

# Remi

2020-2

소프트웨어프로젝트2

## AD프로젝트

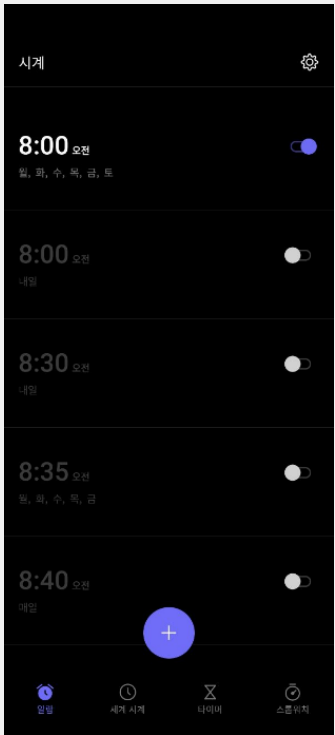
 추헌준  
최진우

# Remi

알리미

Reminder

미리



오늘 < > 2020년 12월 - 2021년 12월			
24	12월, 목	● 종일	크리스마스 이브
25	12월, 금	● 종일	크리스마스
31	12월, 목	● 종일	선달 그믐날
1	1월, 금	● 종일	신정
11	2월, 목	● 종일	설날 연휴
12	2월, 금	● 종일	설날
13	2월, 토	● 종일	설날 연휴
1	3월, 월	● 종일	삼일절
5	4월, 월	● 종일	식목일
5	5월, 수	● 종일	어린이날

Remi

YYYY-MM-DD  
HH:MM:SS

YYYY-MM-DD HH:MM:SS, []요일  
메모

2020-12-13 23:59:59, 일요일  
AD프로젝트 마감일

지난 일정

일정 추가

일정 추가/수정

일정 추가

일정을 추가합니다.

표현하고 싶은 시각을 입력하세요. 현재 시각을 기준으로 합니다.

변환

일정 날짜/시각

YYYY-MM-DD HH:MM:SS

일정 이름

n일 후 반복 (0: 반복 없음)

알림음

DefaultAlarm.wav

찾아보기...

메모

취소

추가

일정 추가/수정

—

□

×

일정 추가

일정을 추가합니다.

표현하고 싶은 시각을 입력하세요. 현재 시각을 기준으로 합니다.

변환

일정 날짜/시각

YYYY-MM-DD HH:MM:SS

일정 이름

n일 후 반복 (0: 반복 없음)

알림음

현재 시각

2020-12-13 09:00:00



입력 문자열

1시간 23분 45초 후



출력 시각

2020-12-13 10:23:45

현재 시각

2020-12-24 09:00:00



입력 문자열

오늘 자정 10분 전



출력 시각

2020-12-24 23:50:00

현재 시각

2020-12-17 09:00:00



입력 문자열

한달뒤 오늘



출력 시각

2020-01-17 09:00:00

입력 문자열

1시간 23분 45초 후

[(1시간 23분 45초) 후]

입력 문자열

오늘 자정 10분 전

[오늘] / [자정] / [(10분) 전]

입력 문자열

한달 뒤 오늘

[(한달) 뒤] / [오늘]



입력 문자열

1시간 23분 45초 후

[(1시간 23분 45초) 후]

입력 문자열

1시간 23분 45초 후

[(1시간 23분 45초) 후]

1시간

23분

45초

후

to\_time(token: str) -&gt; list

```
[|1|-timeObject(1.0)-|0|,  
|0|-timeObject(시간)-|1|,  
|1|-timeObject(23.0)-|0|,  
|0|-timeObject(분)-|1|,  
|1|-timeObject(45.0)-|0|,  
|0|-timeObject(초)-|1|,  
|0|-timeObject(후)-|1|]
```

```
[|1|-timeObject(1.0시간 23.0분 45.0초 후)-|1|]
```

# class timeObject()

```
self.expression
self.frontW
self.backW
self.time_stack
self.name
```

```
def __repr__(self) -> str:
```

```
|{self.frontW}|-timeObject({self.name})-|{self.backW}|
```

```
[|1|-timeObject(1.0)-|0|,
|0|-timeObject(시간)-|1|,
|1|-timeObject(23.0)-|0|,
|0|-timeObject(분)-|1|,
|1|-timeObject(45.0)-|0|,
|0|-timeObject(초)-|1|,
|0|-timeObject(후)-|1|]
```

```
def getTotalW(self) -> int:
```

```
    return self.frontW * self.backW
```

## dataset.py

```
class timeObject():  
    def __init__(self, expression, meta, time_stack, name = ''):
```

```
'초': {  
    'type': 'dependent',  
    'exps': lambda x: timedelta(0, x),  
    'meta': (0, 1),  
    'time_stack': True  
},  
'분': {  
    'type': 'dependent',  
    'exps': lambda x: timedelta(0, 60 * x),  
    'meta': (0, 1),  
    'time_stack': True  
},  
'시간': {  
    'type': 'dependent',  
    'exps': lambda x: timedelta(0, 3600 * x),  
    'meta': (0, 1),  
    'time_stack': True  
},  
'후': {  
    'type': 'dependent',  
    'exps': lambda x: (lambda t: t + x),  
    'meta': (0, 1),  
    'time_stack': False  
},
```

```
[|1|-timeObject(1.0)-|0|,  
 |0|-timeObject(시간)-|1|,  
 |1|-timeObject(23.0)-|0|,  
 |0|-timeObject(분)-|1|,  
 |1|-timeObject(45.0)-|0|,  
 |0|-timeObject(초)-|1|,  
 |0|-timeObject(후)-|1|]
```

# parse\_time()

```
class timeObject():
```

```
self.expression
self.frontW
self.backW
self.time_stack
self.name
```

```
[|1|-timeObject(1.0)-|0|,
 |0|-timeObject(시간)-|1|,
 |1|-timeObject(23.0)-|0|,
 |0|-timeObject(분)-|1|,
 |1|-timeObject(45.0)-|0|,
 |0|-timeObject(초)-|1|,
 |0|-timeObject(후)-|1|]
```

```
for i in range(len(translated) - 1):
```

```
[|1|-timeObject(1.0)-|0|,
 |0|-timeObject(시간)-|1|,
 |1|-timeObject(23.0)-|0|,
 |0|-timeObject(분)-|1|,
 |1|-timeObject(45.0)-|0|,
 |0|-timeObject(초)-|1|]
```

```
|1|-timeObject(1.0시간 23.0분 45.0초)-|0|
```

```
|0|-timeObject(후)-|1|
```

```
|0|-timeObject(후)-|1|
```

```
[|1|-timeObject(1.0시간 23.0분 45.0초 후)-|1|]
```

현재 시각

2020-12-13 09:00:00



1시간 23분 45초 후

[|1|-timeObject(1.0시간 23.0분 45.0초 후)-|1|]

출력 시각

2020-12-13 10:23:45

