摘自：《zw 量化实盘·开源课件》系列
更多量化资料,请浏览 zw 网站：http://ziwang.com
QQ 群：124134140 （zw 量化&大数据)

《TALIB 函数大全·ZW 汉化注解版》

zw 开源量化团队 QQ 群：533233771

作者：ZW=智王+字王 2016.01.25

前言

ta-lib v0.49 版，共 178 个函数。网站：http://ta-lib.org/index.html

TA-Lib 是一个经典的股票金融分析软件库，虽然不是 python 原生软件，却提供了 python 接口封装，可以直接使用。

TA-Lib 常简写为 TALIB，搜索时，最好用关键词：TA-Lib，不然数据很少。

python 量化、金融股票数据分析，目前类似的模块库还有：

- Prophet 是一个 Python 的微框架，用于金融市场。Prophet 可以让开发人员把精力放在金融策略模型、项目组合管理和分析上。
- zipline，目前最热的 py 量化回溯软件包，https://github.com/quantopian/zipline

TALIB 虽然有几年没更新（2014 年），也算是经典了的股票金融算法库了，内置了超过 200 种股市技术指标，比如：ADX,MACD,RSI,Stochastic,Bollinger Bands 等。

TALIB 原本是用 C 语言编写的，属于开源软件。

安装时，类似 opencv，需先编译、安装 talib，再安装 python-talib 接口包。

不过，LFD 提供了二进制版本的封装包，一个文件 ok，支持 Py3.5.

整理 talib 文档时，发现，talib 的 180 个函数，其中有六十多个“CDL”开头的函数，全部是关于 K 线图（英文蜡烛图，candle）。

对于 pandas、现代统计分析为主的，量化分析而言，此类函数，基本无用。

（故此，CDL 相关函数，全部打包作为附件）

再去掉 20-30 个算数、辅助函数，大家真正需要学习的 TA-Lib 函数，大约 100 个左右。

对于初学者而言，虽然 TALIB 与 pandas 的集成，相对较弱，不过，TALIB 提供的金融函数，都是最基础的股票技术指标，相对其他量化软件包，更加容易入手。

需注意的是，TA-LIB 与国内股票软件的技术指标，计算方式有较大的差异，例如：

- ATR 的计算，国内一般是取 TR（真实波幅）的简单平均。而 TA-LIB 则是采取类似 EMA 平均一样的方法求 TR 的平均值。
- MACD(12,26,9)的计算，TA-LIB 对于前 33 个初始值是未定义的，国内股软计算初始值时则是根

据已有的几根 bar 计算的平均值比照 MACD 公式进行换算的。

金融函数，例如：BBAND（布林带）、MACD（指数平滑移动平均线），这些函数，背后的技术指标，都有十分复杂的数学、金融理论和实践支持。

本书作为函数手册，无法一一解释，如有疑惑，请大家，参加：zw 量化培训班，进一步学习深造，或者，自己百度查证。

函数库

ACOS 反余弦函数

ACOS　位于模块：　talib.func:

ACOS(...)
　　ACOS(real)

　　矢量三角　ACos (数学变换)

【输入】
　　real: (any ndarray)
【输出】: real

AD 线随机指标

AD　位于模块：　talib.func:
AD(...)
　　AD(high, low, close, volume)

　　Chaikin A/D Line (Volume 指标)

【输入】
　　prices: ['high', 'low', 'close', 'volume']
【输出】: real

ADD 加法

ADD　位于模块：　talib.func:

ADD(...)
    ADD(real0, real1)

    矢量运算　Add (数学运算)

【输入】
        real0: (any ndarray)
        real1: (any ndarray)
【输出】: real

ADOSC 佳庆指标

ADOSC　位于模块：　talib.func:

ADOSC(...)
    ADOSC(high, low, close, volume[, fastperiod=?, slowperiod=?])

    Chaikin A/D Oscillator (Volume 指标)

【输入】
        prices: ['high', 'low', 'close', 'volume']
【参数】
        fastperiod: 3
        slowperiod: 10
【输出】: real

ADX 平均趋向指数

ADX　位于模块：　talib.func:

ADX(...)
    ADX(high, low, close[, timeperiod=?])
    Average Directional Movement Index (动量指标)

【输入】
        prices: ['high', 'low', 'close']
【参数】
        时间周期: 14
【输出】: real

ADXR 评估指数

ADXR 位于模块： talib.func:

ADXR(...)
    ADXR(high, low, close[, timeperiod=?])

    Average Directional Movement Index Rating (动量指标)

【输入】
        prices: ['high', 'low', 'close']
【参数】
        时间周期: 14
【输出】: real

APO 绝对价格振荡指数

APO 位于模块： talib.func:

APO(...)
    APO(real[, fastperiod=?, slowperiod=?, matype=?])

    Absolute Price Oscillator (动量指标)

【输入】
        real: (any ndarray)
【参数】
        fastperiod: 12
        slowperiod: 26
        matype: 0 (Simple Moving Average)
【输出】: real

AROON 阿隆指标

AROON 位于模块： talib.func:

AROON(...)
    AROON(high, low[, timeperiod=?])

