생산계획 IE Challenge
MPS-MRP 시스템 구축

MRP SYSTEM BUILDING
LEGO BIONICLE

2018170804 박민규
2018170806 엄기영
2018170840 윤승주

PRESENTATION

CONTENT

01. Bill Of Materials (BOM)

The background (Firms) setting with BOM

02. Rules and Assumptions

Defining Rules and Assumptions for production planning and environment

03. System with Simulations

An Example of the SOP, MPS and MRP

04. Code and Execution

The Code of the SOP, MPS and MRP

The background (Firms) setting with BOM

01. BILL OF MATERIALS

- LEGO사 에서는 고객들의 성원에 힘입어 과거 출시했던 인간형 장난감 BIONICLE을 재출시하기로 결정
- 전상품을 모두 재출시하기 전, 과거 가장 판매량이 높았던 세개의 제품만을 우선 시범적으로 생산 및 판매하기로 함

NAME: JIMI STRINGER

Prod. num : BN-008

NAME: NIDHIKI

Prod. num : BN-011

NAME: NUJU

Prod. num : BN-012

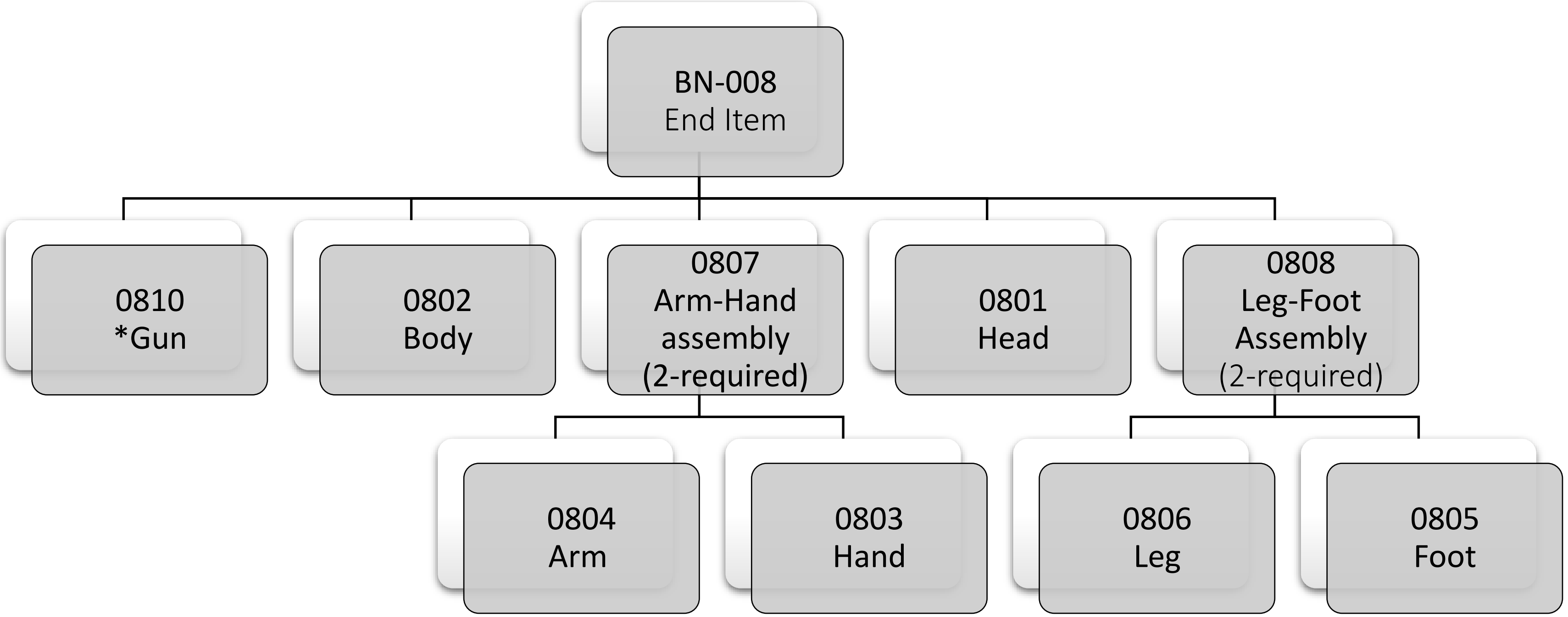
The background (Firms) setting with BOM

01. BILL OF MATERIALS

NAME: JIMI STRINGER

Prod. num : BN-008

Product Structure Diagram



* 는 Special parts로 자체생산이 아닌 vendor에게 공급받는다

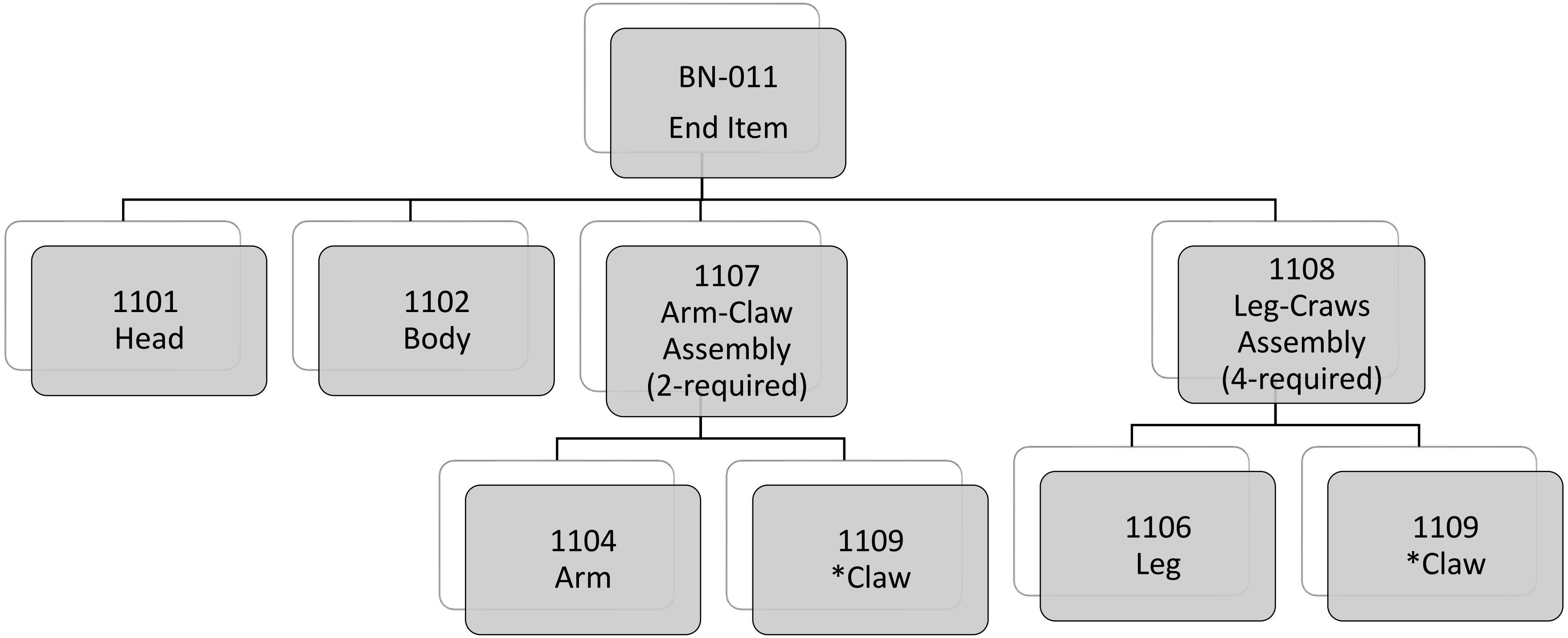
The background (Firms) setting with BOM

01. BILL OF MATERIALS

NAME: NIDHIKI

Prod. num : BN-011

Product Structure Diagram



* 는 Special parts로 자체생산이 아닌 vendor에게 공급받는다

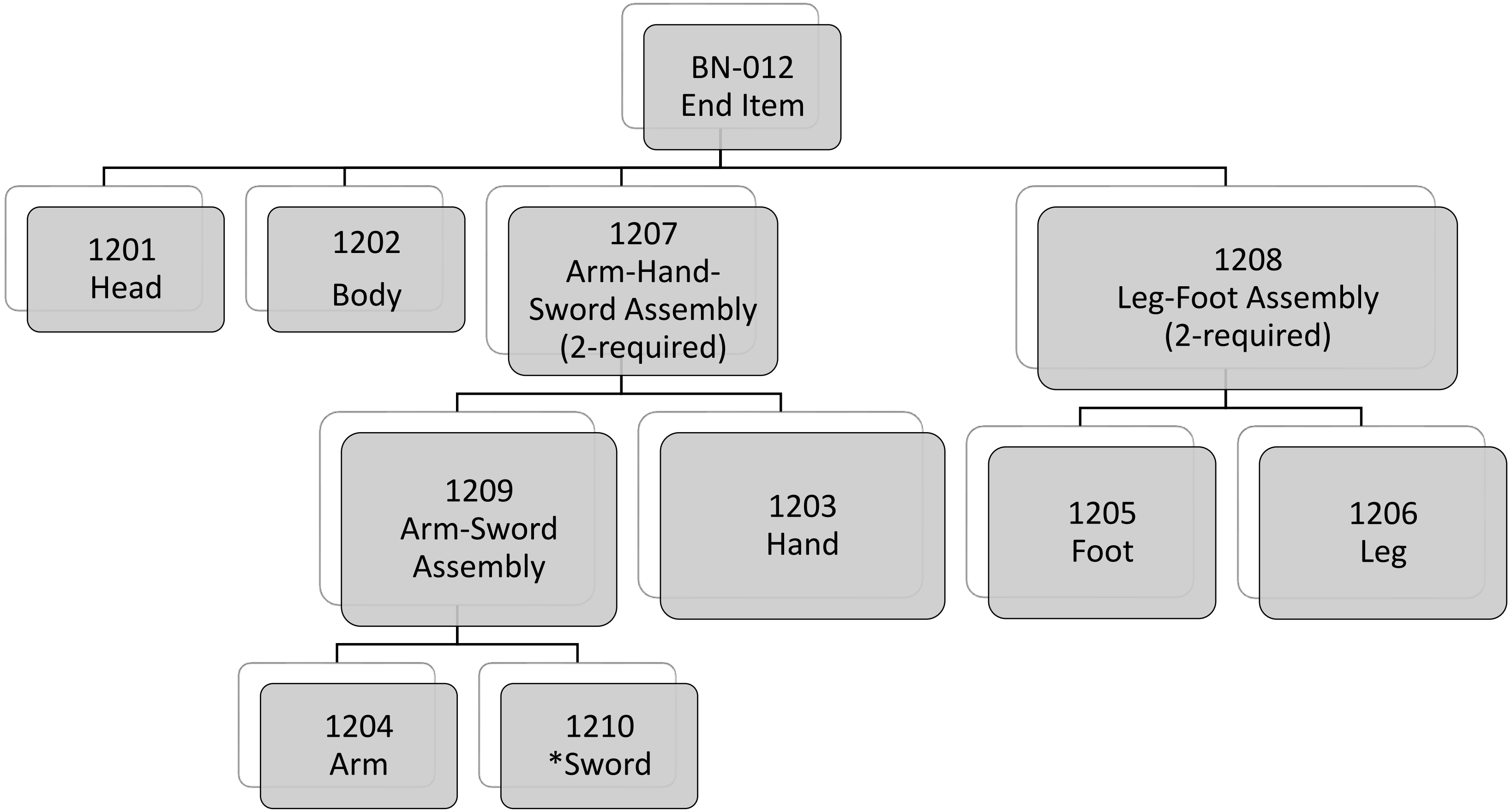
The background (Firms) setting with BOM

01. BILL OF MATERIALS

NAME: NUJU

Prod. num : BN-012

Product Structure Diagram



BN-012
End Item

1201
Head

1202
Body

1207
Arm-Hand-
Sword Assembly
(2-required)

1208
Leg-Foot Assembly
(2-required)

1209
Arm-Sword
Assembly

1203
Hand

1205
Foot

1206
Leg

1204
Arm

1210
*Sword

* 는 Special parts로 자체생산이 아닌 vendor에게 공급받는다

Defining Rules and Assumptions for production planning and environment

02. RULES AND ASSUMPTIONS

❖ **Product Family:** 모델 3가지가 Bionicle Family 를 구성한다.

❖ **Batch production** : MPS와 MRP는 Batch 생산 방식을 택한다.

❖ **Level Strategy** : SOP에서는 Level Strategy 를 택한다.

❖ **Make to Stock** : 완성품을 판매한다.

❖ **Special Parts Lead-time (2 weeks)** : 무기 등 현재 레고 공장 내에서 생산하기 까다로운 부품은 외주를 통해 공급받는다.

❖ **Ordinary Parts Lead-time (1 week)** : 나머지 Parts는 자체 생산하기에 Lead-time은 1주가 소요된다.

❖ **Backorders are not allowed** : 모든 단계에서 Backorder를 허용하지 않는다.

❖ **Safety Stock** : MPS, MRP 단계에서 Backorder 방지를 위하여 Safety Stock을 설정한다.

An Example of the SOP, MPS and MRP

03. SYSTEM WITH SIMULATIONS

SOP EXAMPLE; BIONICLE FAMILY

BN-008		**Plan**											
	On hand	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Sales													
