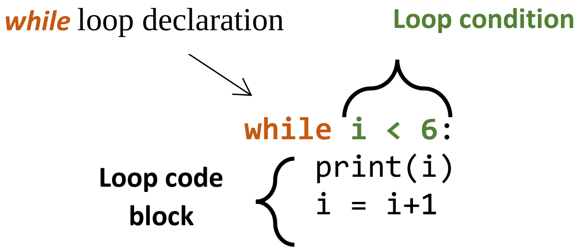
While Loop

Python has two primitive loop commands: for loop and while loop. A *for* loop is a loop of a specific length, whereas a *while* loop is a loop that is used when we do not need the loop anymore.

When a program is in a loop, it performs an operation repeatedly as long as a condition is true. A while loop evaluates the condition and executes the statement if that condition is true. Then it repeats that operation until condition evaluates to false.

The syntax of the *while* loop is as the following:

1. Check the condition is true
2. Execute the code in the while loop block
3. Set the condition again.
4. Repeat from step 1.



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

Counting in while loop

To repeat a task in a specific number of times using a while loop:

1. We will have to setup a counter before the loop
2. Check to see if the counter before and each time we loop.
3. Update the counter each time we loop.

Parts of the while loop

For every while loop to function properly three component must exist:

1. Initialization
2. Comparison: done repeatedly to check for termination condition
3. Update: changes the condition every time, makes loop meaningful

Python Assignment Operators

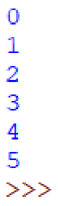
The Python assignment operators are used to assign values to the declared variables

Python Assignment Operators				
Symbol	**Operations**	**Python Example**	**Arithmetic equality**	**Value**
`=`	Assign a value	`x = 5`	`x = 5`	5
`+=`	Self-addition	`x+=3`	`x = x+3`	8
`-=`	Self-subtraction	`x-=3`	`x = x-3`	2
`*=`	Self-multiplication	`x*=3`	`x = x*3`	15
`/=`	Self-division	`x/=3`	`x = x/3`	$1.6\overline{6}$
`//=`	Self-division, returning only the quotient	`x//=3`	`x = x//3`	1
`%=`	Returning remainder of a self-division	`x%=3`	`x = x%3`	2
`**=`	Self-exponential	`x**=3`	`x = x**3`	125

Example) create a while loop that will run as long as the counter ***i*** is less than 6:

```
i=0
while i<6:
    print(i)
    i = i+1  # i +=1
```

Result

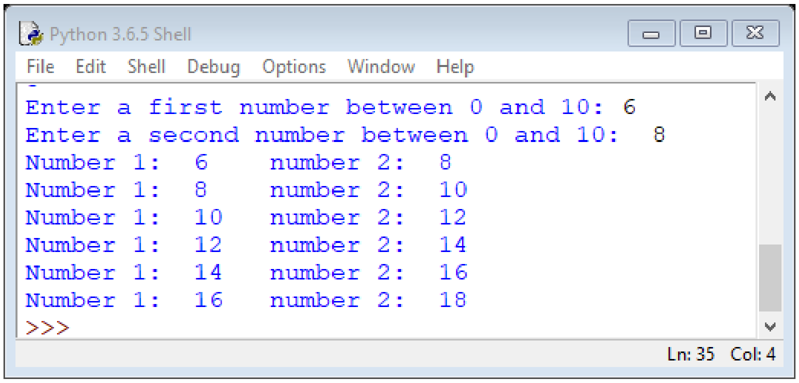


```
0
1
2
3
4
5
>>>
```

Example) Create a python code that will ask the user to Enter two numbers between 0 and 10 and use a while loop to increment each number by 2 until both of them reach up to 25.

```
number1 = int(input("Enter a first number between 0 and 10: "))
number2 = int(input("Enter a second number between 0 and 10:  "))
while number1<20 and number2<20:
    print("Number 1: ", number1, "\t number 2: ", number2)
    number1 += 2
    number2 += 2
```