Aroon (动量指标)

【输入】
prices: ['high', 'low']
【参数】
时间周期: 14
【输出】
aroondown
aroonup

AROONOSC 阿隆震荡线

阿隆震荡线（Aroon Oscillator）。
AROONOSC 位于模块： talib.func:

AROONOSC(...)
AROONOSC(high, low[, timeperiod=?])

Aroon Oscillator (动量指标)

【输入】
prices: ['high', 'low']
【参数】
时间周期: 14
【输出】: real

ASIN 反正弦函数

ASIN 位于模块： talib.func:

ASIN(...)
ASIN(real)

矢量三角 ASin (数学变换)

【输入】
real: (any ndarray)
【输出】: real

ATAN 反正切函数

ATAN 位于模块： talib.func:

ATAN(...)
    ATAN(real)

    矢量三角 ATan (数学变换)

【输入】
        real: (any ndarray)
【输出】: real

ATR 平均真实波幅

ATR 位于模块： talib.func:

ATR(...)
    ATR(high, low, close[, timeperiod=?])

    Average True Range (波动指标)

【输入】
        prices: ['high', 'low', 'close']
【参数】
        时间周期: 14
【输出】: real

AVGPRICE 平均价格

AVGPRICE 位于模块： talib.func:

AVGPRICE(...)
    AVGPRICE(open, high, low, close)

    Average Price (Price Transform)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】: real

BBANDS 布林带

BBANDS 位于模块： talib.func:

BBANDS(...)

BBANDS(real[, timeperiod=?, nbdevup=?, nbdevdn=?, matype=?])

Bollinger Bands (Overlap Studies)

【输入】

real: (any ndarray)

【参数】

时间周期: 5

nbdevup: 2

nbdevdn: 2

matype: 0 (Simple Moving Average)

【输出】

upperband

middleband

lowerband

BETA 指数

BETA 位于模块： talib.func:

BETA(...)

BETA(real0, real1[, timeperiod=?])

Beta (Statistic Functions)

【输入】

real0: (any ndarray)

real1: (any ndarray)

【参数】

时间周期: 5

【输出】: real

BOP 均势指标

均势指标（Balance Of Power）
BOP　位于模块：　talib.func:

BOP(...)
　　BOP(open, high, low, close)
　　Balance Of Power (动量指标)

【输入】
　　prices: ['open', 'high', 'low', 'close']
【输出】：real

CCI 顺势指标

CCI　位于模块：　talib.func:

CCI(...)
　　CCI(high, low, close[, timeperiod=?])

　　Commodity Channel Index (动量指标)

【输入】
　　prices: ['high', 'low', 'close']
【参数】
　　时间周期: 14
【输出】：real

CEIL 取整函数

返回大于或者等于指定表达式的最小整数

CEIL　位于模块：　talib.func:

CEIL(...)
　　CEIL(real)

　　Vector Ceil (数学变换)

【输入】
　　real: (any ndarray)
【输出】：real

CMO 钱德动量摆动指标

CMO　位于模块：　talib.func:

CMO(...)
　　CMO(real[, timeperiod=?])

　　Chande Momentum Oscillator (动量指标)

【输入】
　　real: (any ndarray)
【参数】
　　时间周期: 14
【输出】：real

CORREL 皮尔森相关系数

皮尔森相关系数（R）

CORREL　位于模块：　talib.func:

CORREL(...)
　　CORREL(real0, real1[, timeperiod=?])

　　Pearson's Correlation Coefficient (r) (Statistic Functions)

【输入】
　　real0: (any ndarray)
　　real1: (any ndarray)
【参数】
　　时间周期: 30
【输出】：real

COS 余弦函数

COS　位于模块：　talib.func:

COS(...)
    COS(real)

    矢量三角　Cos (数学变换)

【输入】
        real: (any ndarray)
【输出】: real

COSH 双曲余弦值

COSH　位于模块：　talib.func:

COSH(...)
    COSH(real)

    矢量三角　Cosh (数学变换)

【输入】
        real: (any ndarray)
【输出】: real

DEMA 双指数移动平均线

DEMA　位于模块：　talib.func:

DEMA(...)
    DEMA(real[, timeperiod=?])

    Double Exponential Moving Average (Overlap Studies)

【输入】
        real: (any ndarray)
【参数】
        时间周期: 30
【输出】: real

DIV 整除

DIV 位于模块： talib.func:

DIV(...)
DIV(real0, real1)

矢量运算 Div (数学运算)

【输入】
real0: (any ndarray)
real1: (any ndarray)
【输出】: real

DX 动向指数

DX 位于模块： talib.func:

DX(...)
DX(high, low, close[, timeperiod=?])

Directional Movement Index (动量指标)

【输入】
prices: ['high', 'low', 'close']
【参数】
时间周期: 14
【输出】: real

EMA 指数移动平均线

EMA 位于模块： talib.func:

EMA(...)
EMA(real[, timeperiod=?])