Forecast(in units)		70	80	120	130	60	80	90	100	110	80	130	150
Operations													
Plan(in units)		100	100	100	100	100	100	100	100	100	100	100	100
Inventory													
Plan(in units)	0	30	50	30	0	40	60	70	70	60	80	50	0

An Example of the SOP, MPS and MRP

03. SYSTEM WITH SIMULATIONS

MPS EXAMPLE; JIMI STRINGER (BN-008)

		Month											
	On hand	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Forecast		100	100	100	100	100	100	100	100	100	100	100	100
Orders		75	120	95	60	95	80	100	110	90	95	110	130
Projected available balance	0	200	80	280	180	80	280	180	70	270	170	60	230
Available-to-promise		105		50			10			5			170
MPS		300		300			300			300			300
Lot size=300, Safety stock=30													

An Example of the SOP, MPS and MRP

03. SYSTEM WITH SIMULATIONS

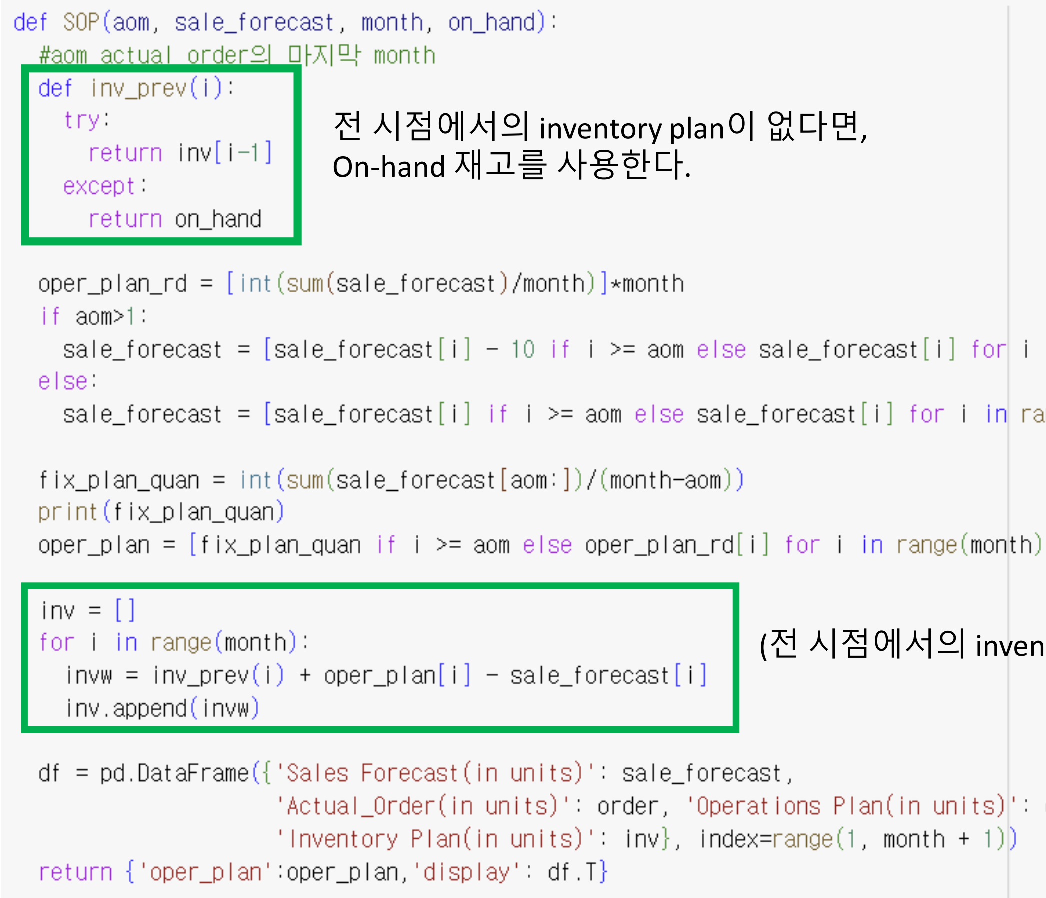
MRP EXAMPLE; ARM-HAND ASSEMBLY (0807)

		Month											
	On hand	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Gross requirements		600		600			600			600			600
Scheduled receipts		600											
Projected available balance	60	60	60	60	60	60	60	60	60	60	60	60	60
Planned receipts				600			600			600			600
Planned order releases			600			600			600			600	
Lead time=1, Lot size=600 Safety stock=60													

The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

1. SOP CODE

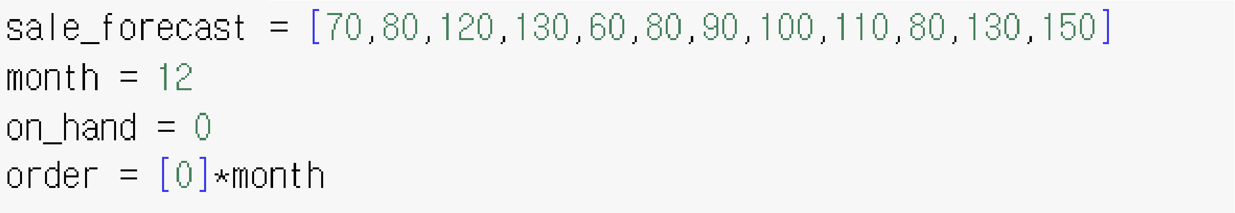


```
def SOP(aom, sale_forecast, month, on_hand):
  #aom actual order의 마지막 month
  def inv_prev(i):
    try:
      return inv[i-1]
    except:
      return on_hand

  oper_plan_rd = [int(sum(sale_forecast)/month)]*month
  if aom>1:
    sale_forecast = [sale_forecast[i] - 10 if i >= aom else sale_forecast[i] for i in range(month)]
  else:
    sale_forecast = [sale_forecast[i] if i >= aom else sale_forecast[i] for i in range(month)]

  fix_plan_quan = int(sum(sale_forecast[aom:])/(month-aom))
  print(fix_plan_quan)
  oper_plan = [fix_plan_quan if i >= aom else oper_plan_rd[i] for i in range(month)]

  inv = []
  for i in range(month):
    invw = inv_prev(i) + oper_plan[i] - sale_forecast[i]
    inv.append(invw)

  df = pd.DataFrame({'Sales Forecast(in units)': sale_forecast,
                     'Actual_Order(in units)': order, 'Operations Plan(in units)': oper_plan,
                     'Inventory Plan(in units)': inv}, index=range(1, month + 1))
  return {'oper_plan':oper_plan,'display': df.T}
```

전 시점에서의 inventory plan이 없다면,
On-hand 재고를 사용한다.

