* Gradient Descent :-

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

θ_0, θ_1 ← Parameters of the model

- How to choose θ_i ?

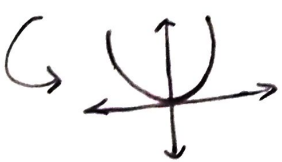
→ Minimize the cost function

So,
Cost function:

$$J_{(\theta_0, \theta_1)} = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right)$$

- $h_\theta(x)$ ← for fixed θ_1, this is a function of x

- $J(\theta_1)$ ← function of parameter θ_1



* Gradient Descent:-

Repeat until Convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

α ← learning Rate

$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ ← Slope of the Cost function

⇒ Simultaneously Update θ_0 & θ_1

$$temp0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$temp1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$\theta_0 = temp0$ ← CORRECT Way

$\theta_1 = temp1$

⇒ Choose α

- α is too low ← Algo will be slow
- α is too High ← Algo will not converge, may diverge

- Gradient descent can converge to a local minimum, even with the learning rate α fixed

Reason :-

As we approach a local minimum gradient descent will automatically take smaller steps (Because the slope will decrease) so no need to decrease α over time.