Exponential Moving Average (Overlap Studies)

【输入】

real: (any ndarray)
【参数】
时间周期: 30
【输出】: real

EXP 指数函数

以自然常数 e 为底的指数函数
EXP 位于模块： talib.func:

EXP(...)
EXP(real)

矢量运算 Exp (数学变换)

【输入】
real: (any ndarray)
【输出】: real

FLOOR 向下取整

其功能是“向下取整”,或者说“向下舍入”,即取不大于 x 的最大整数(与“四舍五入”不同)

FLOOR 位于模块： talib.func:

FLOOR(...)
FLOOR(real)

Vector Floor (数学变换)

【输入】
real: (any ndarray)
【输出】: real

HT_DCPERIOD 希尔伯特变换，主周期

HT_DCPERIOD 位于模块： talib.func:

HT_DCPERIOD(...)
    HT_DCPERIOD(real)

    Hilbert Transform - Dominant Cycle Period (Cycle 指标)

【输入】
        real: (any ndarray)
【输出】: real

HT_DCPHASE 希尔伯特变换,主阶段

HT_DCPHASE 位于模块： talib.func:

HT_DCPHASE(...)
    HT_DCPHASE(real)

    Hilbert Transform - Dominant Cycle Phase (Cycle 指标)

【输入】
        real: (any ndarray)
【输出】: real

HT_PHASOR 希尔伯特变换,相成分

HT_PHASOR 位于模块： talib.func:

HT_PHASOR(...)
    HT_PHASOR(real)

    Hilbert Transform - Phasor Components (Cycle 指标)

【输入】
        real: (any ndarray)
【输出】
        inphase
        quadrature

HT_SINE 希尔伯特变换,正弦波

HT_SINE 位于模块： talib.func:

HT_SINE(...)
    HT_SINE(real)

    Hilbert Transform - SineWave (Cycle 指标)

【输入】
        real: (any ndarray)
【输出】
        sine
        leadsine

HT_TRENDLINE 希尔伯特变换,瞬时趋势

HT_TRENDLINE 位于模块： talib.func:

HT_TRENDLINE(...)
    HT_TRENDLINE(real)

    Hilbert Transform - Instantaneous Trendline (Overlap Studies)

【输入】
        real: (any ndarray)
【输出】: real

HT_TRENDMODE 希尔伯特变换-趋势与周期模式

HT_TRENDMODE 位于模块： talib.func:

HT_TRENDMODE(...)
    HT_TRENDMODE(real)

    Hilbert Transform - Trend vs Cycle Mode (Cycle 指标)

【输入】

real: (any ndarray)

【输出】

integer (values are -100, 0 or 100)

KAMA 适应性移动平均线

KAMA 位于模块： talib.func:

KAMA(...)

KAMA(real[, timeperiod=?])

Kaufman Adaptive Moving Average (Overlap Studies)

【输入】

real: (any ndarray)

【参数】

时间周期: 30

【输出】: real

LINEARREG 线性回归

LINEARREG 位于模块： talib.func:

LINEARREG(...)

LINEARREG(real[, timeperiod=?])

Linear Regression (Statistic Functions)

【输入】

real: (any ndarray)

【参数】

时间周期: 14

【输出】: real

LINEARREG_ANGLE 线性回归的角度

LINEARREG_ANGLE 位于模块： talib.func:

LINEARREG_ANGLE(...)
    LINEARREG_ANGLE(real[, timeperiod=?])

    Linear Regression Angle (Statistic Functions)

【输入】
        real: (any ndarray)
【参数】
        时间周期: 14
【输出】: real

LINEARREG_INTERCEPT 线性回归截距

LINEARREG_INTERCEPT 位于模块： talib.func:

LINEARREG_INTERCEPT(...)
    LINEARREG_INTERCEPT(real[, timeperiod=?])

    Linear Regression Intercept (Statistic Functions)

【输入】
        real: (any ndarray)
【参数】
        时间周期: 14
【输出】: real

LINEARREG_SLOPE 线性回归斜率

LINEARREG_SLOPE 位于模块： talib.func:

LINEARREG_SLOPE(...)
    LINEARREG_SLOPE(real[, timeperiod=?])

    Linear Regression Slope (Statistic Functions)

【输入】
        real: (any ndarray)
【参数】
        时间周期: 14

【输出】：real

LN 自然对数

LN，是以 e 底数的自然对数。
LN 位于模块： talib.func:

LN(...)
    LN(real)

    Vector Log Natural (数学变换)

【输入】
        real: (any ndarray)
【输出】：real

LOG10 对数函数

以 10 底数的对数。
LOG10 位于模块： talib.func:

LOG10(...)
    LOG10(real)

    Vector Log10 (数学变换)

【输入】
        real: (any ndarray)
【输出】：real

MA 移动平均线

MA 位于模块： talib.func:

MA(...)
    MA(real[, timeperiod=?, matype=?])

    Moving average (Overlap Studies)

【输入】
real: (any ndarray)
【参数】
时间周期: 30
matype: 0 (Simple Moving Average)
【输出】: real

MACD 指数平滑移动平均线

MACD 位于模块： talib.func:

MACD(...)
MACD(real[, fastperiod=?, slowperiod=?, signalperiod=?])