(전 시점에서의 inventory plan) + (operations plan) - (sales forecast)

INPUT VARIABLES

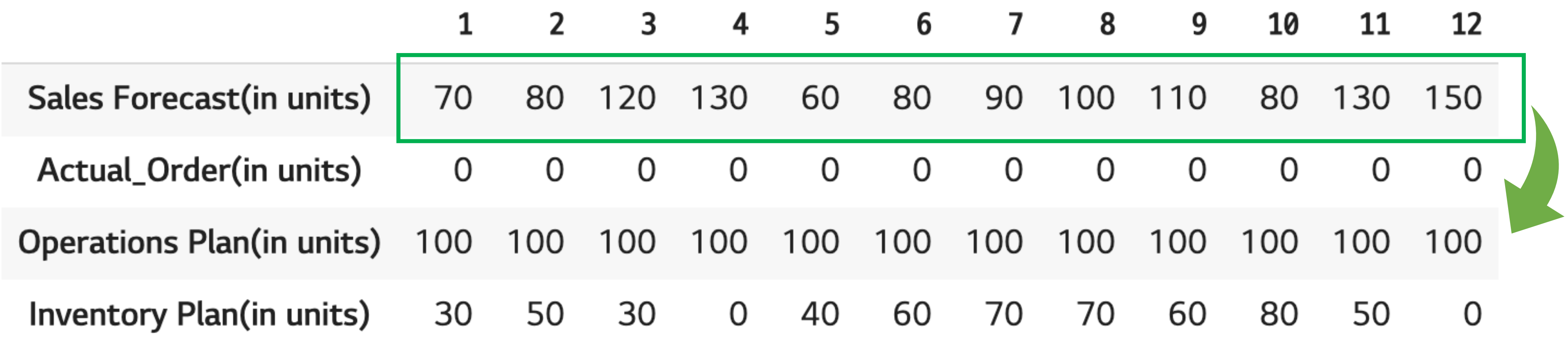


```
sale_forecast = [70,80,120,130,60,80,90,100,110,80,130,150]
month = 12
on_hand = 0
order = [0]*month
```

The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

1. SOP OUTPUT



	1	2	3	4	5	6	7	8	9	10	11	12
Sales Forecast(in units)	70	80	120	130	60	80	90	100	110	80	130	150
Actual_Order(in units)	0	0	0	0	0	0	0	0	0	0	0	0
Operations Plan(in units)	100	100	100	100	100	100	100	100	100	100	100	100
Inventory Plan(in units)	30	50	30	0	40	60	70	70	60	80	50	0

Sum(forecast)/12
= 100

Level production strategy 를 채택했기 때문에
모든 period에서 생산계획(Operation Plan)이 일정하다.

The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

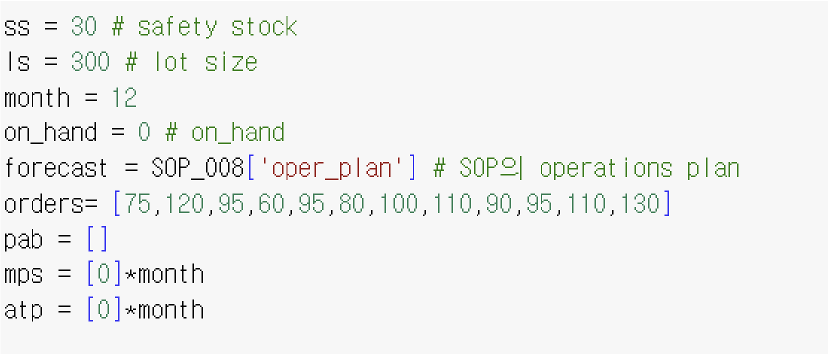
2. MPS CODE

Safety Stock(SS)보다 재고량이 낮아지면,
Lot Size(LS)만큼 생산해준다.

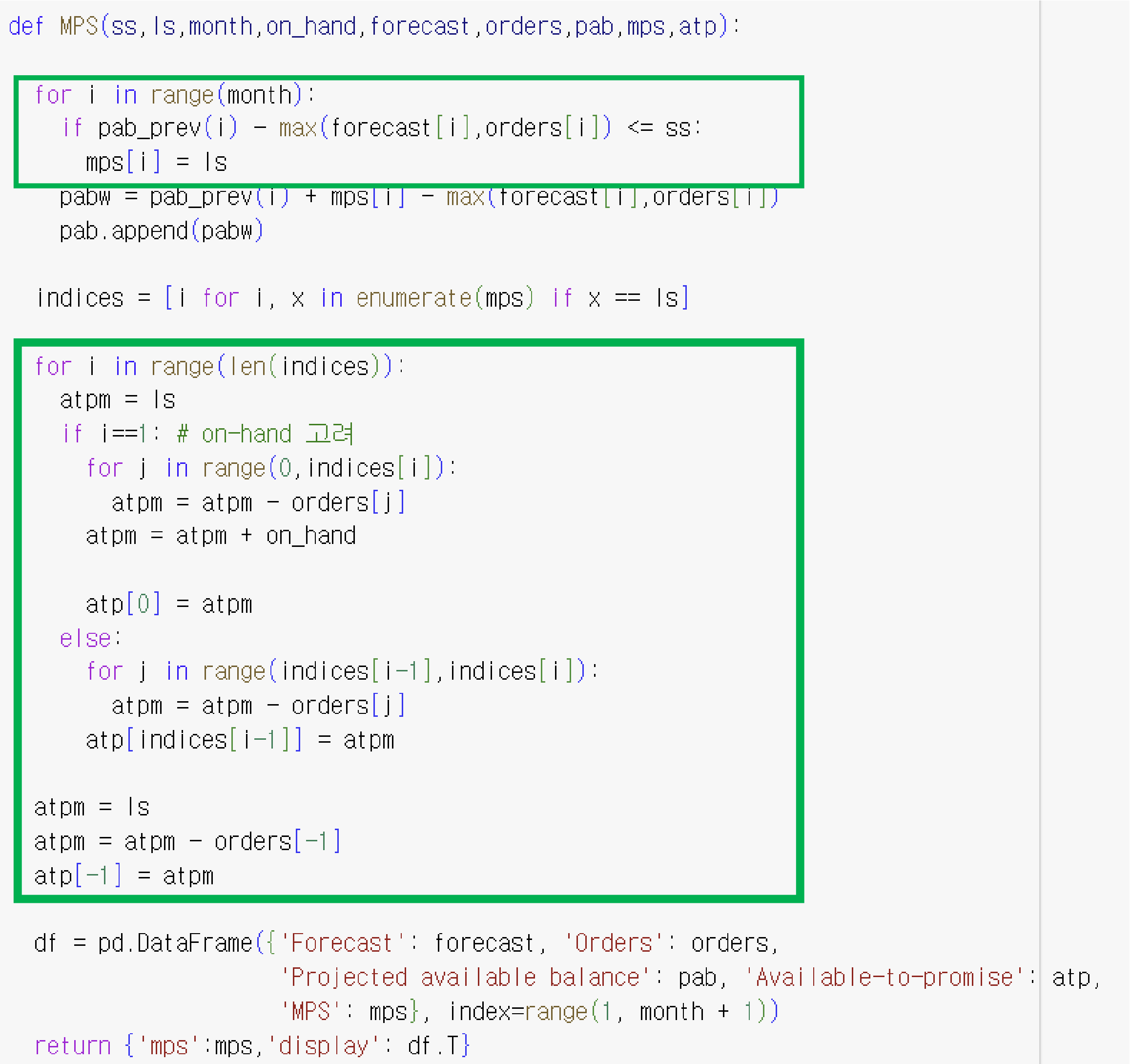
현재 MPS가 공급된 시점부터 다음 MPS 공급시점
까지 orders 뺀다.(ATP)

(만약 첫번째 시점이라면 on hand)만큼 더해준다.

INPUT VARIABLES



```
ss = 30 # safety stock
ls = 300 # lot size
month = 12
on_hand = 0 # on_hand
forecast = SOP_008['oper_plan'] # SOP의 operations plan
orders= [75,120,95,60,95,80,100,110,90,95,110,130]
pab = []
mps = [0]*month
atp = [0]*month
```

