

나의 모델 만들기 : 모델의 모양 만들기

```
<!DOCTYPE html>
<html>

<head>
  <title>TensorFlow.js Tutorial - lemon</title>

  <!-- Import TensorFlow.js -->
  <script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.0.0/dist/tf.min.js"
></script>

</head>

<body>
  <script>
    // 1. 과거의 데이터를 준비합니다.
    var 온도 = [20,21,22,23];
    var 판매량 = [40,42,44,46];
    var 원인 = tf.tensor(온도);
    var 결과 = tf.tensor(판매량);

    // 2. 모델의 모양을 만듭니다.
    var X = tf.input({ shape: [1] }); //모델에 입력된 값이 몇 개인지 지정하
는 함수 1설정 단 하나의 값이 들어온다
    var Y = tf.layers.dense({ units: 1 }).apply(X); //입력을 정의한 X가
출력을 정의 Y에 apply를 통해서 연결된다 그리고 Y 쪽에 몇개의 값이 출력 될 것인가 1을 설정한
다
    var model = tf.model({ inputs: X, outputs: Y }); //모델의 모양을 정의
하는 부분 이 변수를 통해서 학습이나 예측 작업을 하게 될 것이다
    var compileParam = { optimizer: tf.train.adam(), loss:
tf.losses.meanSquaredError } // 컴파일을 할 때는 입력값을 주어야 한다 optimizer :
효율적으로 모델을 만드는 것 loss : 모델이 얼마나 잘 만들었는지 측정할 때 사용하는 것은 현재로
써 설명하기 어렵다.
    model.compile(compileParam);

    // 3. 데이터로 모델을 학습시킵니다.
    // var fitParam = { epochs: 100}
```

```

        // var fitParam = { epochs: 100, callbacks:
{onEpochEnd:function(epoch, logs){console.log('epoch', epoch, logs);}}}
// loss 추가 예제
        // model.fit(원인, 결과, fitParam).then(function (result) {

                // 4. 모델을 이용합니다.
                // 4.1 기존의 데이터를 이용
                // var 예측한결과 = model.predict(원인);
                // 예측한결과.print();

        // });

        // 4.2 새로운 데이터를 이용
        // var 다음주온도 = [15,16,17, 18, 19]
        // var 다음주원인 = tf.tensor(다음주온도);
        // var 다음주결과 = model.predict(다음주원인);
        // 다음주결과.print();
</script>
</body>

</html>

```