Moving Average Convergence/Divergence (动量指标)

【输入】
real: (any ndarray)
【参数】
fastperiod: 12
slowperiod: 26
signalperiod: 9
【输出】
macd
macdsignal
macdhist

MACDEXT ， MA 型可控 MACD

MACDEXT 位于模块： talib.func:

MACDEXT(...)
MACDEXT(real[, fastperiod=?, fastmatype=?, slowperiod=?, slowmatype=?, signalperiod=?, signalmatype=?])

MACD with controllable MA type (动量指标)

【输入】
real: (any ndarray)

【参数】
fastperiod: 12
fastmatype: 0
slowperiod: 26
slowmatype: 0
signalperiod: 9
signalmatype: 0
【输出】
macd
macdsignal
macdhist

MACDFIX，移动平均收敛/发散修复 12 / 26

MACDFIX 位于模块： talib.func:

MACDFIX(...)
MACDFIX(real[, signalperiod=?])

Moving Average Convergence/Divergence Fix 12/26 (动量指标)

【输入】
real: (any ndarray)
【参数】
signalperiod: 9
【输出】
macd
macdsignal
macdhist

MAMA，MESA 移动平均线

MAMA 位于模块： talib.func:

MAMA(...)
MAMA(real[, fastlimit=?, slowlimit=?])

MESA Adaptive Moving Average (Overlap Studies)

【输入】
real: (any ndarray)
【参数】
fastlimit: 0.5
slowlimit: 0.05
【输出】
mama
fama

MAVP，变周期均值

MAVP 位于模块： talib.func:

MAVP(...)
MAVP(real, periods[, minperiod=?, maxperiod=?, matype=?])

Moving average with variable period (Overlap Studies)

【输入】
real: (any ndarray)
periods: (any ndarray)
【参数】
minperiod: 2
maxperiod: 30
matype: 0 (Simple Moving Average)
【输出】: real

MAX 最大值

MAX 位于模块： talib.func:

MAX(...)
MAX(real[, timeperiod=?])

Highest value over a specified period (数学运算)

【输入】
real: (any ndarray)
【参数】

时间周期: 30
【输出】: real

MAXINDEX 最大值索引

MAXINDEX 位于模块： talib.func:

MAXINDEX(...)
MAXINDEX(real[, timeperiod=?])

Index of 最大值 over a specified period (数学运算)

【输入】
real: (any ndarray)
【参数】
时间周期: 30
【输出】
integer (values are -100, 0 or 100)

MA_Type 类别

MA_Type 位于模块： talib.common 对象：
【类定义】
MA_Type(builtins.object)
| 【方法定义】
|
| __getitem__(self, type_)
|
| __init__(self)
|
| --
| 【数据说明】
|
| __dict__
| dictionary for instance variables (if defined)
|
| __weakref__
| list of weak references to the object (if defined)
|
| --
| 【其他数据&属性定义】
|

| DEMA = 3
|
| EMA = 1
|
| KAMA = 6
|
| MAMA = 7
|
| SMA = 0
|
| T3 = 8
|
| TEMA = 4
|
| TRIMA = 5
|
| WMA = 2

MEDPRICE 中位数价格

MEDPRICE 位于模块： talib.func:

MEDPRICE(...)
MEDPRICE(high, low)

Median Price (Price Transform)

【输入】
prices: ['high', 'low']
【输出】: real

MFI 货币流量指数

货币流量指数（Money Flow Index）
MFI 位于模块： talib.func:

MFI(...)
MFI(high, low, close, volume[, timeperiod=?])

Money Flow Index (动量指标)

【输入】
prices: ['high', 'low', 'close', 'volume']
【参数】
时间周期: 14
【输出】: real

MIDPOINT 中点

MIDPOINT 位于模块： talib.func:

MIDPOINT(...)
MIDPOINT(real[, timeperiod=?])

MidPoint over period (Overlap Studies)

【输入】
real: (any ndarray)
【参数】
时间周期: 14
【输出】: real

MIDPRICE 中点价格

MIDPRICE 位于模块： talib.func:

MIDPRICE(...)
MIDPRICE(high, low[, timeperiod=?])

Midpoint Price over period (Overlap Studies)

【输入】
prices: ['high', 'low']
【参数】
时间周期: 14
【输出】: real

MIN 最小值

MIN 位于模块： talib.func:

MIN(...)
MIN(real[, timeperiod=?])

Lowest value over a specified period (数学运算)

【输入】
real: (any ndarray)
【参数】
时间周期: 30
【输出】: real

MININDEX 最小值索引

MININDEX 位于模块： talib.func:

MININDEX(...)
MININDEX(real[, timeperiod=?])

Index of 最小值 over a specified period (数学运算)

【输入】
real: (any ndarray)
【参数】
时间周期: 30
【输出】
integer (values are -100, 0 or 100)

MINMAX 最小最大值

MINMAX 位于模块： talib.func:

MINMAX(...)
MINMAX(real[, timeperiod=?])

Lowest and 最大值 s over a specified period (数学运算)

【输入】
real: (any ndarray)
【参数】
时间周期: 30
【输出】
min
max

MINMAXINDEX 最小最大值索引

MINMAXINDEX 位于模块： talib.func:

MINMAXINDEX(...)
MINMAXINDEX(real[, timeperiod=?])

