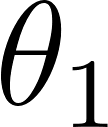
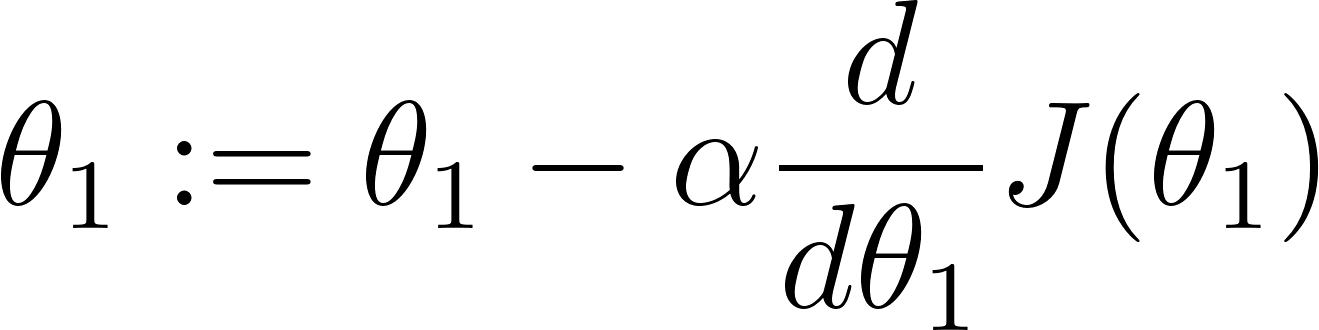
# Gradient Descent Intuition

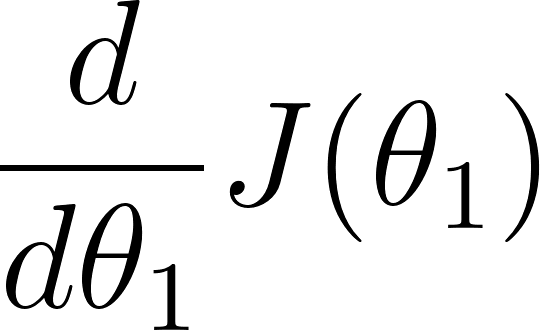
In this video we explored the scenario where we used one parameter

[](https://www.codecogs.com/eqnedit.php?latex=%5Ctheta_1#0) and plotted its cost function to implement a gradient descent. Our formula for a single parameter was :

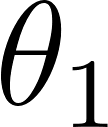
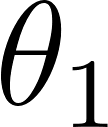
Repeat until convergence:

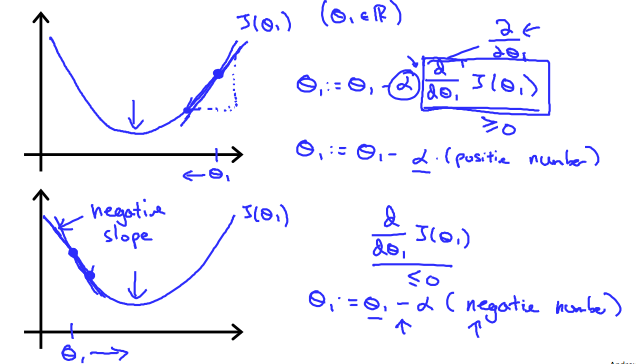
[](https://www.codecogs.com/eqnedit.php?latex=%5Ctheta_1%3A%3D%5Ctheta_1-%5Calpha%20%5Cfrac%7Bd%7D%7Bd%5Ctheta_1%7D%20J(%5Ctheta_1)#0)

Regardless of the slope's sign for

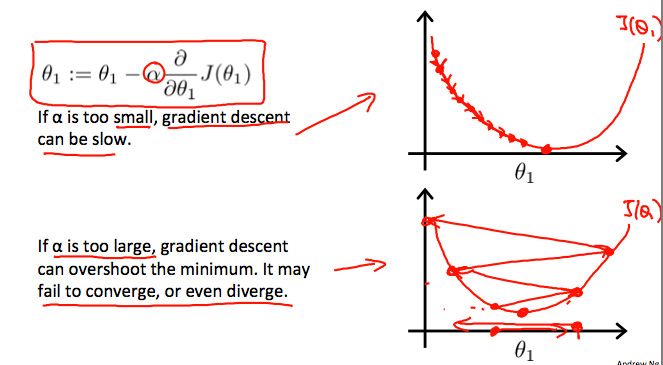
[](https://www.codecogs.com/eqnedit.php?latex=%5Cfrac%7Bd%7D%7Bd%5Ctheta_1%7D%20J(%5Ctheta_1)#0)

eventually converges to its minimum value.

The following graph shows that when the slope is negative, the value of [](https://www.codecogs.com/eqnedit.php?latex=%5Ctheta_1#0) increases and when it is positive, the value of [](https://www.codecogs.com/eqnedit.php?latex=%5Ctheta_1#0) decreases.

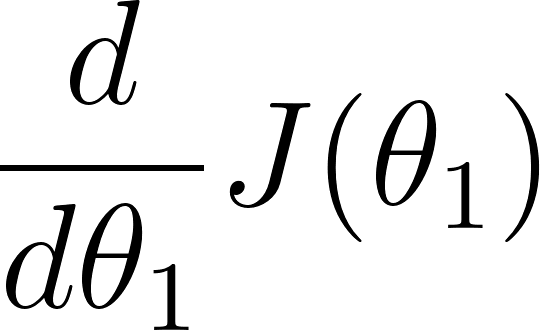


On a side note, we should adjust our parameter [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0) to ensure that the gradient descent algorithm converges in a reasonable time. Failure to converge or too much time to obtain the minimum value imply that our step size is wrong.

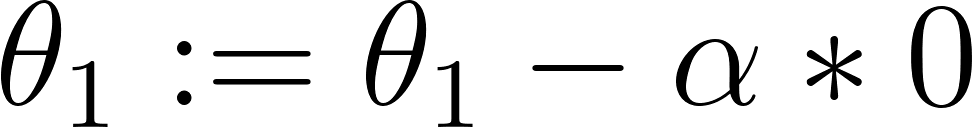


### How does gradient descent converge with a fixed step size ?

The intuition behind the convergence is that

[](https://www.codecogs.com/eqnedit.php?latex=%5Cfrac%7Bd%7D%7Bd%5Ctheta_1%7D%20J(%5Ctheta_1)#0)

​

approaches 0 as we approach the bottom of our convex function. At the minimum, the derivative will always be 0 and thus we get:[](https://www.codecogs.com/eqnedit.php?latex=%5Ctheta_1%3A%3D%5Ctheta_1-%5Calpha%20*%200#0)