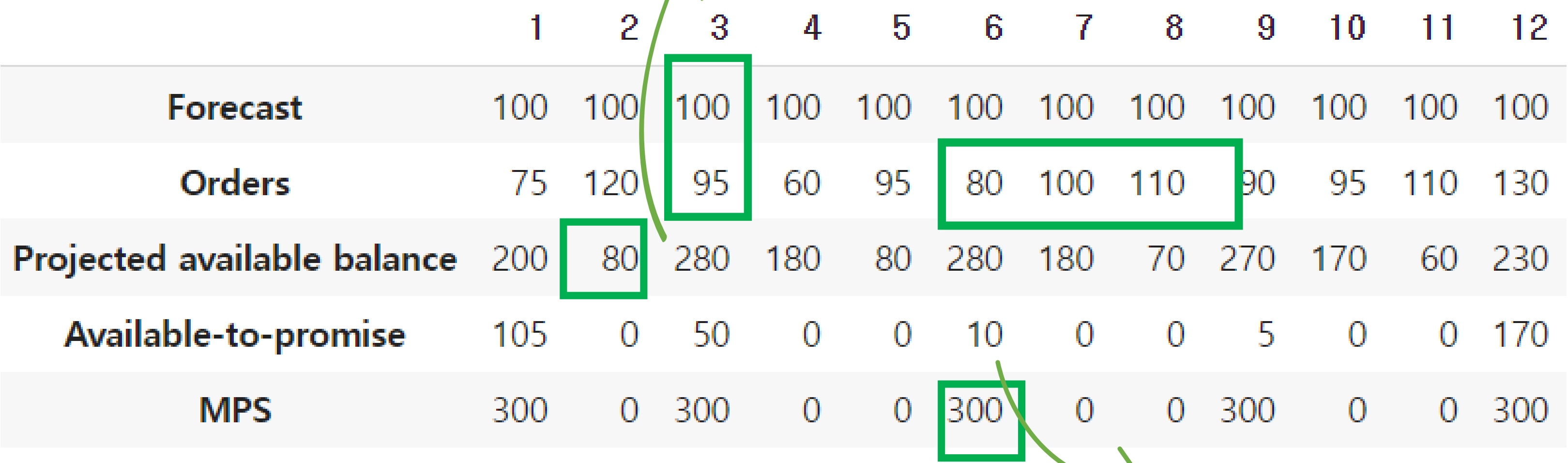


```
def MPS(ss,ls,month,on_hand,forecast,orders,pab,mps,atp):

  for i in range(month):
    if pab_prev(i) - max(forecast[i],orders[i]) <= ss:
      mps[i] = ls
    pabw = pab_prev(i) + mps[i] - max(forecast[i],orders[i])
    pab.append(pabw)

  indices = [i for i, x in enumerate(mps) if x == ls]

  for i in range(len(indices)):
    atpm = ls
    if i==1: # on-hand 고려
      for j in range(0,indices[i]):
        atpm = atpm - orders[j]
      atpm = atpm + on_hand

      atp[0] = atpm
    else:
      for j in range(indices[i-1],indices[i]):
        atpm = atpm - orders[j]
      atp[indices[i-1]] = atpm

  atpm = ls
  atpm = atpm - orders[-1]
  atp[-1] = atpm

  df = pd.DataFrame({'Forecast': forecast, 'Orders': orders,
                     'Projected available balance': pab, 'Available-to-promise': atp,
                     'MPS': mps}, index=range(1, month + 1))
  return {'mps':mps,'display': df.T}
```

04. CODE AND EXECUTION

2. MPS OUTPUT

300 + 80 – max(100,95)



	1	2	3	4	5	6	7	8	9	10	11	12
Forecast	100	100	100	100	100	100	100	100	100	100	100	100
Orders	75	120	95	60	95	80	100	110	90	95	110	130
Projected available balance	200	80	280	180	80	280	180	70	270	170	60	230
Available-to-promise	105	0	50	0	0	10	0	0	5	0	0	170
MPS	300	0	300	0	0	300	0	0	300	0	0	300

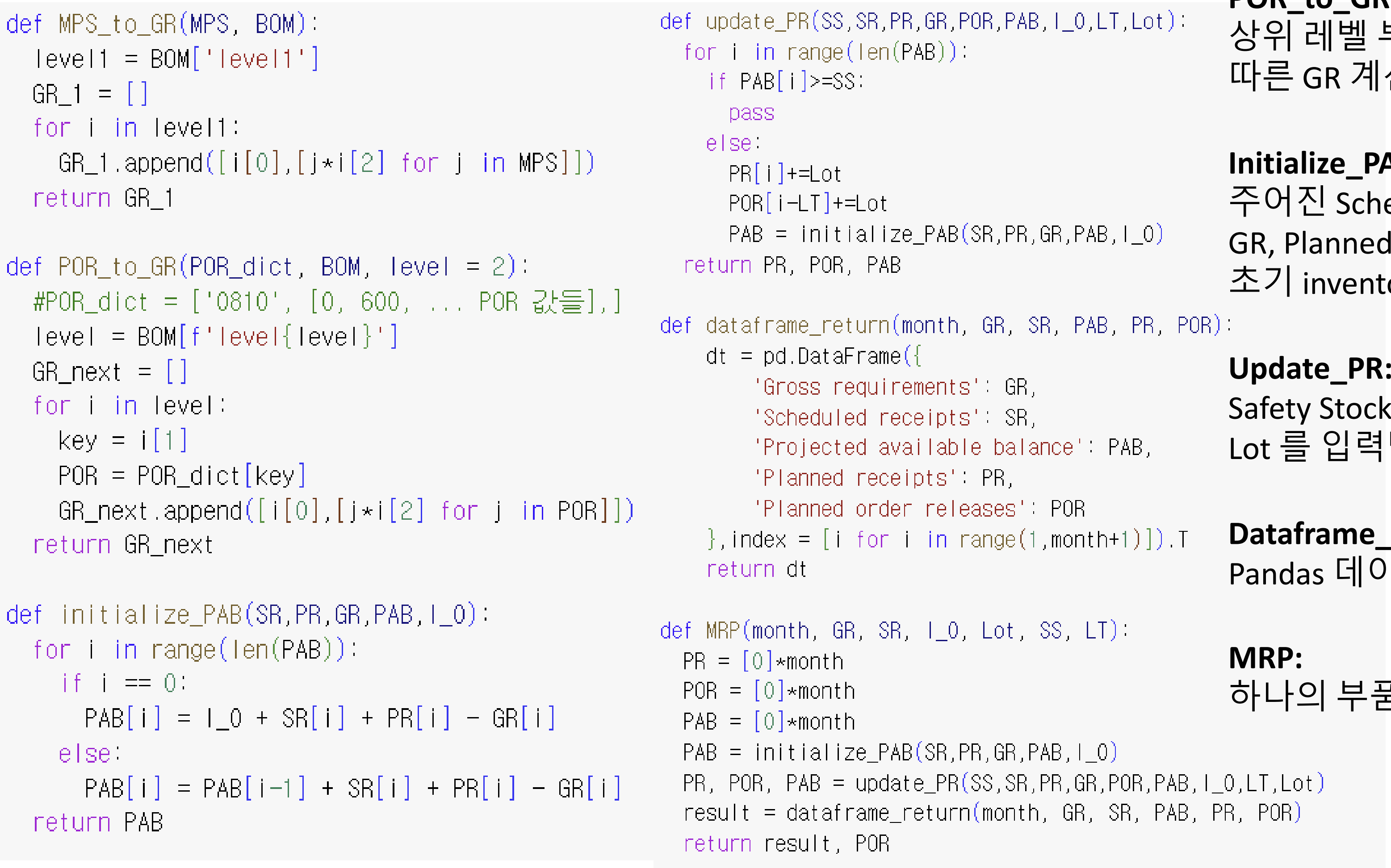
300 – (80+100+110)

Batch production strategy 를 채택했기 때문에
재고가 Safety Stock(30) 로 떨어지게 되면, Lot Size(300)만큼 생산해준다.

The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

3. MRP CODE



```
def MPS_to_GR(MPS, BOM):
  level1 = BOM['level1']
  GR_1 = []
  for i in level1:
    GR_1.append([i[0],[j*i[2] for j in MPS]])
  return GR_1

def POR_to_GR(POR_dict, BOM, level = 2):
  #POR_dict = ['0810', [0, 600, ... POR 값들],]
  level = BOM[f'level{level}']
  GR_next = []
  for i in level:
    key = i[1]
    POR = POR_dict[key]
    GR_next.append([i[0],[j*i[2] for j in POR]])
  return GR_next

def initialize_PAB(SR,PR,GR,PAB,I_0):
  for i in range(len(PAB)):
    if i == 0:
      PAB[i] = I_0 + SR[i] + PR[i] - GR[i]
    else:
      PAB[i] = PAB[i-1] + SR[i] + PR[i] - GR[i]
  return PAB
```

```
def update_PR(SS,SR,PR,GR,POR,PAB,I_0,LT,Lot):
  for i in range(len(PAB)):
    if PAB[i]>=SS:
      pass
    else:
      PR[i]+=Lot
      POR[i-LT]+=Lot
      PAB = initialize_PAB(SR,PR,GR,PAB,I_0)
  return PR, POR, PAB

def dataframe_return(month, GR, SR, PAB, PR, POR):
    dt = pd.DataFrame({
        'Gross requirements': GR,
        'Scheduled receipts': SR,
        'Projected available balance': PAB,
        'Planned receipts': PR,
        'Planned order releases': POR
    },index = [i for i in range(1,month+1)]).T
    return dt

def MRP(month, GR, SR, I_0, Lot, SS, LT):
  PR = [0]*month
  POR = [0]*month
  PAB = [0]*month
  PAB = initialize_PAB(SR,PR,GR,PAB,I_0)
  PR, POR, PAB = update_PR(SS,SR,PR,GR,POR,PAB,I_0,LT,Lot)
  result = dataframe_return(month, GR, SR, PAB, PR, POR)
  return result, POR
```

Code Explanation

MPS_to_GR:
주어진 MPS에 맞춰, BOM 개수에 따른 Gross Requirements(GR) 계산

POR_to_GR:
상위 레벨 부품의 Planned Order Release(POR)에 따른 GR 계산

Initialize_PAB:
주어진 Scheduled Receipts(SR), Planned Receipts(PR), GR, Planned Available Balance(PAB), 초기 inventory (I_0) 에 따라 PAB 초기화

Update_PR:
Safety Stock(SS), SR, PR, POR, PAB, I_0, Lead Time(LT), Lot 를 입력받으면 주어진 규칙에 따라 PR 계산

