

[3-1. 자금의 조달 및 소요 설정]

1. assumption 데이터 확인 및 기본 설정 진행

1-1. 필요한 모듈 import

- 먼저 작성한 FSmodeling_assumption.xlsx 파일, m00_general_function.py 파일, m01_assumption.py 파일, m02_index.py 파일을 폴더에 복사 이동
- 모듈 내에서 필요한 객체들을 import

```
In [ ]: import pandas as pd

        from m01_assumption import assumption
        from m02_index import index
```

1-2. assumption 데이터 확인

```
In [ ]: assumption['business_overview']
```

```
Out [ ]: {'사업개요': {'사업내용': '호텔건물 매입 및 운영 사업',
                      '건물명': 'FS호텔',
                      '주소': '서울시 강남구 테헤란로',
                      '건물규모': '지하1층/지상10층'},
          '객실수': {'TypeA': 40, 'TypeB': 40, 'TypeC': 20},
          '자산매입': {'자산매입금액': 180000000000, '매입부수비용': 10000000000}}
```

```
In [ ]: assumption['period_assumptions']
```

```
Out [ ]: {'기본기간가정': {'모델시작일': datetime.datetime(2023, 12, 1, 0, 0),
  '모델종료일': datetime.datetime(2027, 1, 31, 0, 0),
  '운영시작일': datetime.datetime(2024, 1, 1, 0, 0),
  '운영종료일': datetime.datetime(2026, 12, 31, 0, 0)},
  '자산매입일정': {'자산매입일': datetime.datetime(2023, 12, 31, 0, 0),
  '매입부수비용지급일': datetime.datetime(2023, 12, 31, 0, 0)},
  '자금조달일정': {'자기자본유입일': datetime.datetime(2023, 12, 31, 0, 0),
  '차입금유입일': datetime.datetime(2023, 12, 31, 0, 0),
  '이자지급시작일': datetime.datetime(2024, 1, 1, 0, 0),
  '이자지급종료일': datetime.datetime(2026, 12, 31, 0, 0),
  '원금상환시작일': datetime.datetime(2024, 1, 1, 0, 0),
  '원금상환종료일': datetime.datetime(2026, 12, 31, 0, 0)},
  '연간인상률': {'판매단가': 0.05, '운영비': 0.03, '인건비': 0.05}}
```

```
In [ ]: assumption['funding_assumptions']
```

```
Out [ ]: {'equity': {'amount': 10000000000,
  '최초배당일': datetime.datetime(2025, 12, 31, 0, 0),
  '배당률': 0.3},
  'loan': {'amount': 10000000000,
  '원금상환주기': 3,
  '상환원금': 50000000,
  '이자지급주기': 1,
  '이자율': 0.05}}
```

1-3. 빈 funding 딕셔너리 설정

```
In [ ]: funding = {}
```

2. 자기자본 설정

- index['model'] 기간 중 "자기자본유입일"에 해당하는 날짜에는 해당 금액(equity.amount)을 기록
- 그 외의 경우에는 0을 기록
- 추후 현금흐름 작업 시 배당금을 기록할 수 있도록 미리 "배당금지급" 컬럼에 0으로 기록

```
In [ ]: data = []
for dt in index['model']:
    dct = {}
    if dt == assumption['period_assumptions']['자금조달일정']['자기자본유입일']:
        dct['자기자본유입'] = assumption['funding_assumptions']['equity']['amount']
    else:
        dct['자기자본유입'] = 0

    dct['배당금지급'] = 0

    data.append(dct)
funding['자기자본'] = pd.DataFrame(data, index=index['model'])
```

3. 차입금 설정

- index['model'] 기간 중에 "차입금유입일"에 맞춰서 차입금 유입금액을 기록
- 원금상환주기와 이자지급주기에 맞춰서 원금상환금액과 이자지급금액을 기록

```
In [ ]: 차입금잔액 = 0
data = []
for dt in index['model']:
```

```

dct = {}

# 차입금 유입금액 설정
if dt == assumption['period_assumptions']['자금조달일정']['차입금유입일']:
    차입금유입 = assumption['funding_assumptions']['loan']['amount']
else:
    차입금유입 = 0

# 3개월 주기 원금상환 설정
if (
    (dt in index['원금상환'])
    and (
        index['원금상환'].index(dt) %
        assumption['funding_assumptions']['loan']['원금상환주기'] == 2
    )
):
    차입금상환 = assumption['funding_assumptions']['loan']['상환원금']
else:
    차입금상환 = 0

# 매월 이자 지급 설정
if dt in index['이자지급']:
    이자율 = assumption['funding_assumptions']['loan']['이자율']
    월간일수 = index['days'].loc[dt, '월간일수']
    연간일수 = index['days'].loc[dt, '연간일수']
    # 이자금액 계산
    차입이자 = int(round(차입금잔액 * 이자율 * 월간일수 / 연간일수, -1))
else:
    차입이자 = 0

차입금잔액 = 차입금잔액 + 차입금유입 - 차입금상환

dct['차입금유입'] = 차입금유입
dct['차입금상환'] = 차입금상환
dct['차입이자'] = 차입이자

