Implement Gradient Descent in Python

In [1]:
```
cur_x = 2 # The algorithm starts at x=2
rate = 0.01 # Learning rate
precision = 0.000001 #This tells us when to stop the algorithm
previous_step_size = 1 #
max_iters = 10000 # maximum number of iterations
iters = 0 #iteration counter
df = lambda x: 2*(x+3) #Gradient of our function

while previous_step_size > precision and iters < max_iters:
    prev_x = cur_x #Store current x value in prev_x
    cur_x = cur_x - rate * df(prev_x) #Grad descent
    previous_step_size = abs(cur_x - prev_x) #Change in x
    iters = iters+1 #iteration count
    print("Iteration",iters,"\nX value is",cur_x) #Print iterations

print("The local minimum occurs at", cur_x)
```

```
X value is -2.9999415828709818
Iteration 563
X value is -2.999942555213562
Iteration 564
X value is -2.999943704109291
Iteration 565
X value is -2.999944830027105
Iteration 566
X value is -2.999945933426563
Iteration 567
X value is -2.999947014758032
Iteration 568
X value is -2.9999480744628713
Iteration 569
X value is -2.999949112973614
Iteration 570
X value is -2.999950130714142
Iteration 571
X value is -2.999951128099859
The local minimum occurs at -2.999951128099859
```

OUTPUT

In [ ]:
```
From the output Above, we can observe the x values for the first 10
iterations- which can be cross checked with our calculation above. The algorithm
runs for 571 iterations before it terminates.
```