

```
Returns
    dj_dw (scalar): The gradient of the cost w.r.t. the parameters w
    dj_db (scalar): The gradient of the cost w.r.t. the parameter b
    """

# Number of training examples
m = x.shape[0]

# You need to return the following variables correctly
dj_dw = 0
dj_db = 0

### START CODE HERE ###
for i in range(m):
    prediction = w * x[i] + b
    dj_db_i = prediction-y[i]
    dj_dw_i = (prediction-y[i])*x[i]
    dj_db += dj_db_i
    dj_dw += dj_dw_i
dj_dw /= m
dj_db /= m

### END CODE HERE ###

return dj_dw, dj_db
```

► Click for hints

Run the cells below to check your implementation of the `compute_gradient` function with two different initializations of the parameters  $w, b$ .

```
In [32]: # Compute and display gradient with w initialized to zeroes
initial_w = 0
initial_b = 0

tmp_dj_dw, tmp_dj_db = compute_gradient(x_train, y_train, initial_w, initial_b)
print('Gradient at initial w, b (zeros):', tmp_dj_dw, tmp_dj_db)

compute_gradient_test(compute_gradient)
```