

Chapter 9

청산 당하면 어떻게? 양방향 그리드 매매 개선하기!

봇 제작을 도와 드려요 : <https://blog.naver.com/zacra/222930668193>

영상에서 구현된 내용과 다르게 **추가적으로 수정된 부분이 있으니 첨부 파일의 설명(주석)을 꼭 정독하** 세요 ^^ 다음 강의에서 또 추가 설명을 해드리니 이어서 수강 해보세요!

만약 더 복잡해져서 이해가 안가신다면 **간략한 이전 강의 9-3의 코드를 기반으로 수정** 해 보세요. 이번 챗터와 제가 추가적으로 수정한 이 봇은 어디까지나 양방향 매매의 예시지 정답이 아니고 **수익을 보장하지 않는**다는 점 기억하세요!

#영상에서 다루진 않지만 선물 봇은 바이비트(Bybit) 거래소 코드도 함께 제공 합니다!

바이낸스 양방향 헷지 모드 정리 :

<https://blog.naver.com/zacra/222662884649>

내 PC에서 AWS서버 직접 접속하는 방법 :

<https://blog.naver.com/zacra/222622754580>

#업비트 거미줄 매매는 여기서 :

<https://blog.naver.com/zacra/222659507362>

[수업목표]

양방향 그리드 거미줄 매매 전략을 보완해봅니다!

[수업개요]

```

binance_auto_Grid.py 1 X
Users > zabob > Desktop > autobot > binance_auto_Grid.py > ...
186 print("Remain Money:", float(balance["USD"]['free']))
187
188
189 leverage = set_leverage #레이버리지
190
191 #매달 코인 가격을 가져온다.
192 coin_price = mybinance.GetCoinPrice(binanceX, Target_Coin_Ticker)
193
194
195 #매달 코인에 할당된 금액에 따른 최대 매수수량을 구해준다!
196 Max_Ant = float(binanceX.amount_to_precision(Target_Coin_Ticker, mybinance.GetAmount(float(balance["USD"]['total']), coin_price, Invest_Rate / CoinCnt))) * leverage
197
198 print("Max_Ant:", Max_Ant)
199
200
201 #100분마다 매서 1회 매수 코인 수량으로 정한다!
202 Buy_Ant = Max_Ant / 100.0
203 Buy_Ant = float(binanceX.amount_to_precision(Target_Coin_Ticker, Buy_Ant))
204
205 #####
206 #동 수 각각 거미줄들에 할당할 액스 수량!
207 Max_Water_Ant = Buy_Ant * 40.0
208 #####
209
210 print("Buy_Ant:", Buy_Ant)
211
212
213 #최소 주문 수량보다 작으면 이렇게 설정!
214 if Buy_Ant < minimum_amount:
215     Buy_Ant = minimum_amount
216
217
218 print("Final Buy_Ant:", Buy_Ant)
219
220
221 ant_s = 0
222
223

```



01:05 거미줄 까는 로직 개선하기

옵션1 **익절해서** 포지션을 새로 잡을때.. 물량을 좀 늘려주는거예요.

-> 그러면 롱은 계속 익절! 숏은 계속 물린다.

-> 숏 물량이 50인데 롱은 계속 1씩 익절한다.

-> 간의 기별도 안가겠조?

-> 그래서 1/4 수준으로 물량을 맞추주는거예요.

-> 왜 똑같이 물량을 맞춰주지 않나요? (반대로 움직이면 이쪽이 물리잖아요)

옵션2 물렸다. 거미줄을 다 소진했다. 그러면 **손절**

-> 절반을 손절합니다.

그리고 그 절반을 절반에 해당하는 물량만큼 거미줄을 또 끈다.

-> 그래서 이를 반복한다...

-> 언젠간 탈출 하겠지..




11:57 거미줄이 뚫렸을 때 대응 옵션 2개 소개

```

227 data = binanceX.create_market_sell_order(Target_Coin_Ticker, Buy_Amt, params)
228
229
230 #익절할 가격을 구합니다.
231 target_price = data['price'] * (1.0 - target_rate)
232
233
234 #그리고 지정가로 익절 주문을 걸어놓는다!
235 params = {
236     'positionSide': 'SHORT'
237 }
238 print(binanceX.create_limit_buy_order(Target_Coin_Ticker, data['amount'], target_price, params))
239
240
241 #아래는 물타기 라인을 긋는 로직입니다.
242
243
244
245 TotalWater_Amt = 0 #누적 거미줄에 걸린 수량
246
247 Water_amt = Buy_Amt
248 print("Water_amt", Water_amt)
249
250 i = 1
251
252 while TotalWater_Amt + Water_amt < Max_Water_Amt:
253     print("----->", i, ", Grid!!!")
254
255     water_price = data['price'] * (1.0 + (st_water_gap_rate * i)) # 0.25%씩 가격이 상승합니다.
256
257     #실제 물타는 매수 라인 주문을 넣는다.
258     params = {
259         'positionSide': 'SHORT'

```



14:02 옵션1 구현하기


```

#할만 손절!! 시장거로!
params = {
    'positionSide': 'SHORT'
}
print(binanceX.create_market_buy_order(Target_Coin_Ticker, abs(amt_s) * 0.5, params))

#라고 데이터 가져오기
balance = binanceX.fetch_balance(params={"type": "future"})
time.sleep(0.1)

print("-----")
#숫자고
for posi in balance['info']['positions']:
    if posi['symbol'] == Target_Coin_Symbol and posi['positionSide'] == 'SHORT':
        print(posi)
        amt_s = float(posi['positionAmt'])
        entryPrice_s = float(posi['entryPrice'])
        leverage = float(posi['leverage'])
        isolated = posi['isolated']
        break

```



15:09 옵션2 구현하기