

[3-1. cafle 라이브러리 소개]

1. "cafle" 라이브러리 소개

1) 재무모델 작성에 활용할 목적으로 작성된 파이썬 라이브러리

- 기존 엑셀로 작성하던 재무모델을 파이썬으로 작성할 수 있도록 여러가지 모듈과 함수를 제공.
- pandas를 기반으로 모듈과 함수들이 작성되어 있으며 결과물을 모두 pandas.DataFrame으로 출력하므로, pandas에 대한 이해가 선행되어야 함.
- 주요 모듈 : Account, Index, Write

2) 주요 모듈

- **Account** : 각각의 개별 현금흐름의 내용을 저장하고, 현금흐름의 유출입을 관리하는 모듈.
- **Index** : 날짜 등 현금흐름의 시간적인 요소를 컨트롤하기 위한 모듈.
- **Write** : 최종적으로 작성된 현금흐름을 엑셀 파일 상에 출력해주는 모듈.

3) 재무모델 작성 개요

1. **기간설정** : 사업 기간, 조달된 자금의 만기 등을 Index 모듈에 설정.
2. **Account 설정** : 자금의 조달, 매출, 사업비 등 사업의 주요 요소들 별로 재무모델 가정에 따른 조건들을 Account 모듈에 설정.
3. **현금흐름 실행** : Index에 설정된 시간 흐름에 따라 각각의 요소들 간 사업 진행에 따라 발생하는 현금 유출입을 Account 내에 기록.
4. **최종 결과물 출력** : Account 내에 기록된 현금흐름을 최종적으로 확인하여 최종적인 현금흐름 결과물을 Excel파일로 추출.

In []:

2. cafle 라이브러리의 설치

- jupyter notebook에서 아래 구문을 이용하여 설치.
- 코드 작성 후 "shift" + "Enter"를 이용해 코드를 실행함.

```
! pip install cafle
```

```
In [4]: ! pip install cafle
```

```
Collecting cafle
  Using cached cafle-0.2.4-py3-none-any.whl (27 kB)
Installing collected packages: cafle
Successfully installed cafle-0.2.4
```

```
In [8]: import cafle
```

```
In [9]: cafle
```

```
Out[9]: <module 'cafle' from '/Users/KP_Hong/opt/anaconda3/envs/FastCampus2/lib/python3.7/site-packages/cafle/__init__.py'>
```

```
In [ ]:
```

- "import cafle"을 입력하여 cafle 라이브러리를 임포트 한 후, cafle을 입력하면 cafle에 대한 기초 내용이 출력됨.

```
In [ ]:
```

3. cafle импорт 하기

```
In [1]: import cafle
```

```
In [2]: idx = cafle.Index('2023.01', 12)
        idx
```

```
Out[2]: DateIndex(['2023.01.31', '2023.02.28', '2023.03.31', '2023.04.30',
                  '2023.05.31', '2023.06.30', '2023.07.31', '2023.08.31', '2023.09.30',
                  '2023.10.31', '2023.11.30', '2023.12.31'])
```

```
In [3]: acc = cafle.Account(idx)
        acc
```

```
Out[3]: Account(main, len 12)
```

```
In [4]: acc.df
```

```
Out[4]:
```

