

تابع PositionSize:

تابع `PositionSize` بر اساس پارامترهای ترید داده شده، اندازه پوزیشن را محاسبه می‌کند. این تابع ابتدا با استفاده از `mt5.symbol_select`() نماد مورد نظر را انتخاب می‌کند و اطلاعات مربوط به اندازه تیک ترید (`trade_tick_size`) و ارزش هر تیک ترید (`trade_tick_value`) را با استفاده از `mt5.symbol_info`() دریافت می‌کند. سپس با استفاده از `mt5.symbol_info_tick`() اطلاعات کنونی تیک برای نماد را دریافت کرده و حداقل فاصله مجاز را محاسبه می‌کند که برابر است با ۵ برابر اندازه تیک.

سپس تعداد تیک‌های در معرض خطر براساس قیمت ورودی و قیمت توقف ضرر محاسبه می‌شود و اندازه پوزیشن براساس حداکثر مقدار سرمایه‌گذاری در ارزش پایه حساب محاسبه می‌شود.

در نهایت، اندازه پوزیشن محاسبه شده (به صورت لات) به عنوان یک عدد اعشاری با دقت دو رقم پس از علامت اعشار، برگردانده می‌شود.

مثال استفاده از این تابع:

```
symbol = 'EURUSD'
entry = 1.2345
sl = 1.2300
risk = 1000 # Maximum amount of capital to risk is $1000
position_size = PositionSize(symbol, entry, sl, risk)
```

تابع Open_Position:

تابع `Open_Position` با گرفتن یک دیکشنری (`dictionary`) حاوی اطلاعات ترید از جمله قیمت ورود، سود دلخواه، توقف ضرر، اندازه پوزیشن، نام ارز و عملیات (خرید یا فروش)، یک سفارش بازار یا سفارش معلق برای بازکردن یک پوزیشن در نماد مالی مشخص شده ایجاد می‌کند.

ورودی‌ها:

- `trade_info` (دیکشنری): یک دیکشنری که اطلاعات ترید را شامل می‌شود، از جمله قیمت ورود، سود دلخواه، توقف ضرر، اندازه پوزیشن، نام ارز و عملیات (خرید یا فروش).

خروجی:

- `dict` (دیکشنری): یک دیکشنری که شامل اطلاعات درباره سفارش ترید است، از جمله شماره بلیط سفارش، عملیات ترید (خرید یا فروش) و نتیجه ترید (موفق یا ناموفق).

تابع ابتدا تعداد رقم‌ها را برای جفت ارزی که در حال معامله است، با استفاده از `mt5.symbol_info(symbol).digits` دریافت می‌کند.

سپس پارامترهای ترید را به تعداد رقم‌های جفت ارزی که در حال معامله است، گرد می‌کند و به میزان معامله (`lot`) اعشار می‌دهد.

سپس فاصله حداقل بین قیمت ورود و سود دلخواه را محاسبه می‌کند که برابر است با مقدار مطلق از تفاوت آنها تقسیم بر ۳.

سپس با توجه به عملیات ترید قیمت bid/ask کنونی را دریافت می‌کند. اگر عملیات "فروش" باشد، قیمت پرس را دریافت می‌کند و در غیر این صورت قیمت بید را دریافت می‌کند.

سپس بررسی می‌کند که آیا قیمت کنونی در فاصله حداقلی از قیمت ورود قرار دارد یا خیر. اگر در این فاصله قرار داشت، یک سفارش بازار ایجاد می‌کند، در غیر این صورت یک سفارش معلق ایجاد می‌کند. نوع سفارش معلق بسته به عملیات ترید و پوزیشن قیمت نسبت به ورودی تعیین می‌شود.

در آخر، سفارش معلق را به سرور ترید ارسال می‌کند و اطلاعات مربوط به سفارش ترید را برمی‌گرداند تا تابع `Open_Position` برای بازکردن یک پوزیشن تریدی به ازای یک ترید ورودی از نماد، قیمت ورود، سود دلخواه، توقف ضرر، اندازه پوزیشن، نام ارز و عملیات (خرید یا فروش) استفاده می‌شود. این تابع ابتدا تعداد رقم‌های نماد معاملاتی را بررسی می‌کند، سپس پارامترهای ترید را به تعداد رقم‌های معاملاتی گرد می‌کند و سپس بررسی می‌کند آیا قیمت کنونی در حداقل فاصله مجاز قرار دارد یا نه. در صورتی که قیمت در فاصله مجاز قرار داشته باشد، یک سفارش بازار ایجاد می‌کند و در غیر این صورت یک سفارش معلق ایجاد می‌شود. سپس سفارش را به سرور ترید ارسال کرده و اطلاعات مربوط به سفارش را برمی‌گرداند.

مثال استفاده از تابع `Open_Position`:

...

```
trade_info = {  
    'Currency': 'EURUSD',  
    'EntryPoint': 1.2345,  
    'TakeProfit': 1.2400,  
    'StepLoss': 1.2300,
```

```
'PositionSize': 0.01,

'Action': 'Buy'

}
```

```
result = Open_Position(trade_info)

print(result)
```

...

این مثال یک ترید با استفاده از نماد 'EURUSD'، قیمت ورود 1.2345، سود دلخواه 1.2400، توقف ضرر 1.2300، اندازه پوزیشن 0.01، نام ارز 'EUR' و عملیات خرید ('Buy') را ایجاد می‌کند. سپس تابع 'Open_Position' فراخوانی می‌شود و نتیجه در متغیر 'result' ذخیره می‌شود. در نهایت، متغیر 'result' چاپ می‌شود.

