# 可以自己import我们平台支持的第三方python模块，比如pandas、numpy等。

# 在这个方法中编写任何的初始化逻辑。context对象将会在你的算法策略的任何方法之间做传递。

def init(context):

# 在context中保存全局变量

context.s1 = "000001.XSHE"

# 实时打印日志

logger.info("RunInfo: {}".format(context.run\_info))

# before\_trading此函数会在每天策略交易开始前被调用，当天只会被调用一次

def before\_trading(context):

pass

# 你选择的证券的数据更新将会触发此段逻辑，例如日或分钟历史数据切片或者是实时数据切片更新

def handle\_bar(context, bar\_dict):

# 开始编写你的主要的算法逻辑

# bar\_dict[order\_book\_id] 可以拿到某个证券的bar信息

# context.portfolio 可以拿到现在的投资组合信息

# 使用order\_shares(id\_or\_ins, amount)方法进行落单

# TODO: 开始编写你的算法吧！

order\_shares(context.s1, 1000)

# after\_trading函数会在每天交易结束后被调用，当天只会被调用一次

def after\_trading(context):

pass