Indexes of lowest and 最大值 s over a specified period (数学运算)

【输入】
real: (any ndarray)
【参数】
时间周期: 30
【输出】
minidx
maxidx

MINUS_DI 负向指标

MINUS_DI 位于模块： talib.func:

MINUS_DI(...)
MINUS_DI(high, low, close[, timeperiod=?])

Minus Directional Indicator (动量指标)

【输入】
prices: ['high', 'low', 'close']
【参数】
时间周期: 14
【输出】: real

MINUS_DM 负向运动

MINUS_DM 位于模块： talib.func:

MINUS_DM(...)
MINUS_DM(high, low[, timeperiod=?])

Minus Directional Movement (动量指标)

【输入】
prices: ['high', 'low']
【参数】
时间周期: 14
【输出】: real

MOM 动量

MOM 位于模块： talib.func:

MOM(...)
MOM(real[, timeperiod=?])

Momentum (动量指标)

【输入】
real: (any ndarray)
【参数】
时间周期: 10
【输出】: real

MULT 乘法

MULT 位于模块： talib.func:

MULT(...)
MULT(real0, real1)

矢量运算　Mult (数学运算)

【输入】
real0: (any ndarray)
real1: (any ndarray)
【输出】: real

NATR 归一化平均值范围

NATR　位于模块：　talib.func:
NATR(...)
NATR(high, low, close[, timeperiod=?])

Normalized Average True Range (波动指标)

【输入】
prices: ['high', 'low', 'close']
【参数】
时间周期: 14
【输出】: real

OBV 能量潮

OBV　位于模块：　talib.func:

OBV(...)
OBV(real, volume)

On Balance Volume (Volume 指标)

【输入】
real: (any ndarray)
prices: ['volume']
【输出】: real

PLUS_DI 更向指示器

PLUS_DI 位于模块： talib.func:

PLUS_DI(...)
PLUS_DI(high, low, close[, timeperiod=?])

Plus Directional Indicator (动量指标)

【输入】
prices: ['high', 'low', 'close']
【参数】
时间周期: 14
【输出】: real

PLUS_DM 定向运动

PLUS_DM 位于模块： talib.func:

PLUS_DM(...)
PLUS_DM(high, low[, timeperiod=?])

Plus Directional Movement (动量指标)

【输入】
prices: ['high', 'low']
【参数】
时间周期: 14
【输出】: real

PPO 价格振荡百分比

PPO 位于模块： talib.func:

PPO(...)
PPO(real[, fastperiod=?, slowperiod=?, matype=?])

Percentage Price Oscillator (动量指标)

【输入】
real: (any ndarray)
【参数】

fastperiod: 12
slowperiod: 26
matype: 0 (Simple Moving Average)
【输出】：real

ROC 变动率指标

ROC 位于模块： talib.func:

ROC(...)
ROC(real[, timeperiod=?])

Rate of change : ((real/prevPrice)-1)*100 (动量指标)

【输入】
real: (any ndarray)
【参数】
时间周期: 10
【输出】：real

ROCP 价格变化率

ROCP 价格变化率=（价格-前价格）/前价格

ROCP 位于模块： talib.func:

ROCP(...)
ROCP(real[, timeperiod=?])

Rate of change Percentage: (real-prevPrice)/prevPrice (动量指标)

【输入】
real: (any ndarray)
【参数】
时间周期: 10
【输出】：real

ROCR 价格变化率

ROCR 变化率=价格/前格

ROCR　位于模块：　talib.func:

ROCR(...)
　　ROCR(real[, timeperiod=?])

　　Rate of change ratio: (real/prevPrice) (动量指标)

【输入】
　　real: (any ndarray)
【参数】
　　时间周期: 10
【输出】：real

ROCR100 价格变化率

ROCR100 变化率=价格/前格*100

ROCR100　位于模块：　talib.func:

ROCR100(...)
　　ROCR100(real[, timeperiod=?])

　　Rate of change ratio 100 scale: (real/prevPrice)*100 (动量指标)

【输入】
　　real: (any ndarray)
【参数】
　　时间周期: 10
【输出】：real

RSI 相对强弱指标

RSI　位于模块：　talib.func:

RSI(...)

RSI(real[, timeperiod=?])

Relative Strength Index (动量指标)

【输入】
real: (any ndarray)
【参数】
时间周期: 14
【输出】: real

SAR 抛物线转向

SAR 位于模块： talib.func:

SAR(...)
SAR(high, low[, acceleration=?, maximum=?])

Parabolic SAR (Overlap Studies)

【输入】
prices: ['high', 'low']
【参数】
acceleration: 0.02
maximum: 0.2
【输出】: real

SAREXT 增强型抛物线转向

SAREXT 位于模块： talib.func:

SAREXT(...)
SAREXT(high, low[, startvalue=?, offsetonreverse=?, accelerationinitlong=?, accelerationlong=?, accelerationmaxlong=?, accelerationinitshort=?, accelerationshort=?, accelerationmaxshort=?])

