

## تابع open\_position:

تابعی برای ارسال درخواست باز کردن موقعیت تجاری.

param request: درخواست باز کردن موقعیت تجاری.  
type request: Dict  
return: موفقیت ارسال درخواست و اطلاعات معامله.  
rtype: Tuple[bool, Trade]

## تابع modify\_position:

تابعی برای ارسال درخواست تغییر موقعیت تجاری.

param request: درخواست تغییر موقعیت تجاری.  
type request: Dict  
return: موفقیت ارسال درخواست و اطلاعات معامله.  
rtype: Tuple[bool, List[Dict]]

## تابع set\_action:

تابعی برای تعیین عملیات (خرید، فروش یا لغو) بر اساس وضعیت کنونی کندل و موقعیت‌های باز.

param initialize: لیست اطلاعات مربوط به متاپیر.  
type initialize: List  
param positions: لیست موقعیت‌های باز.  
type positions: List  
param position\_index: اندیس موقعیت مورد نظر.  
type position\_index: int  
param symbol: نماد معاملاتی.  
type symbol: str  
param price\_news\_time: قیمت موقعیت تجاری در زمان اطلاعیه خبری.  
type price\_news\_time: float  
return: عملیات مورد نظر (خرید، فروش یا لغو).  
rtype: str

## تابع count\_num\_hits:

تابعی برای شمارش تعداد دفعاتی که قیمت در یک داده‌فریم مشخص از یک مقدار دیگر بزرگتر یا کوچکتر است.

param price: قیمت مقایسه.  
type price: float  
param df: داده‌فریم مورد بررسی.  
type df: pd.DataFrame

**return:** تعداد دفعات قیمت بزرگتر یا کوچکتر از مقدار دیگر در داده فریم.  
**rtype:** int:

## تابع PositionSize:

محاسبه اندازه موقعیت بر اساس پارامترهای معامله داده شده.

**param symbol:** یک رشته که نماد ارز را نشان می دهد که می خواهیم اندازه موقعیت را محاسبه کنیم.  
**type symbol:** str  
**param entry:** یک عدد اعشاری که نشان دهنده قیمت ورود معامله است.  
**type entry:** float  
**param sl:** یک عدد اعشاری که نشان دهنده قیمت توقف ضرر معامله است.  
**type sl:** float  
**param risk:** یک عدد اعشاری که نشان دهنده حداکثر میزان سرمایه گذاری در ارزش پایه حساب است.  
**type risk:** float  
**return:** یک عدد اعشاری که نشان دهنده اندازه موقعیت به واحد لات است.  
**rtype:** float:

این تابع ابتدا نماد معاملاتی را با استفاده از `mt5.symbol_select()` انتخاب می کند، اطلاعات مربوط به اندازه و ارزش تیک معامله را با استفاده از `mt5.symbol_info()` دریافت می کند. سپس اطلاعات تیک کنونی برای نماد را با استفاده از `mt5.symbol_info_tick()` دریافت کرده، حداقل فاصله مجاز را به عنوان 5 برابر اندازه تیک محاسبه می کند. سپس تعداد تیک های خطرناک بر اساس قیمت ورودی و قیمت توقف ضرر محاسبه شده و اندازه موقعیت به واحد لات بر اساس حداکثر میزان سرمایه گذاری در ارزش پایه حساب محاسبه می شود. حاصل اندازه موقعیت به صورت یک عدد اعشاری و به دو رقم اعشار گرد می شود.

مثال:

```
# محاسبه اندازه موقعیت برای یک معامله با نماد EURUSD
'symbol' = 'EURUSD'
entry = 1.2345
sl = 1.2300
risk = 1000 # حداکثر میزان سرمایه گذاری 1000$
(position_size = PositionSize(symbol, entry, sl, risk
```

## تابع Open\_Position:

این تابع یک دیکشنری حاوی اطلاعات معامله را دریافت می کند، از جمله نقطه ورود، تیک پروفیت، استاپ لاس، اندازه موقعیت، نام ارز و عملیات (خرید یا فروش) و یک معامله باز یا یک معامله در حال انتظار را ایجاد می کند.

ورودی ها:

**trade\_info (dict):** یک دیکشنری که شامل اطلاعات معامله است، از جمله نقطه ورود، تیک پروفیت، استاپ لاس، اندازه موقعیت، نام ارز و عملیات (خرید یا فروش).

خروجی:

**dict:** یک دیکشنری که حاوی اطلاعات مربوط به دستور معامله است، از جمله

شماره تیکت سفارش، عملیات معامله (خرید یا فروش) و نتیجه معامله (موفق یا ناموفق).

## تابع Close\_Position:

بستن یا حذف یک موقعیت در متاتریدر 5.

trade\_order: int، شماره تیکت معامله برای بستن یا حذف  
symbol: str، نماد جفت ارز برای معامله  
sleep\_time: int، تعداد ثانیه‌ای که قبل از اجرای عمل معامله منتظر می‌مانیم

## تابع Control\_Position:

کنترل چرخه عمر یک موقعیت در متاتریدر 5.

initialize: list، حاوی اطلاعات ورود به سیستم، رمز عبور، و اطلاعات سرور برای اتصال به ترمینال MT5  
trade\_info: dict، حاوی اطلاعات مربوط به معامله برای باز کردن، شامل جفت ارز، جهت معامله، اندازه لات، استاپ لاس، و تیک پروفیت  
max\_pending\_time: int، حداکثر زمان به ثانیه برای انتظار یک دستور در حال انتظار تا اجرا شود  
max\_open\_time: int، حداکثر زمان به ثانیه برای نگه داشتن یک معامله باز قبل از بستن آن

## تابع Control\_Positions:

تابع Control\_Positions برای کنترل چرخه حیات چندین موقعیت در محیط متاتریدر 5 نوشته شده است. این تابع با دریافت اطلاعات ابتدایی و لیستی از موقعیت‌ها، به مدیریت اقدامات متعددی از جمله باز کردن، لغو کردن یا تغییر موقعیت‌ها می‌پردازد. ابتدا، این تابع منتظر می‌ماند تا یکی از موقعیت‌ها آماده به باز شدن شود و سپس عمل مورد نیاز (به شکل 'Buy'، 'Sell' یا 'Cancel') را تعیین می‌کند.

سپس با بررسی اطلاعات معامله و قیمت‌های بازار، دستورات لازم برای اجرای معامله را تشکیل می‌دهد. در نهایت، با انجام محاسبات مرتبط با ریسک و سود، دستور معامله را به سرور MT5 ارسال کرده و در نهایت موقعیت را بسته یا تغییر می‌دهد. همچنین، این تابع به صورت جزئی از سیستم معاملاتی مبتنی بر خودکار می‌باشد که با استفاده از اطلاعات متنوع و تحلیل‌های مختلف، تصمیمات معاملاتی را انجام می‌دهد.