Dataframe_return:
Pandas 데이터 프레임 형식으로 변환

MRP:
하나의 부품에 대한 MRP 전개

The Code of the SOP, MPS and MRP

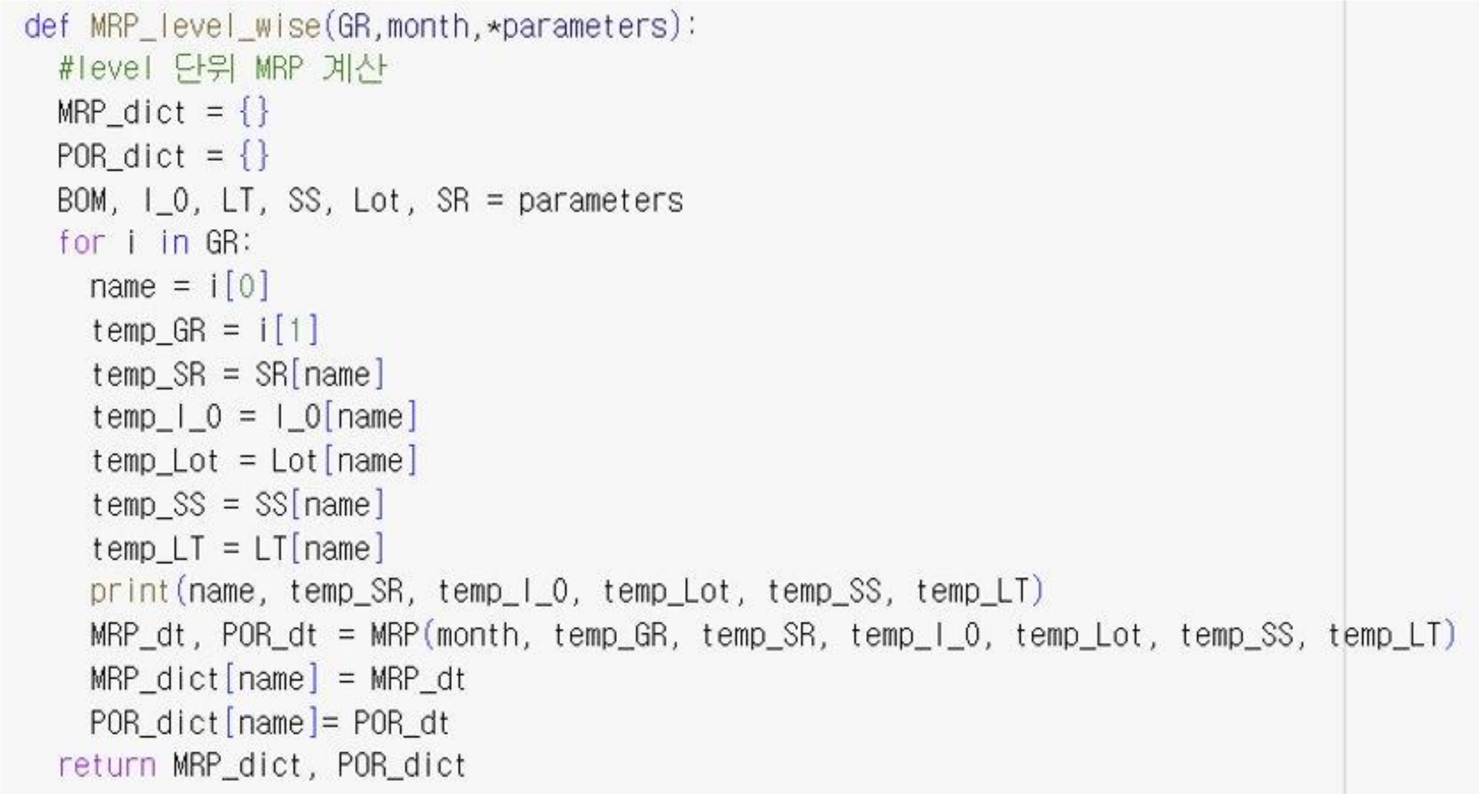
04. CODE AND EXECUTION

3. MRP CODE

Code Explanation

MRP_level_wise:

MRP 레벨 단위로 전개

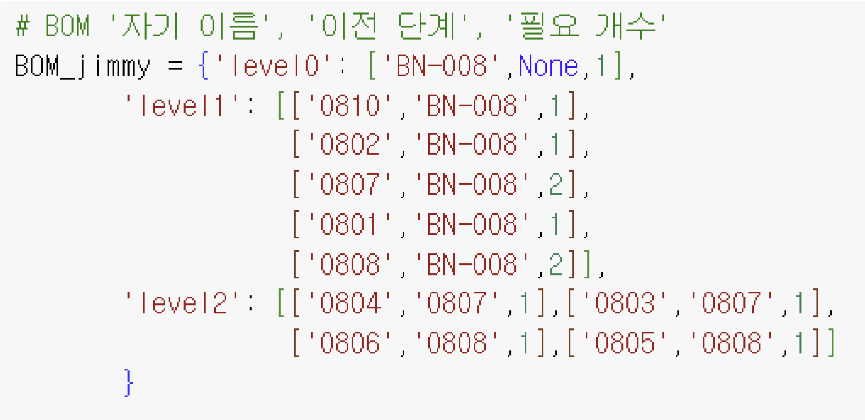


```
def MRP_level_wise(GR,month,*parameters):
  #level 단위 MRP 계산
  MRP_dict = {}
  POR_dict = {}
  BOM, I_0, LT, SS, Lot, SR = parameters
  for i in GR:
    name = i[0]
    temp_GR = i[1]
    temp_SR = SR[name]
    temp_I_0 = I_0[name]
    temp_Lot = Lot[name]
    temp_SS = SS[name]
    temp_LT = LT[name]
    print(name, temp_SR, temp_I_0, temp_Lot, temp_SS, temp_LT)
    MRP_dt, POR_dt = MRP(month, temp_GR, temp_SR, temp_I_0, temp_Lot, temp_SS, temp_LT)
    MRP_dict[name] = MRP_dt
    POR_dict[name]= POR_dt
  return MRP_dict, POR_dict
```

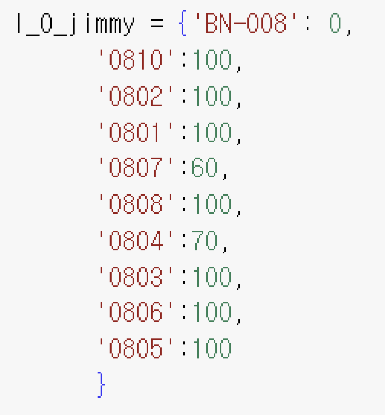
The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

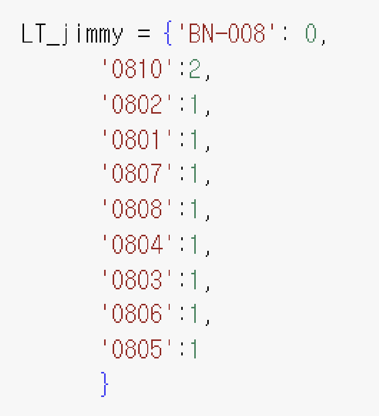
3. MRP CODE   INPUT VARIABLES



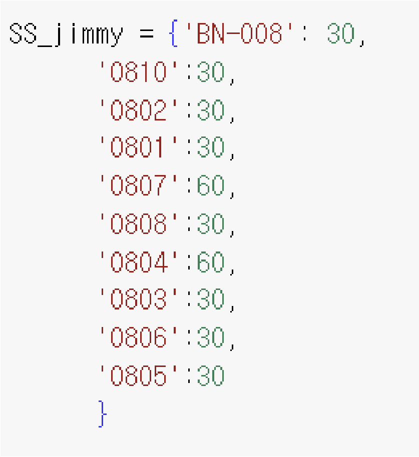
```
# BOM '자기 이름', '이전 단계', '필요 개수'
BOM_jimmy = {'level0': ['BN-008',None,1],
        'level1': [['0810','BN-008',1],
                   ['0802','BN-008',1],
                   ['0807','BN-008',2],
                   ['0801','BN-008',1],
                   ['0808','BN-008',2]],
        'level2': [['0804','0807',1],['0803','0807',1],
                   ['0806','0808',1],['0805','0808',1]]
        }
```



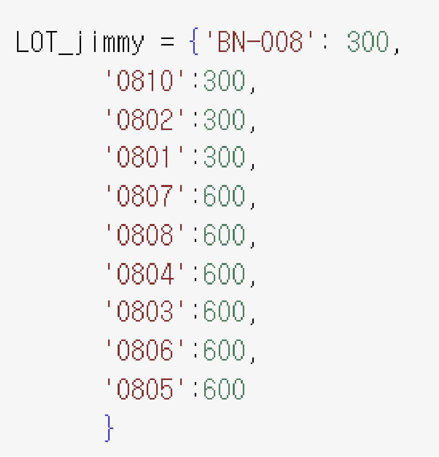
```
I_O_jimmy = {'BN-008': 0,
       '0810':100,
       '0802':100,
       '0801':100,
       '0807':60,
       '0808':100,
       '0804':70,
       '0803':100,
       '0806':100,
       '0805':100
       }
```



