

선배님께서 잘 보일 것같은 15pt로 작성했습니다!  
잘했죠?

## ▶ 현재 모델의 목적

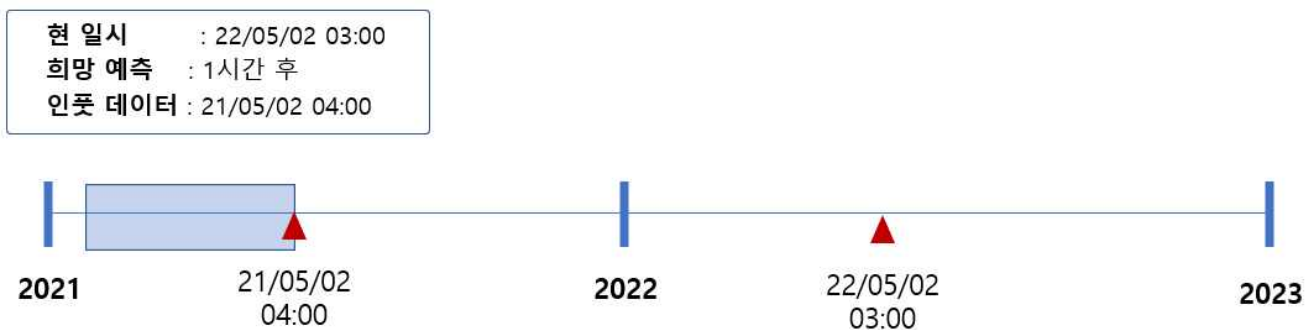
시계열 데이터 200개를 입력받아 온도/습도/조도를 예측

## ▶ 오늘 하고 싶던 거

1시간 후, 2시간 후, 3시간 후의 예측 데이터를 학습된 모델에서 뽑아내자!

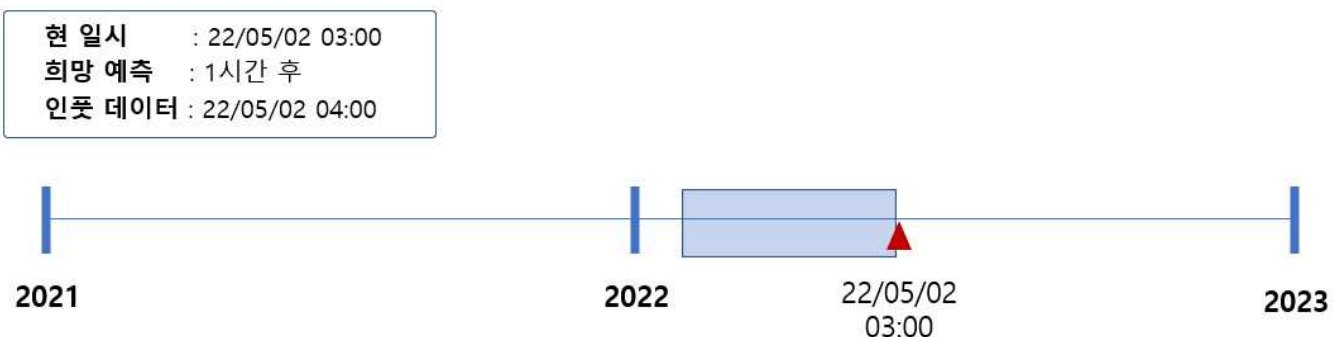
## ▶ 어떤 데이터가 효율적일까?

가설 1) 측정할 일시의 작년의 200개 데이터를 가져오자!



- 지금의 미래 데이터는 없으니까 작년 것을 참고해볼까?