تابع Close_Position:

تابع 'Close_Position' به منظور بستن یا حذف یک پوزیشن در پلتفرم MetaTrader 5 استفاده می‌شود.

ورودی‌ها:

- 'trade_order' (عدد صحیح): شماره بلیط معامله که باید برای بستن یا حذف آن استفاده شود.
 - 'request' (دیکشنری): شیء درخواست معامله که توسط 'mt5.orders_get()' برای معامله برگشت داده شده است.
 - 'action' (رشته): 'Close' برای بستن معامله و یا 'Remove' برای حذف آن.
 - 'symbol' (رشته): نماد جفت ارزی که برای معامله استفاده شده است.
 - 'sleep_time' (عدد صحیح): تعداد ثانیه‌هایی که قبل از اجرای عملیات معامله باید صبر کنید.
- تابع، با تأخیر مشخص شده توسط 'sleep_time' صبر میکند. سپس بررسی می‌کند که عملیات معامله چه نوع است. اگر 'action' برابر با 'Close' باشد، تابع 'mt5.Close()' را فراخوانی می‌کند تا معامله بسته شود. در غیر این صورت، اگر 'action' برابر با 'Remove' باشد، ابتدا بررسی می‌شود که سود معامله برابر با صفر است یا خیر. اگر سود برابر با صفر باشد، تابع 'mt5.order_send()' را با پارامتر "action": mt5.TRADE_ACTION_REMOVE فراخوانی می‌کند تا معامله حذف شود. در غیر این صورت، تابع 'None' را برمی‌گرداند (معامله حذف نمی‌شود). سپس نتیجه عملیات معامله را به عنوان خروجی برمی‌گرداند.

```
...  
  
trade_order = 12345  
  
request = {'order': 12345, 'volume': 0.01, 'symbol': 'EURUSD', 'type': 0, 'magic': 0, 'time':  
datetime.datetime(2023, 9, 15, 12, 0), 'price': 1.2345, 'sl': 0, 'tp': 0, 'comment': '', 'commission': 0,  
'swap': 0, 'profit': 0, 'taxes': 0, 'margin_rate': 0}  
  
action = 'Close'  
  
symbol = 'EURUSD'  
  
sleep_time = 5  
  
result = Close_Position(trade_order, request, action, symbol, sleep_time)  
  
print(result)  
...
```

در این مثال، تابع `Close_Position` برای بستن یک معامله با شماره بلیط 12345، درخواست معامله `request`، عملیات 'Close'، نماد 'EURUSD' و زمان انتظار 5 ثانیه فراخوانی می‌شود. سپس نتیجه عملیات بستن معامله در متغیر `result` ذخیره می‌شود. در نهایت، متغیر `result` چاپ می‌شود.

تابع Control_Position:

تابع `Control_Position` برای کنترل چرخه یک پوزیشن در پلتفرم MetaTrader 5 استفاده می‌شود.

ورودی‌ها:

- `initialize` (لیست): حاوی اطلاعات ورود به سیستم برای اتصال به ترمینال MT5، از جمله نام کاربری (login)، رمز عبور

(password) و سرور (server).

- `trade_info` (دیکشنری): حاوی اطلاعات مورد نیاز برای باز کردن معامله، از جمله جفت ارز، جهت معامله، اندازه لات، توقف

ضرر و سود دلخواه.

- `max_pending_time` (عدد صحیح): حداکثر زمان در ثانیه برای انتظار برای اجرای یک دستور معلق.

- `max_open_time` (عدد صحیح): حداکثر زمان در ثانیه برای نگهداشتن یک معامله باز قبل از بستن آن.

ابتدا، پس از مقداردهی اولیه، تابع `initialize` فراخوانی شده و ارتباط با پلتفرم MT5 برقرار می‌شود.

سپس، تابع `Open_Position` فراخوانی می‌شود تا یک معامله را با استفاده از اطلاعات موجود در `trade_info` باز کند. نتیجه

این تابع در متغیرهای `trade` و `request` ذخیره می‌شود.

اگر عملیات معامله در حالت معلق (`mt5.TRADE_ACTION_PENDING`) باشد، پیام مناسب چاپ می‌شود و یک

(`thread`) جدید با استفاده از تابع `Close_Position` ایجاد می‌شود. این thread برای حذف معامله با استفاده از

`trade.order` و `request` به همراه پارامترهای مناسب فراخوانی می‌شود. سپس thread شروع می‌شود.

همچنین، یک thread دیگر با استفاده از تابع `Close_Position` ایجاد می‌شود. این thread برای بستن معامله با استفاده از

`trade.order` و `request` به همراه پارامترهای مناسب فراخوانی می‌شود. سپس thread شروع می‌شود.

مثال استفاده از این تابع:

...

```
initialize = ['123456', 'password123', 'MT5_Server']
```

```
trade_info = {'Currency': 'EURUSD', 'EntryPoint': 1.2345, 'TakeProfit': 1.2400, 'StepLoss': 1.2300,  
'PositionSize': 0.01, 'Action': 'Buy'}
```

```
max_pending_time = 120
```

```
max_open_time = 1200
```

```
Control_Position(initialize, trade_info, max_pending_time, max_open_time)
```

...

در مثال ارائه شده، تابع `Control_Position` با استفاده از اطلاعات ورود به سیستم، اطلاعات معامله، حداکثر زمان انتظار برای

اجرای دستور معلق و حداکثر زمان نگهداری معامله باز فراخوانی می‌شود. این تابع به صورت همزمان اجرا می‌شود و دو thread

ایجاد می‌کند تا در صورت لزوم معامله را بسته و یا حذف کند.