Parabolic SAR - Extended (Overlap Studies)

【输入】
prices: ['high', 'low']
【参数】
startvalue: 0
offsetonreverse: 0

accelerationinitlong: 0.02
accelerationlong: 0.02
accelerationmaxlong: 0.2
accelerationinitshort: 0.02
accelerationshort: 0.02
accelerationmaxshort: 0.2

【输出】: real

SIN 正弦值

SIN 位于模块： talib.func:

SIN(...)
SIN(real)

矢量三角 Sin (数学变换)

【输入】
real: (any ndarray)
【输出】: real

SINH 双曲正弦函数

SINH 位于模块： talib.func:

SINH(...)
SINH(real)

矢量三角 Sinh (数学变换)

【输入】
real: (any ndarray)
【输出】: real

SMA 简单移动平均

SMA 位于模块： talib.func:

SMA(...)
SMA(real[, timeperiod=?])

Simple Moving Average (Overlap Studies)

【输入】
real: (any ndarray)
【参数】
时间周期: 30
【输出】: real

SQRT 平方根

SQRT 位于模块： talib.func:

SQRT(...)
SQRT(real)

Vector Square Root (数学变换)

【输入】
real: (any ndarray)
【输出】: real

STDDEV 标准偏差

标准偏差(Std Dev,Standard Deviation) -统计学名词。一种量度数据分布的分散程度之标准，用以衡量数据值偏离算术平均值的程度。标准偏差越小，这些值偏离平均值就越少，反之亦然。标准偏差的大小可通过标准偏差与平均值的倍率关系来衡量。

STDDEV 位于模块： talib.func:

STDDEV(...)
STDDEV(real[, timeperiod=?, nbdev=?])

Standard Deviation (Statistic Functions)

【输入】
real: (any ndarray)

【参数】
时间周期: 5
nbdev: 1
【输出】: real

STOCH 指标

STOCH 位于模块： talib.func:

STOCH(...)
STOCH(high, low, close[, fastk_period=?, slowk_period=?, slowk_matype=?, slowd_period=?, slowd_matype=?])

Stochastic (动量指标)

【输入】
prices: ['high', 'low', 'close']
【参数】
fastk_period: 5
slowk_period: 3
slowk_matype: 0
slowd_period: 3
slowd_matype: 0
【输出】
slowk
slowd

STOCHF，快速 STOCH 指标

STOCHF 位于模块： talib.func:

STOCHF(...)
STOCHF(high, low, close[, fastk_period=?, fastd_period=?, fastd_matype=?])

Stochastic Fast (动量指标)

【输入】
prices: ['high', 'low', 'close']
【参数】
fastk_period: 5
fastd_period: 3

fastd_matype: 0

【输出】

fastk

fastd

STOCHRSI，随机强弱指数

STOCHRSI 位于模块： talib.func:

STOCHRSI(...)

STOCHRSI(real[, timeperiod=?, fastk_period=?, fastd_period=?, fastd_matype=?])

Stochastic Relative Strength Index (动量指标)

【输入】

real: (any ndarray)

【参数】

时间周期: 14

fastk_period: 5

fastd_period: 3

fastd_matype: 0

【输出】

fastk

fastd

SUB 减法

SUB 位于模块： talib.func:

SUB(...)

SUB(real0, real1)

矢量运算 Substraction (数学运算)

【输入】

real0: (any ndarray)

real1: (any ndarray)

【输出】: real

SUM 求和

SUM 位于模块： talib.func:

SUM(...)
SUM(real[, timeperiod=?])

Summation (数学运算)

【输入】
real: (any ndarray)
【参数】
时间周期: 30
【输出】: real

T3

三指数移动平均（T3）

T3 位于模块： talib.func:

T3(...)
T3(real[, timeperiod=?, vfactor=?])

Triple Exponential Moving Average (T3) (Overlap Studies)

【输入】
real: (any ndarray)
【参数】
时间周期: 5
vfactor: 0.7
【输出】: real

TAN 正切

TAN 位于模块： talib.func:

TAN(...)
TAN(real)

矢量三角　Tan (数学变换)

【输入】
real: (any ndarray)
【输出】: real

TANH 双曲正切函数

TANH　位于模块：　talib.func:

TANH(...)
TANH(real)

矢量三角　Tanh (数学变换)

【输入】
real: (any ndarray)
【输出】: real

TEMA 三指数移动平均

TEMA　位于模块：　talib.func:

TEMA(...)
TEMA(real[, timeperiod=?])

Triple Exponential Moving Average (Overlap Studies)

【输入】
real: (any ndarray)
【参数】
时间周期: 30
【输出】: real

TRANGE 真实范围

真实范围（True Range）

TRANGE 位于模块： talib.func:

TRANGE(...)
    TRANGE(high, low, close)

    True Range (波动指标)

【输入】
    prices: ['high', 'low', 'close']
【输出】: real

TRIMA 三指数移动平均

TRIMA 位于模块： talib.func:

TRIMA(...)
    TRIMA(real[, timeperiod=?])

    Triangular Moving Average (Overlap Studies)

【输入】
    real: (any ndarray)
【参数】
    时间周期: 30
【输出】: real

TRIX 三重指数平滑平均线

TRIX 位于模块： talib.func:

TRIX(...)
    TRIX(real[, timeperiod=?])