Result



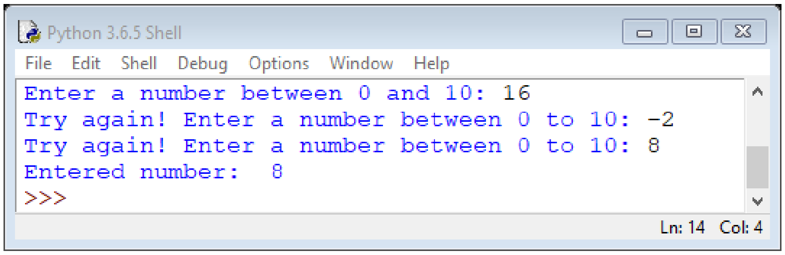
```
Enter a first number between 0 and 10: 6
Enter a second number between 0 and 10:  8
Number 1:  6    number 2:  8
Number 1:  8    number 2:  10
Number 1:  10   number 2:  12
Number 1:  12   number 2:  14
Number 1:  14   number 2:  16
Number 1:  16   number 2:  18
>>>
```

Example) Create a Python code using while loop that will check if a number is within 0 and 10

```
number = int(input("Enter a number between 0 and 10: "))
while number<0 or number>10:
    number = int(input("Try again! Enter a number between 0 to 10: "))

print("Entered number: ", number)
```

Result

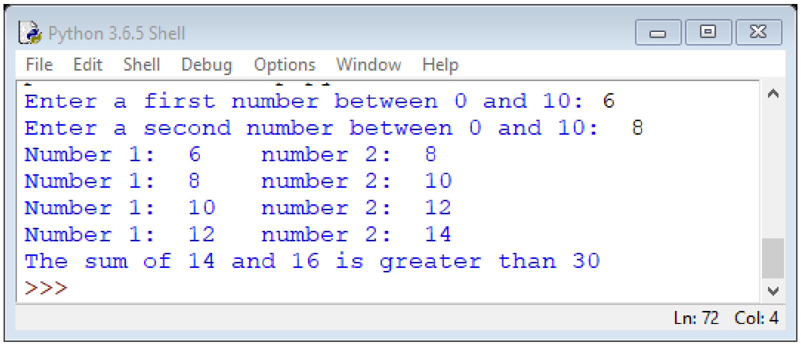


```
Enter a number between 0 and 10: 16
Try again! Enter a number between 0 to 10: -2
Try again! Enter a number between 0 to 10: 8
Entered number:  8
>>>
```

Example) We can have a *while* program from the previous example and the program will stop if the sum of the first and second number is greater than or equal to 30. To run this program, we can have an *if* statement inside the while loop:

```
number1 = int(input("Enter a first number between 0 and 10: "))
number2 = int(input("Enter a second number between 0 and 10:  "))
while number1<20 and number2<20:
    print("Number 1: ", number1, "\t number 2: ", number2)
    number1 += 2
    number2 += 2
    if number1 + number2 >=30:
        print("The sum of %s and %s is greater than 30" %(number1,number2))
        break
```

Result

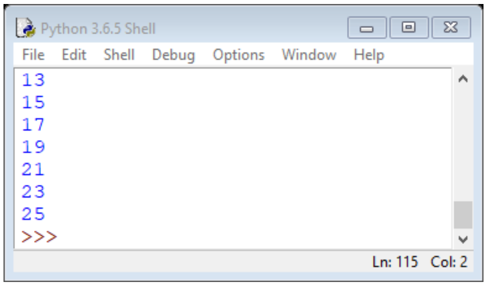


```
Enter a first number between 0 and 10: 6
Enter a second number between 0 and 10:  8
Number 1:  6    number 2:  8
Number 1:  8    number 2:  10
Number 1:  10   number 2:  12
Number 1:  12   number 2:  14
The sum of 14 and 16 is greater than 30
>>>
```

Example) Write a Python program using a while to print all odd numbers between 12 and 25

```
x = 11
while x<=25:
    x += 1
    if x%2==0:
        continue
    print(x)
```

Result



```
13
15
17
19
21
23
25
>>>
```

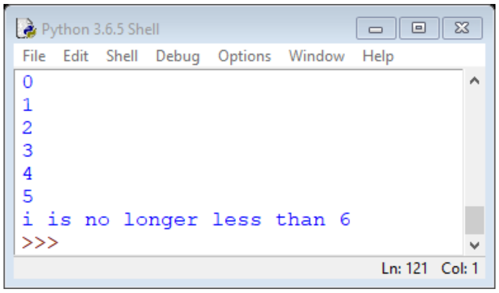
The else statement

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

Example) Use while loop to print all number less than 6 starting from 0 with an increment of 1.

```
i = 0
while i < 6:
    print(i)
    i += 1
else:
    print("i is no longer less than 6")
```

Result



```
0
1
2
3
4
5
i is no longer less than 6
>>>
```