

Loops

Sometimes, it is desirable to repeat the same lines of code using different inputs over and over again. We can use iteration in Python to achieve this efficiently.

There are two main types of loops:

1. Indefinite loop: In indefinite loops, at the beginning we usually don't know how many times the loop will be executed. It is dependent on factors outside the programmer's /program's control. The while construct is usually used for indefinite loops though it is also possible to use it for definite loops.
2. Definite loop: In definite loops, we know upfront the number of times the loop will be executed. The for construct is usually used for definite loops.

While loop

The reserved word `while` is followed by a conditional expression. The variable in the conditional expression is called the iteration variable. The indented set of statements (the block) after the `while` is called the body of the loop. The body is executed repeatedly as long as the condition in the while clause is true.

In [1]:

```
n = 3
while n > 0: # the body of the loop will be executed as long as n > 0
    print(n)
    n -= 1 # this line changes the value of n each time the loop is executed
print('Blastoff')
```

```
3
2
1
Blastoff
```

In []:

```
'''
A correctly defined while loop should change the iteration variable inside the loop
in a manner that ensures that the condition eventually become False.
If not, the loop will continue to execute infinitely until the computer runs out of resources

This is an example of an infinite loop. Select 'Kernel --> Restart and clear output' once you execute this cell to forcibly
terminate the code.
'''
n = 5
while n > 0:
    print(n)

    #Since we increment the value of n each time, n will ALWAYS be greater than zero and hence
    #the loop body will execute infinitely
    n += 1

print('Blastoff')
```

In []:

```
'''
A loop body that will never execute even once is referred to as a dead loop.

This is an example of a dead loop.
'''
n = 5

'''
Note that we begin with n = 5 while the conditional expression requires that n should be greater than 10.
This means the conditional expression will evaluate to false the very first time it is evaluated
and the loop will never be entered
'''
```

and the loop will never be entered.
'''

```
while n > 10:
    print(n)
    n += 1
print('Blastoff')
```

Write a program that asks a user for an integer and prints it out. The program should do this repeatedly as long as the number is greater than 0. This is a clear example of why we call them indefinite loops. We don't know when the user will enter a 0. The first time, we will ask the user for an integer value OUTSIDE the loop. Then we enter the loop and repeatedly execute until the conditional expression evaluates to `False`.

In [2]:

```
#Unless the user enters a negative value the very first time, this code will execute at least once
n = int(input('Enter a positive integer. Enter a value of 0 or less to stop: '))
while n > 0:
    print(n)

    #We ask for the next number. The body of the loop will be executed again, until the user enters 0 or a negative number.
    n = int(input('Enter a positive integer. Enter a value of 0 or less to stop: ')) #

print('done')
```

```
Enter a positive integer. Enter a value of 0 or less to stop: 10
10
Enter a positive integer. Enter a value of 0 or less to stop: 10
10
Enter a positive integer. Enter a value of 0 or less to stop: 5
5
Enter a positive integer. Enter a value of 0 or less to stop: -3
done
```