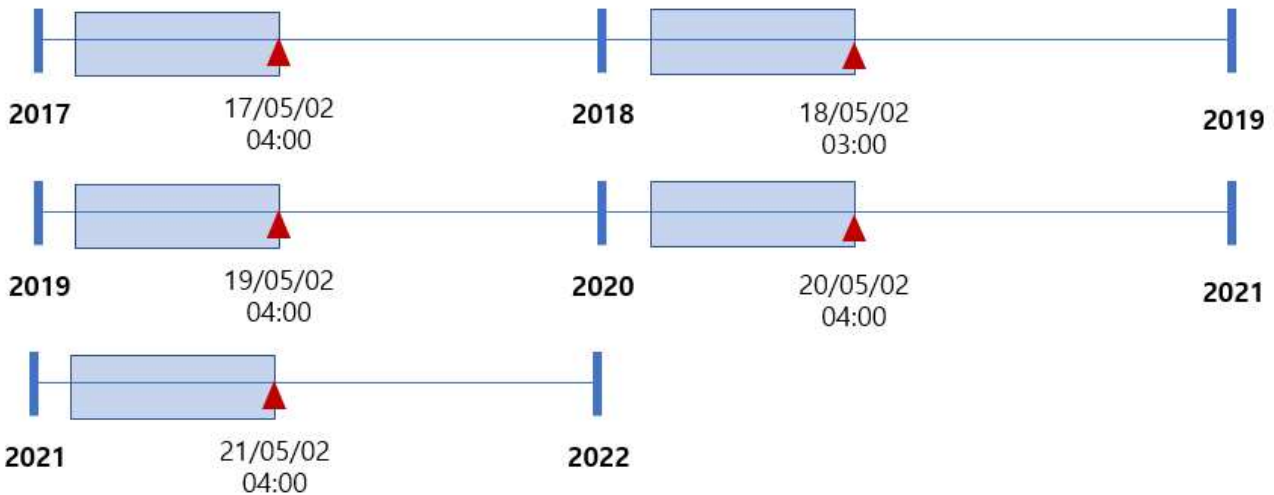
가설 2) 지금 기준으로 200개 데이터를 가져오자!



- 기후변화가 심하잖아! 지금이 훨 좋지!

### 가설 3) 지금으로부터 과거 5년의 데이터 평균을 내보자!

현 일시 : 22/05/02 03:00  
 희망 예측 : 1시간 후  
 인풋 데이터 : 21/05/02 04:00, 20/05/02 04:00, 19/05/02 04:00, 18/05/02 04:00, 17/05/02 04:00



- 평균은 짱 썩니까 좋지 않을까?? 데이터는 다다익선이랬어!

### ▶ 검증

검증 하려면 1시간 기다려야 해서 아직 안 함 시간상 내일 할 듯해요

### ▶ 1시간 후 3시간 후 예측을 어떻게 하지

가설 1) 인풋 데이터를 3시간 간격으로 넣으면 3시간 후 예측이 나오겠지?

희망 예측 시간 : 1시간 후			
시간	온도	습도	조도
2022/08/01 00:00:00	29.5	74	1007.5
2022/08/01 01:00:00	29.6	74	1007.5
2022/08/01 02:00:00	29.5	75	1007.4
2022/08/01 03:00:00	29.6	75	1007.2
2022/08/01 04:00:00	28.2	83	1006.9
2022/08/01 05:00:00	28.6	79	1006.8
...	...	...	...

희망 예측 시간 : 3시간 후			
시간	온도	습도	조도
2022/08/01 00:00:00	29.5	74	1007.5
2022/08/01 01:00:00	29.6	74	1007.5
2022/08/01 02:00:00	29.5	75	1007.4
2022/08/01 03:00:00	29.6	75	1007.2
2022/08/01 04:00:00	28.2	83	1006.9
2022/08/01 05:00:00	28.6	79	1006.8
...	...	...	...

3시간 차이

## ▶ 검증

아하! 나는 똥멍청이구나!

1시간 간격 데이터				3시간 간격 데이터			
Type	Prediction	Real Value	Loss	Type	Prediction	Real Value	Loss
temperature	24.1409	24.3	0.159122	temperature	30.4426	28.6	1.84259
humidity	77.2959	79	1.70409	humidity	62.8805	72	9.11947
illuminance	1013.59	1013.5	0.0897217	illuminance	1003.58	1004.8	1.21533

- 학습된 모델이 1시간 간격 데이터로 학습했는데,
- 3시간 데이터 넣으면 loss값만 커지겠구나!!

## ▶ 결론

- => 각 시간 간격으로 학습한 모델에
- => 각 시간 간격의 데이터를 넣어야겠다!

## ▶ 자랑

```
# 라즈베리 파이의 값을 LSTM을 위해 200개 획득
@app.route('/getIntervalValueForLSTM', methods=['POST'])
def getIntervalValueForLSTM():
    params = request.get_json()
    # year_, month_, day_, hour_, min_, sec_, iter_
    # timeData, temperature, humidity, illuminance
    time, temp, humi, illu = getIntervalData(int(params['year']), int(params['month']), int(params['day']),
                                             int(params['hour']), int(params['min']), int(params['sec']), int(params['iter']))
    json_string = {
        "dateTime": time,
        "temperature": temp,
        "humidity": humi,
        "illuminance": illu
    }

    print("전송 데이터 : ", json_string['dateTime'])
    return jsonify(json_string)
```

- 원하는 시간과, 시간 간격을 넣으면
- 해당 시간 간격으로 데이터를 뽑아주는 API를 만들었어요!
- SQL문을 간만에 짜는게 재미났어요!