    1-day Rate-Of-Change (ROC) of a Triple Smooth EMA (动量指标)

【输入】
    real: (any ndarray)
【参数】
    时间周期: 30

【输出】：real

TSF 时间序列预测

TSF 位于模块： talib.func:

TSF(...)
    TSF(real[, timeperiod=?])

    Time Series Forecast (Statistic Functions)

【输入】
        real: (any ndarray)
【参数】
        时间周期: 14
【输出】：real

TYPPRICE 典型价格

TYPPRICE 位于模块： talib.func:

TYPPRICE(...)
    TYPPRICE(high, low, close)

    Typical Price (Price Transform)

【输入】
        prices: ['high', 'low', 'close']
【输出】：real

ULTOSC 极限振子

ULTOSC 位于模块： talib.func:

ULTOSC(...)

ULTOSC(high, low, close[, timeperiod1=?, timeperiod2=?, timeperiod3=?])

Ultimate Oscillator (动量指标)

【输入】
prices: ['high', 'low', 'close']
【参数】
timeperiod1: 7
timeperiod2: 14
timeperiod3: 28
【输出】: real

VAR 变量定义

VAR 位于模块： talib.func:

VAR(...)
VAR(real[, timeperiod=?, nbdev=?])

Variance (Statistic Functions)

【输入】
real: (any ndarray)
【参数】
时间周期: 5
nbdev: 1
【输出】: real

WCLPRICE 加权收盘价

WCLPRICE 位于模块： talib.func:

WCLPRICE(...)
WCLPRICE(high, low, close)

Weighted Close Price (Price Transform)

【输入】
prices: ['high', 'low', 'close']
【输出】: real

WILLR 威廉指标

WILLR 位于模块： talib.func:

WILLR(...)
    WILLR(high, low, close[, timeperiod=?])

    Williams' %R (动量指标)

【输入】
        prices: ['high', 'low', 'close']
【参数】
        时间周期: 14
【输出】: real

WMA 加权移动平均

WMA 位于模块： talib.func:

WMA(...)
    WMA(real[, timeperiod=?])

    Weighted Moving Average (Overlap Studies)

【输入】
        real: (any ndarray)
【参数】
        时间周期: 30
【输出】: real

__all__

list 对象：
【类定义】
list(object)
| list() -> new empty list
| list(iterable) -> new list initialized from iterable's items
|
| 【方法定义】
|
| __add__(self, value, /),【返回值】: self+value.
|
| __contains__(self, key, /),【返回值】: key in self.
|
| __delitem__(self, key, /)
| Delete self[key].
|
| __eq__(self, value, /),【返回值】: self==value.
|
| __ge__(self, value, /),【返回值】: self>=value.
|
| __getattribute__(self, name, /),【返回值】: getattr(self, name).
|
| __getitem__(...)
| x.__getitem__(y) <==> x[y]
|
| __gt__(self, value, /),【返回值】: self>value.
|
| __iadd__(self, value, /)
| Implement self+=value.
|
| __imul__(self, value, /)
| Implement self*=value.
|
| __init__(self, /, *args, **kwargs)
| Initialize self. See help(type(self)) for accurate signature.
|
| __iter__(self, /)
| Implement iter(self).
|
| __le__(self, value, /),【返回值】: self<=value.

|
| __len__(self, /),【返回值】: len(self).
|
| __lt__(self, value, /),【返回值】: self<value.
|
| __mul__(self, value, /),【返回值】: self*value.n
|
| __ne__(self, value, /),【返回值】: self!=value.
|
| __new__(*args, **kwargs) from builtins.type
| Create and return a new object. See help(type) for accurate signature.
|
| __repr__(self, /),【返回值】: repr(self).
|
| __reversed__(...)
| L.__reversed__() -- return a reverse iterator over the list
|
| __rmul__(self, value, /),【返回值】: self*value.
|
| __setitem__(self, key, value, /)
| Set self[key] to value.
|
| __sizeof__(...)
| L.__sizeof__() -- size of L in memory, in bytes
|
| append(...)
| L.append(object) -> None -- append object to end
|
| clear(...)
| L.clear() -> None -- remove all items from L
|
| copy(...)
| L.copy() -> list -- a shallow copy of L
|
| count(...)
| L.count(value) -> integer -- return number of occurrences of value
|
| extend(...)
| L.extend(iterable) -> None -- extend list by appending elements from the iterable
|
| index(...)
| L.index(value, [start, [stop]]) -> integer -- return first index of value.
| Raises ValueError if the value is not present.
|
| insert(...)

```
|       L.insert(index, object) -- insert object before index
|
|   pop(...)
|       L.pop([index]) -> item -- remove and return item at index (default last).
|       Raises IndexError if list is empty or index is out of range.
|
|   remove(...)
|       L.remove(value) -> None -- remove first occurrence of value.
|       Raises ValueError if the value is not present.
|
|   reverse(...)
|       L.reverse() -- reverse *IN PLACE*
|
|   sort(...)
|       L.sort(key=None, reverse=False) -> None -- stable sort *IN PLACE*
|
|   ----------------------------------------------------------------------
|   【其他数据&属性定义】
|
|   __hash__ = None
```