	bal_strt	amt_in	amt_out	bal_end
2023-01-31	0.0	0.0	0.0	0.0
2023-02-28	0.0	0.0	0.0	0.0
2023-03-31	0.0	0.0	0.0	0.0
2023-04-30	0.0	0.0	0.0	0.0
2023-05-31	0.0	0.0	0.0	0.0
2023-06-30	0.0	0.0	0.0	0.0
2023-07-31	0.0	0.0	0.0	0.0
2023-08-31	0.0	0.0	0.0	0.0
2023-09-30	0.0	0.0	0.0	0.0
2023-10-31	0.0	0.0	0.0	0.0
2023-11-30	0.0	0.0	0.0	0.0
2023-12-31	0.0	0.0	0.0	0.0

```
In [5]: type(acc.df)
```

```
Out[5]: pandas.core.frame.DataFrame
```

```
In [6]: import cafle

idx = cafle.Index('2023.01', 12)
idx

acc = cafle.Account(idx)
acc

acc.df

type(acc.df)
```

```
Out[6]: pandas.core.frame.DataFrame
```

```
In [ ]:
```

```
In [7]: import cafle as cf
```

```
In [8]: idx = cf.Index("2023.01", 12)
        idx
```

```
Out[8]: DateIndex(['2023.01.31', '2023.02.28', '2023.03.31', '2023.04.30',
                  '2023.05.31', '2023.06.30', '2023.07.31', '2023.08.31', '2023.09.30',
                  '2023.10.31', '2023.11.30', '2023.12.31'])
```

```
In [9]: acc = cf.Account(idx)
        acc
```

```
Out[9]: Account(main, len 12)
```

```
In [10]: acc.df
```

```
Out[10]:
```

	bal_strt	amt_in	amt_out	bal_end
2023-01-31	0.0	0.0	0.0	0.0
2023-02-28	0.0	0.0	0.0	0.0
2023-03-31	0.0	0.0	0.0	0.0
2023-04-30	0.0	0.0	0.0	0.0
2023-05-31	0.0	0.0	0.0	0.0
2023-06-30	0.0	0.0	0.0	0.0
2023-07-31	0.0	0.0	0.0	0.0
2023-08-31	0.0	0.0	0.0	0.0
2023-09-30	0.0	0.0	0.0	0.0
2023-10-31	0.0	0.0	0.0	0.0
2023-11-30	0.0	0.0	0.0	0.0
2023-12-31	0.0	0.0	0.0	0.0

```
In [ ]:
```

```
In [11]: from cafle import Index, Account
```

```
In [12]: idx = Index('2023.01', 12)
        idx
```

```
Out[12]: DateIndex(['2023.01.31', '2023.02.28', '2023.03.31', '2023.04.30',
                  '2023.05.31', '2023.06.30', '2023.07.31', '2023.08.31', '2023.09.30',
                  '2023.10.31', '2023.11.30', '2023.12.31'])
```

```
In [13]: acc = Account(idx)
acc
```

```
Out[13]: Account(main, len 12)
```

```
In [14]: acc.df
```

```
Out[14]:
```

	bal_strt	amt_in	amt_out	bal_end
2023-01-31	0.0	0.0	0.0	0.0
2023-02-28	0.0	0.0	0.0	0.0
2023-03-31	0.0	0.0	0.0	0.0
2023-04-30	0.0	0.0	0.0	0.0
2023-05-31	0.0	0.0	0.0	0.0
2023-06-30	0.0	0.0	0.0	0.0
2023-07-31	0.0	0.0	0.0	0.0
2023-08-31	0.0	0.0	0.0	0.0
2023-09-30	0.0	0.0	0.0	0.0
2023-10-31	0.0	0.0	0.0	0.0
2023-11-30	0.0	0.0	0.0	0.0
2023-12-31	0.0	0.0	0.0	0.0

```
In [15]: acc.addamt('2023.01', 30_000)
acc.subamt('2023.04', 10_000)
acc.subamt('2023.05', 10_000)
```

In [16]: `acc.df`

Out[16]:

	bal_strt	amt_in	amt_out	bal_end
2023-01-31	0.0	30000.0	0.0	30000.0
2023-02-28	30000.0	0.0	0.0	30000.0
2023-03-31	30000.0	0.0	0.0	30000.0
2023-04-30	30000.0	0.0	10000.0	20000.0
2023-05-31	20000.0	0.0	10000.0	10000.0
2023-06-30	10000.0	0.0	0.0	10000.0
2023-07-31	10000.0	0.0	0.0	10000.0
2023-08-31	10000.0	0.0	0.0	10000.0
2023-09-30	10000.0	0.0	0.0	10000.0
2023-10-31	10000.0	0.0	0.0	10000.0
2023-11-30	10000.0	0.0	0.0	10000.0
2023-12-31	10000.0	0.0	0.0	10000.0

In []: