



- It is what happened after initially choose value of w and b and running the gradient descent also.
- After we reaching to the local minima that is the lowest cost, we get our model that is the func $J_{w,b}$
- Now for any given input we can predict an output.
- Python implementation of linear regression in optional lab.



```
In [ ]: def compute_gradient(x, y, w, b):  
    """  
    Computes the gradient for linear regression  
  
    Args:  
        x (ndarray): Shape (m,) variable such as house size  
        y (ndarray): Shape (m,) target value  
        w (scalar):      Initial values of the parameter  
        b (scalar):      Initial values of the parameter  
  
    Returns:  
        dj_dw (scalar): The gradient of the cost w.r.t. the parameter w.  
        dj_db (scalar): The gradient of the cost w.r.t. the parameter b.  
  
    """  
    m = len(x)  
    dj_dw = 0  
    dj_db = 0  
  
    for i in range(m): # loop over all examples  
        ## calculate prediction for this example  
        f_wb_i = w * x[i] + b  
        ## calculate partial derivatives of cost for this example  
        dj_db_i = f_wb_i - y[i]  
        dj_dw_i = (f_wb_i - y[i]) * x[i]  
        ## add to totals  
        dj_db = dj_db + dj_db_i  
        dj_dw = dj_dw + dj_dw_i
```

→ It shows things in visual that is easy to understand.