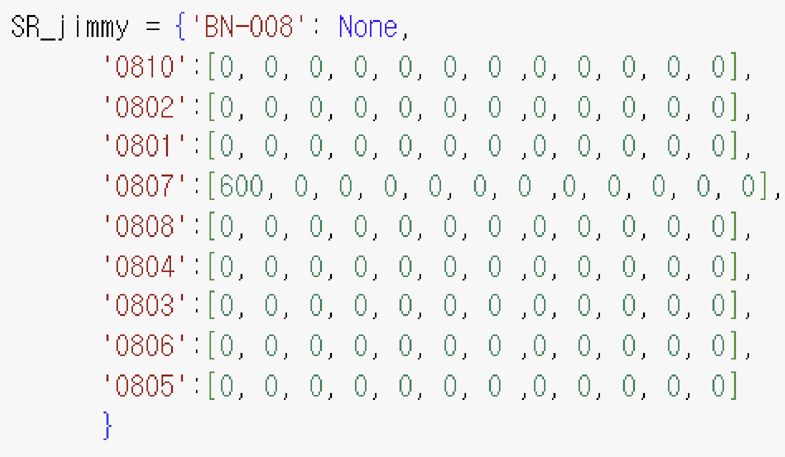
```
LT_jimmy = {'BN-008': 0,
       '0810':2,
       '0802':1,
       '0801':1,
       '0807':1,
       '0808':1,
       '0804':1,
       '0803':1,
       '0806':1,
       '0805':1
       }
```



```
SS_jimmy = {'BN-008': 30,
       '0810':30,
       '0802':30,
       '0801':30,
       '0807':60,
       '0808':30,
       '0804':60,
       '0803':30,
       '0806':30,
       '0805':30
       }
```



```
LOT_jimmy = {'BN-008': 300,
       '0810':300,
       '0802':300,
       '0801':300,
       '0807':600,
       '0808':600,
       '0804':600,
       '0803':600,
       '0806':600,
       '0805':600
       }
```

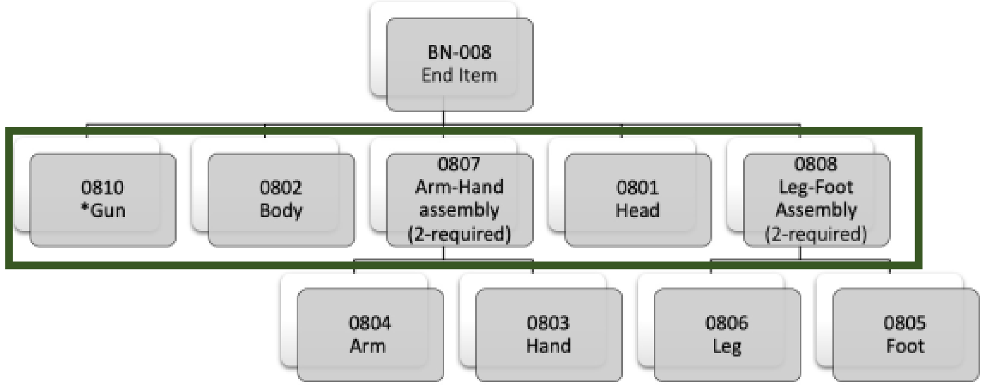


```
SR_jimmy = {'BN-008': None,
       '0810':[0, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0],
       '0802':[0, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0],
       '0801':[0, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0],
       '0807':[600, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0],
       '0808':[0, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0],
       '0804':[0, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0],
       '0803':[0, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0],
       '0806':[0, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0],
       '0805':[0, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0]
       }
```

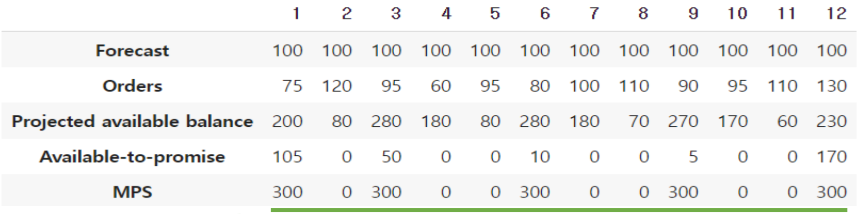
The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

3. MRP_LEVEL1 (OUTPUT 일부)

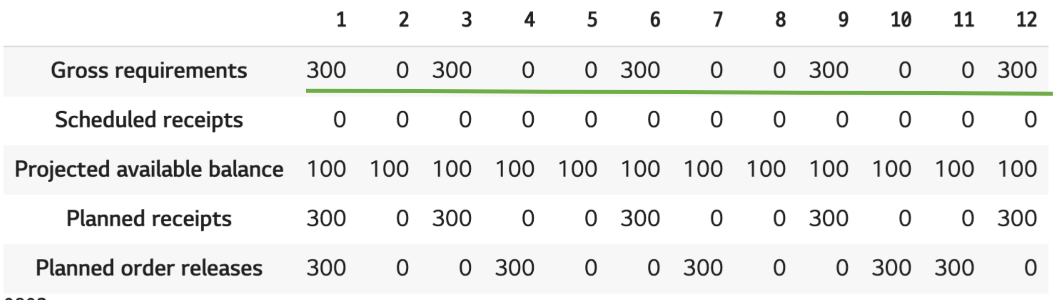


MPS



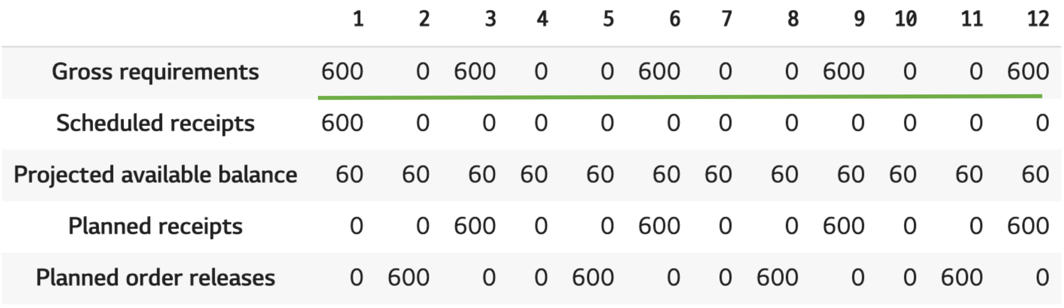
	1	2	3	4	5	6	7	8	9	10	11	12
Forecast	100	100	100	100	100	100	100	100	100	100	100	100
Orders	75	120	95	60	95	80	100	110	90	95	110	130
Projected available balance	200	80	280	180	80	280	180	70	270	170	60	230
Available-to-promise	105	0	50	0	0	10	0	0	5	0	0	170
MPS	300	0	300	0	0	300	0	0	300	0	0	300

Gun (0810)



	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	300	0	300	0	0	300	0	0	300	0	0	300
Scheduled receipts	0	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	100	100	100	100	100	100	100	100	100	100	100	100
Planned receipts	300	0	300	0	0	300	0	0	300	0	0	300
Planned order releases	300	0	0	300	0	0	300	0	0	300	300	0

Arm-Hand assembly (0807)



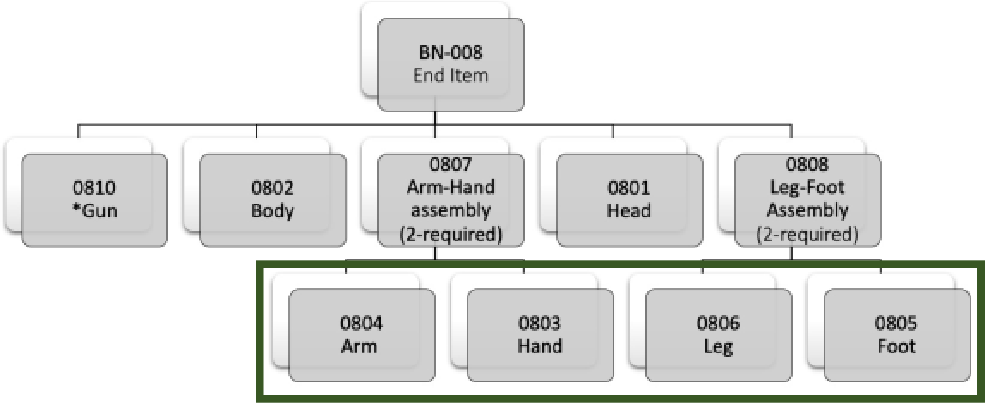
	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	600	0	600	0	0	600	0	0	600	0	0	600
Scheduled receipts	600	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	60	60	60	60	60	60	60	60	60	60	60	60
Planned receipts	0	0	600	0	0	600	0	0	600	0	0	600
Planned order releases	0	600	0	0	600	0	0	600	0	0	600	0

Batch production strategy 를 채택했기 때문에 재고가 Safety Stock 아래로 떨어지게 되면,
Lead Time을 고려하여 Lot Size만큼 생산해준다.

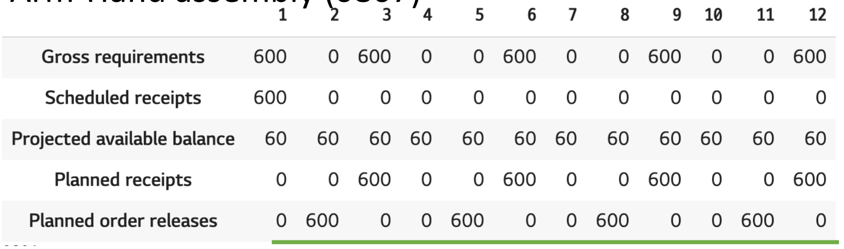
The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

3. MRP_LEVEL2 (OUTPUT 일부)

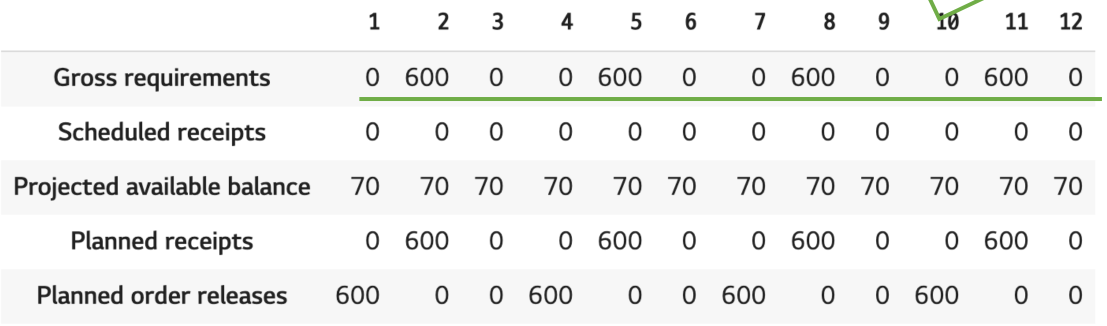


Arm-Hand assembly (0807)



