```python
import numpy as np
import matplotlib.pyplot as plt


def mse(y,y_hat):
    return ((y-y_hat)**2).mean()

def mse_val(y,predict_result):
  return mse(np.array(y),np.array(predict_result))



X = np.array([0.0, 1.0, 2.0])
y = np.array([3.0, 3.5, 5.5])

w = 0        # 기울기
b = 0        # 절편

lrate = 0.01  # 학습률
epochs = 500  # 반복 횟수

n = float(len(X)) # 입력 데이터의 개수
```

```python
# 경사 하강법
j=0
for i in range(epochs):
    y_pred = w*X + b         # 선형 회귀 예측값
    dw = (2/n) * sum(X * (y_pred-y))  # 넘파이 배열간의 산술 계산은 요소별로 적용
    db = (2/n) * sum(y_pred-y)    # sum()은 모든 요소들의 합을 계산하는 내장 함수
    w = w - lrate * dw       # 기울기 수정
    b = b - lrate * db       # 절편 수정
    if (i%50==0):
      print('iteration %3d: loss  %4.2f w %3.2f b %3.2f '%(i,mse(y,y_pred),w,b
))
      j=j+1

# 기울기와 절편을 출력한다.
print ('########## final w,b',w, b)

# 예측값을 만든다.
y_pred = w*X + b


# 입력 데이터를 그래프 상에 찍는다.
plt.scatter(X, y)

# 예측값은 선그래프로 그린다.
plt.plot([min(X), max(X)], [min(y_pred), max(y_pred)], color='red')
plt.show()
```

첫번째 loss 값과 w,b 값을 저에게 채팅으로 보내 주세요.