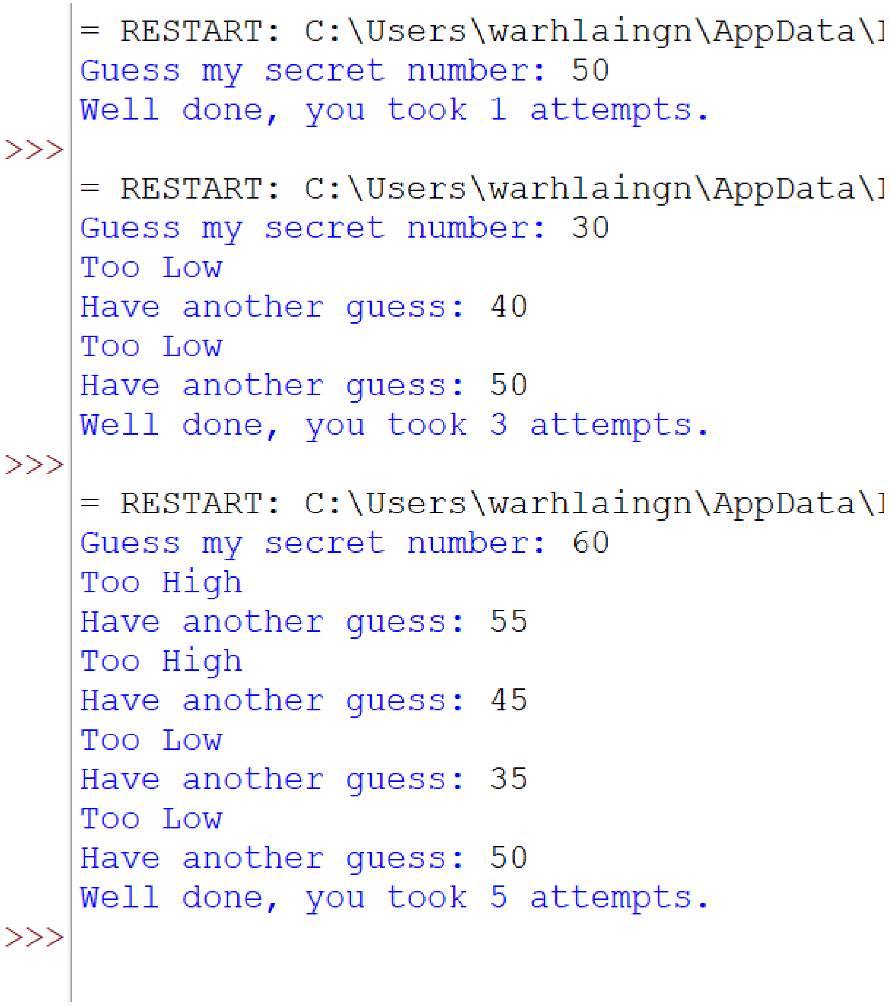


```
total = 0
s = input ('Enter a number (or DONE): '). upper ()
while s!='DONE':
    num = int(s)
    total = total+num
    s = input('Enter a number (or DONE): '). upper ()
print ('The sum is '+ str (total))
```



```
>>>
   = RESTART: C:\Users\warhlaingn\
   Enter a number (or DONE): 1
   Enter a number (or DONE): 2
   Enter a number (or DONE): DONE
   The sum is 3
>>>
```

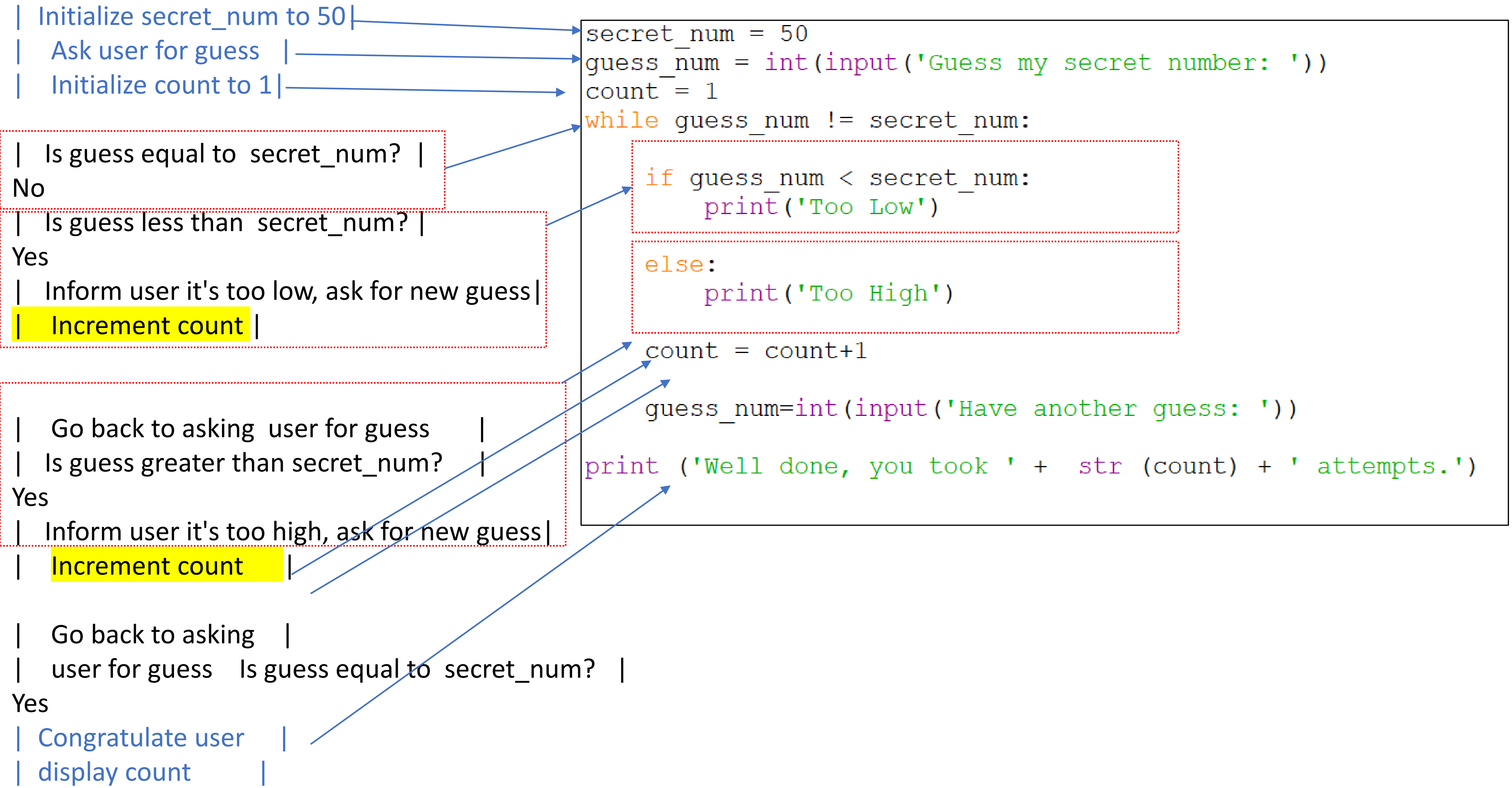
5. Create a variable called **secret_num** and set the value to 50. Ask the user to enter a number. While their guess is not the same as the **secret_num**, tell them if their guess is too low or too high, and ask them to have another guess. If they enter the same value as **secret_num**, display the message 'Well done, you took [**count**] attempts'.



```
    = RESTART: C:\Users\warhlaingn\AppData\
    Guess my secret number: 50
    Well done, you took 1 attempts.
>>> 
    = RESTART: C:\Users\warhlaingn\AppData\
    Guess my secret number: 30
    Too Low
    Have another guess: 40
    Too Low
    Have another guess: 50
    Well done, you took 3 attempts.
>>> 
    = RESTART: C:\Users\warhlaingn\AppData\
    Guess my secret number: 60
    Too High
    Have another guess: 55
    Too High
    Have another guess: 45
    Too Low
    Have another guess: 35
    Too Low
    Have another guess: 50
    Well done, you took 5 attempts.