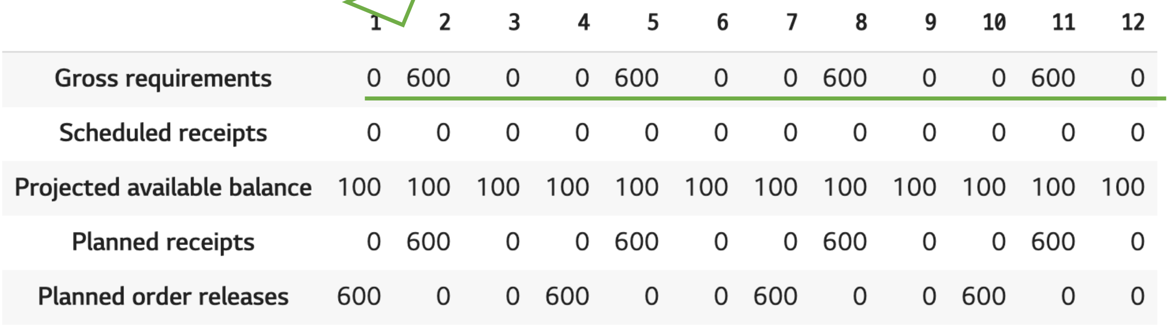
	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	600	0	600	0	0	600	0	0	600	0	0	600
Scheduled receipts	600	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	60	60	60	60	60	60	60	60	60	60	60	60
Planned receipts	0	0	600	0	0	600	0	0	600	0	0	600
Planned order releases	0	600	0	0	600	0	0	600	0	0	600	0

Hand (0804)



	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	0	600	0	0	600	0	0	600	0	0	600	0
Scheduled receipts	0	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	70	70	70	70	70	70	70	70	70	70	70	70
Planned receipts	0	600	0	0	600	0	0	600	0	0	600	0
Planned order releases	600	0	0	600	0	0	600	0	0	600	0	0

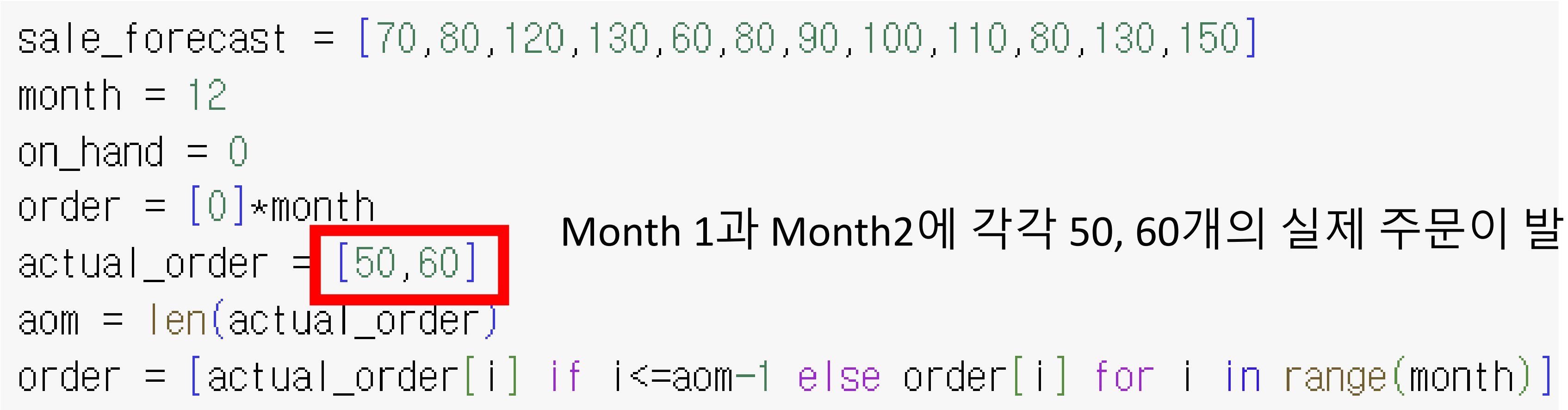
Arm (0803)



	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	0	600	0	0	600	0	0	600	0	0	600	0
Scheduled receipts	0	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	100	100	100	100	100	100	100	100	100	100	100	100
Planned receipts	0	600	0	0	600	0	0	600	0	0	600	0
Planned order releases	600	0	0	600	0	0	600	0	0	600	0	0

04. CODE AND EXECUTION

4. ACTUAL ORDERS ADDED



```
sale_forecast = [70,80,120,130,60,80,90,100,110,80,130,150]
month = 12
on_hand = 0
order = [0]*month
actual_order = [50,60]
aom = len(actual_order)
order = [actual_order[i] if i<=aom-1 else order[i] for i in range(month)]
```

Month 1과 Month2에 각각 50, 60개의 실제 주문이 발생하였다 가정

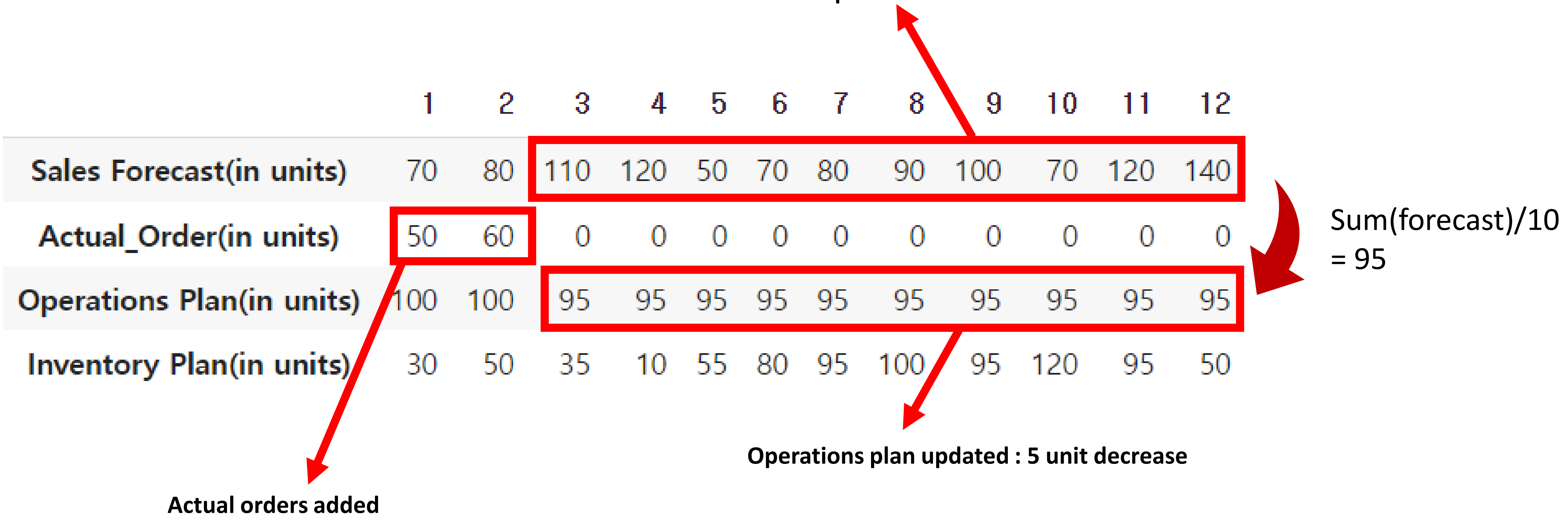
실제 주문이 sale_forcast보다 적게 발생했기 때문에 Month3~12까지의 sale forecast을 모두 10 낮춘다.

SOP는 Level production을 따르기 때문에 각 period에서의 **operation plan UPDATE 100 >> 95**

The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

4. ACTUAL ORDERS ADDED



	1	2	3	4	5	6	7	8	9	10	11	12
Sales Forecast(in units)	70	80	110	120	50	70	80	90	100	70	120	140
Actual_Order(in units)	50	60	0	0	0	0	0	0	0	0	0	0
Operations Plan(in units)	100	100	95	95	95	95	95	95	95	95	95	95
Inventory Plan(in units)	30	50	35	10	55	80	95	100	95	120	95	50

Forecast updated : 10 unit decrease

Sum(forecast)/10
= 95

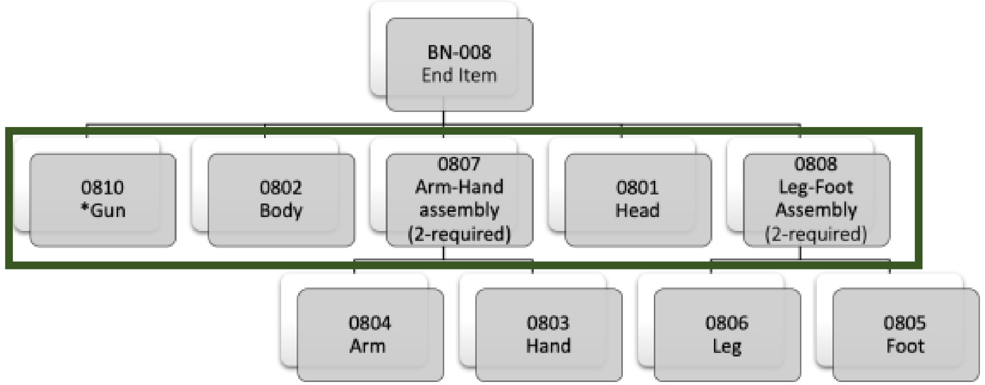
Operations plan updated : 5 unit decrease