```

```
dct['차입금잔액'] = 차입금잔액

data.append(dct)
funding['차입금'] = pd.DataFrame(data, index=index['model'])
```

4. 자산매입 설정

- 자산매입일과 매입부수비용지급일에 매칭하여 재산매입금액과 매입부수비용 기록

```
In [ ]: data = []
for dt in index['model']:
    dct = {}
    if dt == assumption['period_assumptions']['자산매입일정']['자산매입일']:
        dct['자산매입'] = assumption['business_overview']['자산매입']['자산매입금액']
    else:
        dct['자산매입'] = 0

    if dt == assumption['period_assumptions']['자산매입일정']['매입부수비용지급일']:
        dct['매입부수비용'] = assumption['business_overview']['자산매입']['매입부수비용']
    else:
        dct['매입부수비용'] = 0

    data.append(dct)
funding['자산매입'] = pd.DataFrame(data, index=index['model'])
```

5. funding.py 파일 작성

```
In [ ]: # m03_funding.py

import pandas as pd

from m01_assumption import assumption
```

```
from m02_index import index

#### 1. 빈 funding 딕셔너리 설정
funding = {}

#### 2. 자기자본 설정
data = []
for dt in index['model']:
    dct = {}
    if dt == assumption['period_assumptions']['자금조달일정']['자기자본유입일']:
        dct['자기자본유입'] = assumption['funding_assumptions']['equity']['amount']
    else:
        dct['자기자본유입'] = 0

    dct['배당금지급'] = 0

    data.append(dct)
funding['자기자본'] = pd.DataFrame(data, index=index['model'])

#### 3. 차입금 설정
차입금잔액 = 0
data = []
for dt in index['model']:
    dct = {}

    # 차입금 유입금액 설정
    if dt == assumption['period_assumptions']['자금조달일정']['차입금유입일']:
        차입금유입 = assumption['funding_assumptions']['loan']['amount']
    else:
        차입금유입 = 0

    # 3개월 주기 원금상환 설정
```

```

if (
    (dt in index['원금상환'])
    and (
        index['원금상환'].index(dt) %
        assumption['funding_assumptions']['loan']['원금상환주기'] == 2
    )
):
    차입금상환 = assumption['funding_assumptions']['loan']['상환원금']
else:
    차입금상환 = 0

# 매월 이자 지급 설정
if dt in index['이자지급']:
    이자율 = assumption['funding_assumptions']['loan']['이자율']
    월간일수 = index['days'].loc[dt, '월간일수']
    연간일수 = index['days'].loc[dt, '연간일수']
    # 이자금액 계산
    차입이자 = int(round(차입금잔액 * 이자율 * 월간일수 / 연간일수, -1))
else:
    차입이자 = 0

차입금잔액 = 차입금잔액 + 차입금유입 - 차입금상환

dct['차입금유입'] = 차입금유입
dct['차입금상환'] = 차입금상환
dct['차입이자'] = 차입이자
dct['차입금잔액'] = 차입금잔액

data.append(dct)
funding['차입금'] = pd.DataFrame(data, index=index['model'])

#### 4. 자산매입 설정
data = []
for dt in index['model']:

```

```
dct = {}
if dt == assumption['period_assumptions']['자산매입일정']['자산매입일']:
    dct['자산매입'] = assumption['business_overview']['자산매입']['자산매입금액']
else:
    dct['자산매입'] = 0

if dt == assumption['period_assumptions']['자산매입일정']['매입부수비용지급일']:
    dct['매입부수비용'] = assumption['business_overview']['자산매입']['매입부수비용']
else:
    dct['매입부수비용'] = 0

data.append(dct)
funding['자산매입'] = pd.DataFrame(data, index=index['model'])
```