# Gradient Descent Algorithm

In [2]:
```
x=2 #starting point
lr=0.01 #learning rate
precision=0.000001
previous_step_size=1
iter=0
max_iter=10000
gf=lambda x: (x+3)**2 # function
```

In [3]:
```
import matplotlib.pyplot as plt
gd=[]
```

In [4]:
```
while precision < previous_step_size and iter < max_iter:
    prev=x
    x=x-lr*gf(prev)
    previous_step_size=abs(x-prev)
    iter+=1
    print('iteration:',iter,'Value:',x)
    gd.append(x)
```

```
iteration: 372 Value: -2.7468404337431966
iteration: 373 Value: -2.74748133140307
iteration: 374 Value: -2.7481189881829695
iteration: 375 Value: -2.7487534286241093
iteration: 376 Value: -2.749384677020391
iteration: 377 Value: -2.7500127574215125
iteration: 378 Value: -2.7506376936360324
iteration: 379 Value: -2.751259509234384
iteration: 380 Value: -2.751878227551847
iteration: 381 Value: -2.7524938716914753
iteration: 382 Value: -2.753106464526978
iteration: 383 Value: -2.7537160287055618
iteration: 384 Value: -2.7543225866507273
iteration: 385 Value: -2.7549261605650273
iteration: 386 Value: -2.7555267724327814
iteration: 387 Value: -2.7561244440227526
iteration: 388 Value: -2.7567191968907845
iteration: 389 Value: -2.757311052382399
iteration: 390 Value: -2.7579000316353564
iteration: 391 Value: -2.758486155582178
```

In [5]:
```
print('Local Minima',x)
```

```
Local Minima -2.990001240409911
```

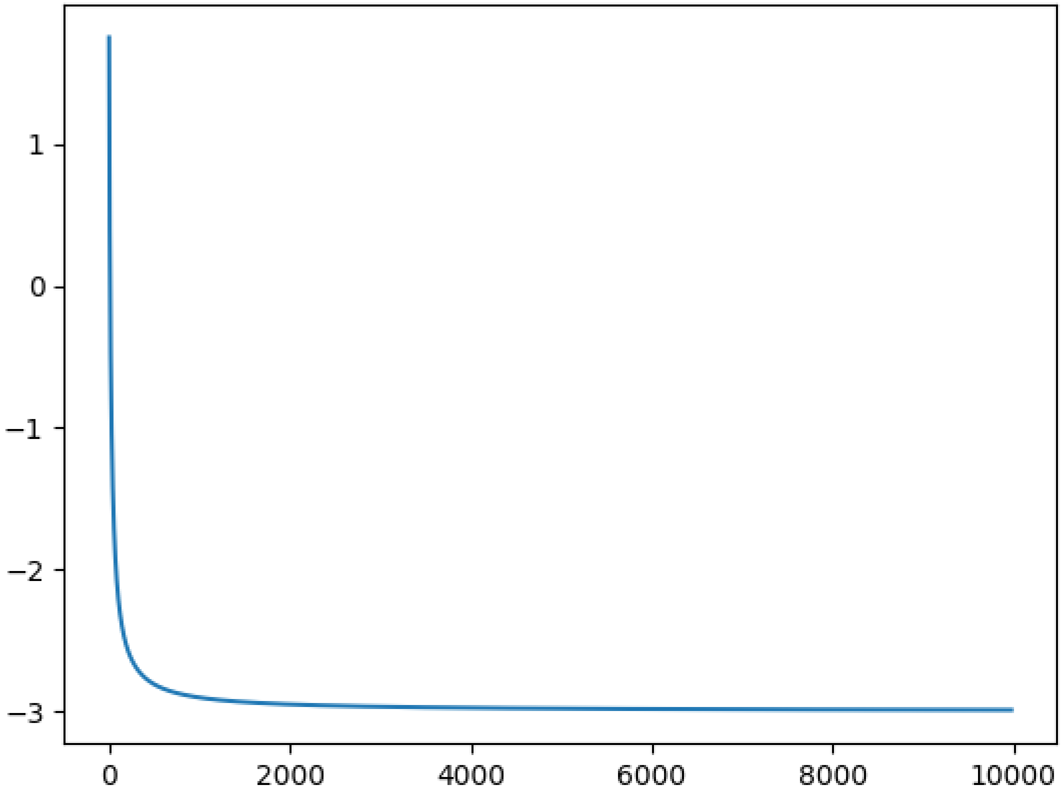
# Gradient Descent

In [7]:

```
plt.plot(gd)
```

Out[7]:

```
[<matplotlib.lines.Line2D at 0x22170af1d20>]
```



In [ ]:

```
```