Actual orders added

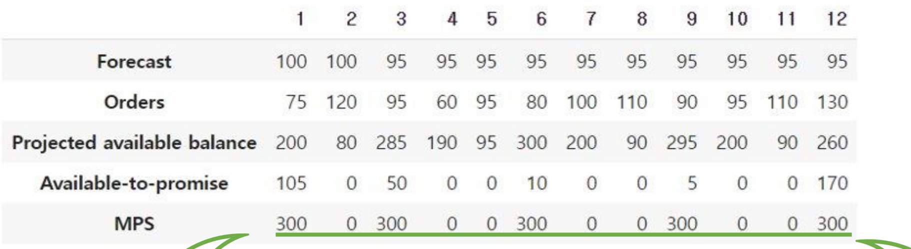
The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

4. ACTUAL ORDERS ADDED; MRP_LEVEL1 (OUTPUT 일부)

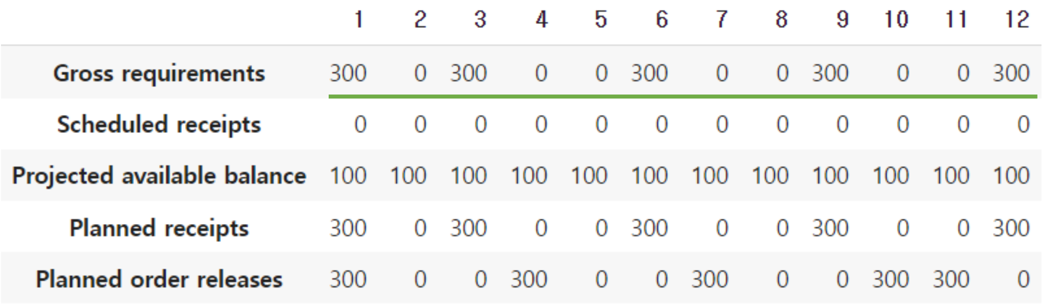


MPS



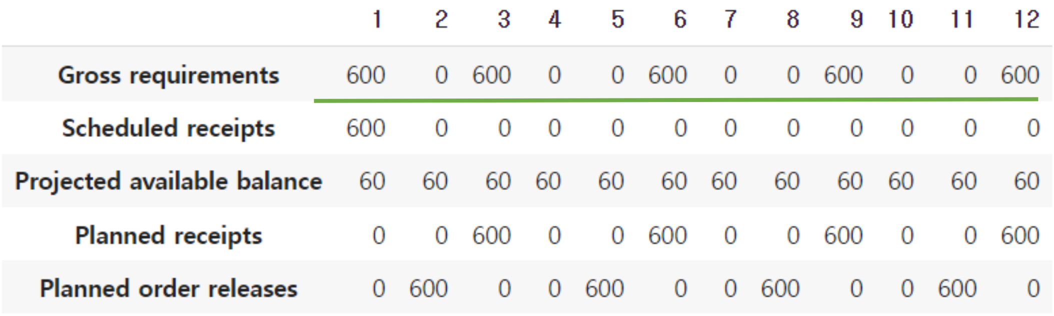
	1	2	3	4	5	6	7	8	9	10	11	12
Forecast	100	100	95	95	95	95	95	95	95	95	95	95
Orders	75	120	95	60	95	80	100	110	90	95	110	130
Projected available balance	200	80	285	190	95	300	200	90	295	200	90	260
Available-to-promise	105	0	50	0	0	10	0	0	5	0	0	170
MPS	300	0	300	0	0	300	0	0	300	0	0	300

Gun (0810)



	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	300	0	300	0	0	300	0	0	300	0	0	300
Scheduled receipts	0	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	100	100	100	100	100	100	100	100	100	100	100	100
Planned receipts	300	0	300	0	0	300	0	0	300	0	0	300
Planned order releases	300	0	0	300	0	0	300	0	0	300	300	0

Arm-Hand assembly (0807)

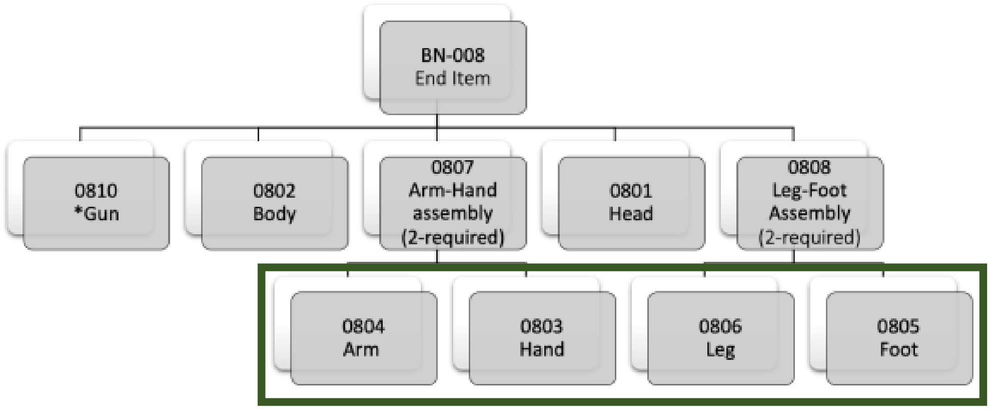


	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	600	0	600	0	0	600	0	0	600	0	0	600
Scheduled receipts	600	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	60	60	60	60	60	60	60	60	60	60	60	60
Planned receipts	0	0	600	0	0	600	0	0	600	0	0	600
Planned order releases	0	600	0	0	600	0	0	600	0	0	600	0

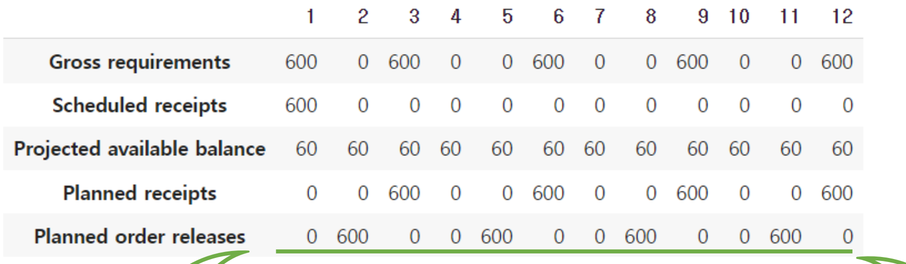
The Code of the SOP, MPS and MRP

04. CODE AND EXECUTION

4. ACTUAL ORDERS ADDED; MRP_LEVEL2 (OUTPUT 일부)

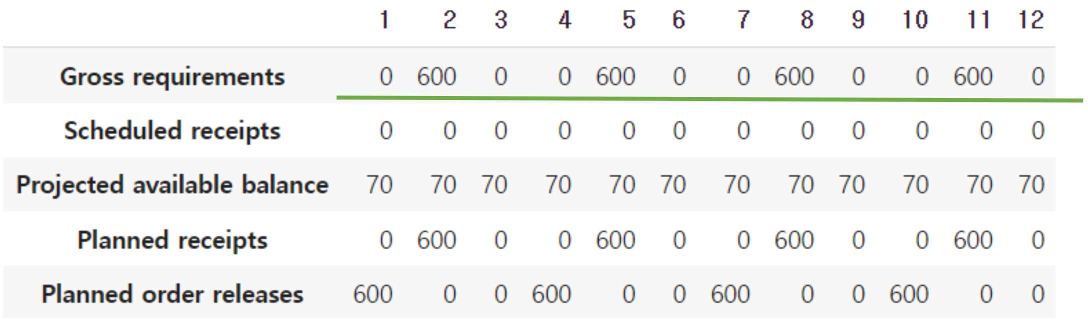


Arm-Hand assembly (0807)



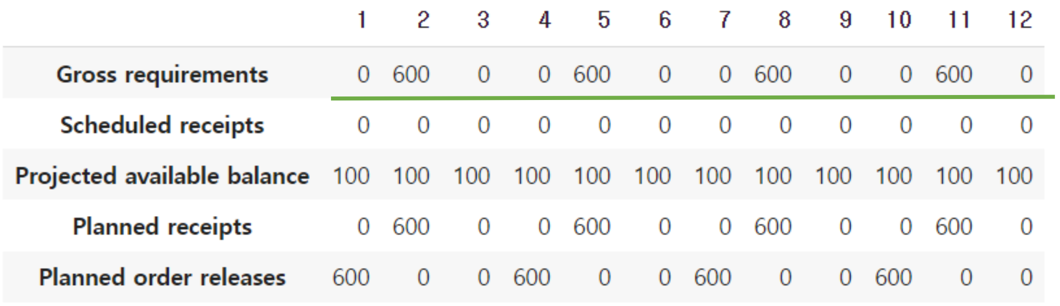
	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	600	0	600	0	0	600	0	0	600	0	0	600
Scheduled receipts	600	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	60	60	60	60	60	60	60	60	60	60	60	60
Planned receipts	0	0	600	0	0	600	0	0	600	0	0	600
Planned order releases	0	600	0	0	600	0	0	600	0	0	600	0

Hand (0804)



	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	0	600	0	0	600	0	0	600	0	0	600	0
Scheduled receipts	0	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	70	70	70	70	70	70	70	70	70	70	70	70
Planned receipts	0	600	0	0	600	0	0	600	0	0	600	0
Planned order releases	600	0	0	600	0	0	600	0	0	600	0	0

Arm (0803)



	1	2	3	4	5	6	7	8	9	10	11	12
Gross requirements	0	600	0	0	600	0	0	600	0	0	600	0
Scheduled receipts	0	0	0	0	0	0	0	0	0	0	0	0
Projected available balance	100	100	100	100	100	100	100	100	100	100	100	100
Planned receipts	0	600	0	0	600	0	0	600	0	0	600	0
Planned order releases	600	0	0	600	0	0	600	0	0	600	0	0

EOD