>>> 
```

```
|  Initialize secret_num to 50|
|    Ask user for guess    |
|    Initialize count to 1|

|   Is guess equal to  secret_num?  |
No
|   Is guess less than  secret_num? |
Yes
|   Inform user it's too low, ask for new guess|
|    Increment count

|    Go back to asking  user for guess       |
|   Is guess greater than secret_num?      |
Yes
|   Inform user it's too high, ask for new guess|
|    Increment count       |

|    Go back to asking     |
|    user for guess     Is guess equal to  secret_num?    |
Yes
|  Congratulate user      |
|  display count          |
```

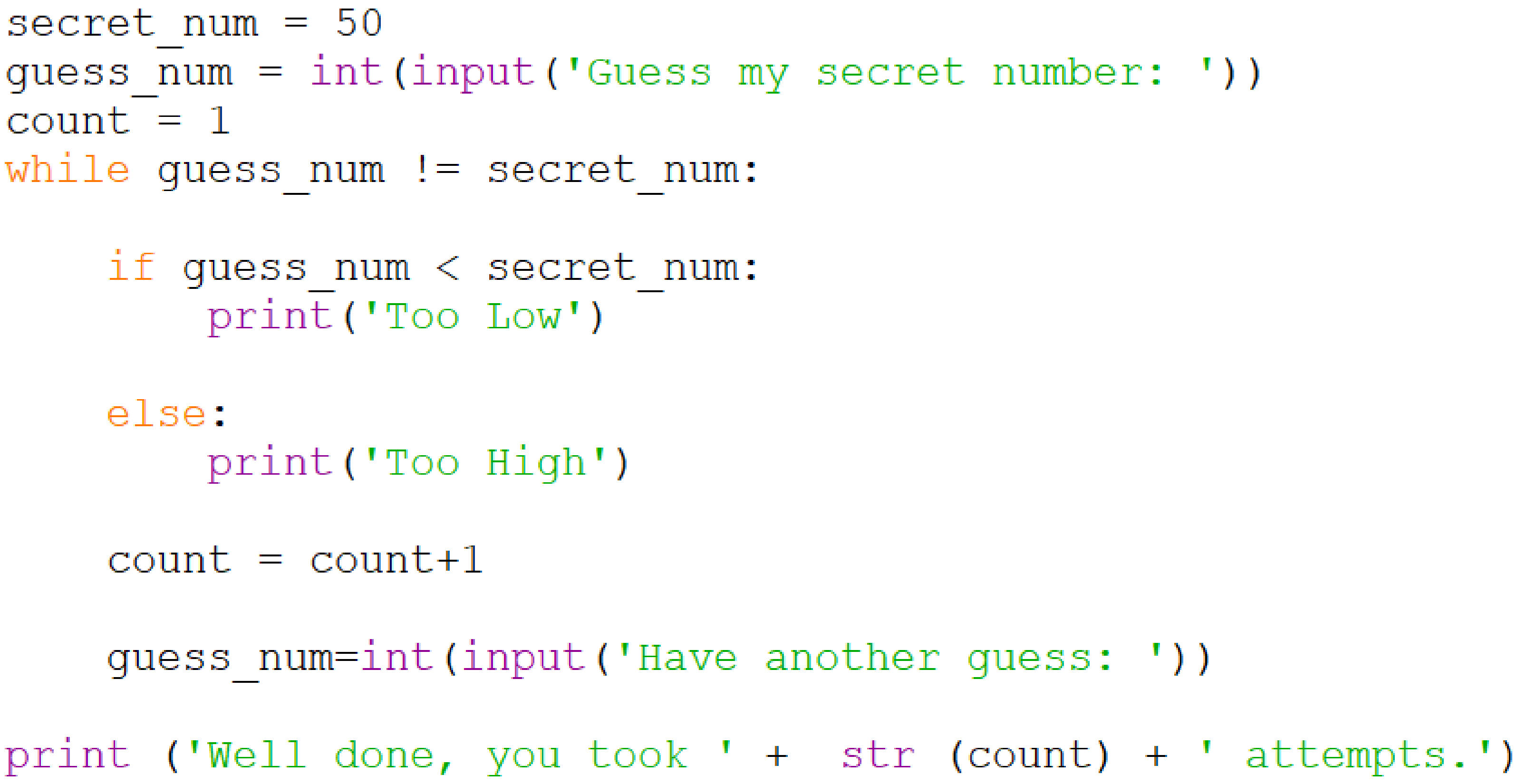
Draw flowchart



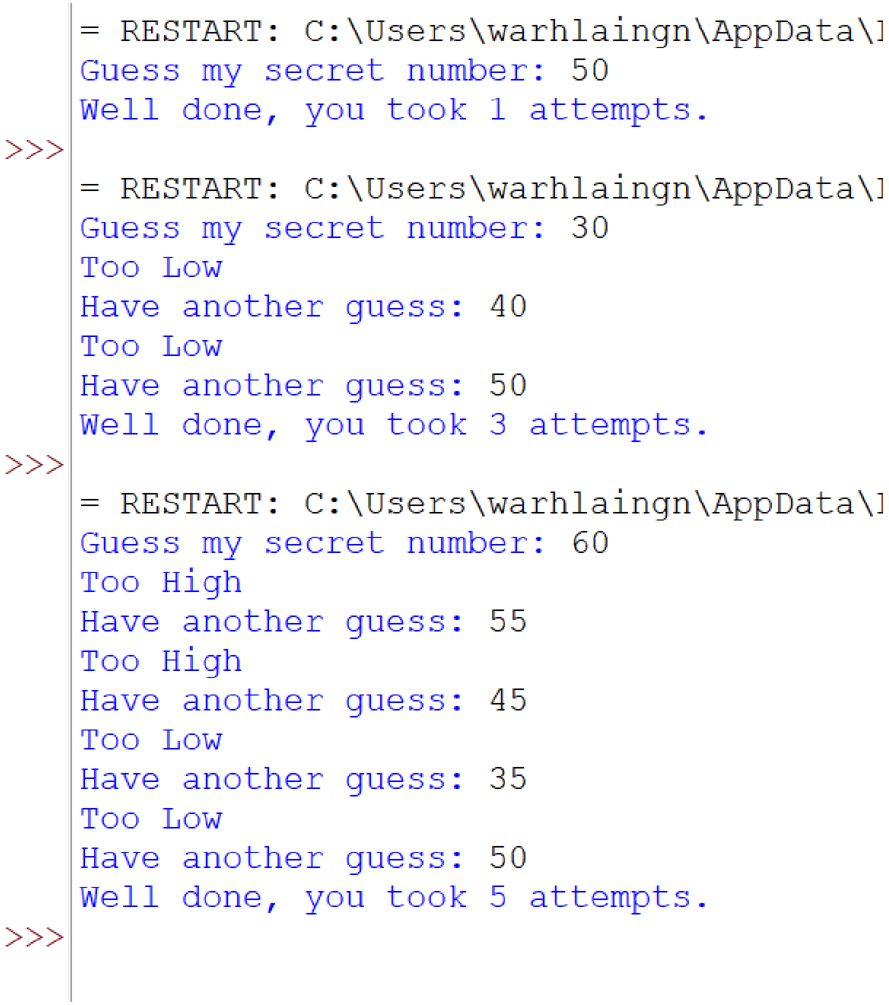
```
| Initialize secret_num to 50|
|   Ask user for guess   |
|   Initialize count to 1|
|   Is guess equal to  secret_num?  |
No
|   Is guess less than  secret_num? |
Yes
|   Inform user it's too low, ask for new guess|
|   Increment count |

|   Go back to asking  user for guess      |
|   Is guess greater than secret_num?      |
Yes
|   Inform user it's too high, ask for new guess|
|   Increment count        |

|   Go back to asking    |
|   user for guess    Is guess equal to  secret_num?   |
Yes
| Congratulate user     |
| display count         |
```

```python
secret_num = 50
guess_num = int(input('Guess my secret number: '))
count = 1
while guess_num != secret_num:
    if guess_num < secret_num:
        print('Too Low')
    else:
        print('Too High')
    count = count+1

    guess_num=int(input('Have another guess: '))
print ('Well done, you took ' +  str (count) + ' attempts.')
```

5. Create a variable called **secret_num** and set the value to 50. Ask the user to enter a number. While their guess is not the same as the **secret_num**, tell them if their guess is too low or too high, and ask them to have another guess. If they enter the same value as **secret_num**, display the message 'Well done, you took [**count**] attempts'.



```
secret_num = 50
guess_num = int(input('Guess my secret number: '))
count = 1
while guess_num != secret_num:

    if guess_num < secret_num:
        print('Too Low')

    else:
        print('Too High')

    count = count+1

    guess_num=int(input('Have another guess: '))

print ('Well done, you took ' +  str (count) + ' attempts.')
```



```
= RESTART: C:\Users\warhlaingn\AppData\
Guess my secret number: 50
Well done, you took 1 attempts.
>>>
= RESTART: C:\Users\warhlaingn\AppData\
Guess my secret number: 30
Too Low
Have another guess: 40
Too Low
Have another guess: 50
Well done, you took 3 attempts.
>>>
= RESTART: C:\Users\warhlaingn\AppData\
Guess my secret number: 60
Too High
Have another guess: 55
Too High
Have another guess: 45
Too Low
Have another guess: 35
Too Low
Have another guess: 50
Well done, you took 5 attempts.
>>>
```