__builtins__内建函数

dict 对象：

【类定义】

```
dict(object)
|   dict() -> new empty dictionary
|   dict(mapping) -> new dictionary initialized from a mapping object's
|       (key, value) pairs
|   dict(iterable) -> new dictionary initialized as if via:
|       d = {}
|       for k, v in iterable:
|           d[k] = v
|   dict(**kwargs) -> new dictionary initialized with the name=value pairs
|       in the keyword argument list.  For example:  dict(one=1, two=2)
|
|   【方法定义】
|
|   __contains__(self, key, /)
|       True if D has a key k, else False.
|
|   __delitem__(self, key, /)
```

```
|       Delete self[key].
|
|   __eq__(self, value, /),【返回值】: self==value.
|
|   __ge__(self, value, /),【返回值】: self>=value.
|
|   __getattribute__(self, name, /),【返回值】: getattr(self, name).
|
|   __getitem__(...)
|       x.__getitem__(y) <==> x[y]
|
|   __gt__(self, value, /),【返回值】: self>value.
|
|   __init__(self, /, *args, **kwargs)
|       Initialize self.  See help(type(self)) for accurate signature.
|
|   __iter__(self, /)
|       Implement iter(self).
|
|   __le__(self, value, /),【返回值】: self<=value.
|
|   __len__(self, /),【返回值】: len(self).
|
|   __lt__(self, value, /),【返回值】: self<value.
|
|   __ne__(self, value, /),【返回值】: self!=value.
|
|   __new__(*args, **kwargs) from builtins.type
|       Create and return a new object.  See help(type) for accurate signature.
|
|   __repr__(self, /),【返回值】: repr(self).
|
|   __setitem__(self, key, value, /)
|       Set self[key] to value.
|
|   __sizeof__(...)
|       D.__sizeof__() -> size of D in memory, in bytes
|
|   clear(...)
|       D.clear() -> None.  Remove all items from D.
|
|   copy(...)
|       D.copy() -> a shallow copy of D
|
|   fromkeys(iterable, value=None, /) from builtins.type
```

```
|       Returns a new dict with keys from iterable and values equal to value.
|
|   get(...)
|       D.get(k[,d]) -> D[k] if k in D, else d.    d defaults to None.
|
|   items(...)
|       D.items() -> a set-like object providing a view on D's items
|
|   keys(...)
|       D.keys() -> a set-like object providing a view on D's keys
|
|   pop(...)
|       D.pop(k[,d]) -> v, remove specified key and return the corresponding value.
|       If key is not found, d is returned if given, otherwise KeyError is raised
|
|   popitem(...)
|       D.popitem() -> (k, v), remove and return some (key, value) pair as a
|       2-tuple; but raise KeyError if D is empty.
|
|   setdefault(...)
|       D.setdefault(k[,d]) -> D.get(k,d), also set D[k]=d if k not in D
|
|   update(...)
|       D.update([E, ]**F) -> None.    Update D from dict/iterable E and F.
|       If E is present and has a .keys() method, then does:    for k in E: D[k] = E[k]
|       If E is present and lacks a .keys() method, then does:    for k, v in E: D[k] = v
|       In either case, this is followed by: for k in F:    D[k] = F[k]
|
|   values(...)
|       D.values() -> an object providing a view on D's values
|
|   ----------------------------------------------------------------------
|   【其他数据&属性定义】
|
|   __hash__ = None
```

__cached__

未找到文档：\talib__pycache____init__.cpython-35.pyc'.

__doc__ 说明文档

NoneType 对象：
【类定义】
NoneType(object)
 | 【方法定义】
 |
 | __bool__(self, /)
 | self != 0
 |
 | __new__(*args, **kwargs) from builtins.type
 | Create and return a new object. See help(type) for accurate signature.
 |
 | __repr__(self, /),【返回值】：repr(self).

__file__

未找到文档：\talib__init__.py'.

__function_groups__

dict 对象：
【类定义】
dict(object)
 | dict() -> new empty dictionary
 | dict(mapping) -> new dictionary initialized from a mapping object's
 | (key, value) pairs
 | dict(iterable) -> new dictionary initialized as if via:
 | d = {}
 | for k, v in iterable:
 | d[k] = v
 | dict(**kwargs) -> new dictionary initialized with the name=value pairs
 | in the keyword argument list. For example: dict(one=1, two=2)
 |
 | 【方法定义】
 |
 | __contains__(self, key, /)

| True if D has a key k, else False.
|
| __delitem__(self, key, /)
| Delete self[key].
|
| __eq__(self, value, /),【返回值】: self==value.
|
| __ge__(self, value, /),【返回值】: self>=value.
|
| __getattribute__(self, name, /),【返回值】: getattr(self, name).
|
| __getitem__(...)
| x.__getitem__(y) <==> x[y]
|
| __gt__(self, value, /),【返回值】: self>value.
|
| __init__(self, /, *args, **kwargs)
| Initialize self. See help(type(self)) for accurate signature.
|
| __iter__(self, /)
| Implement iter(self).
|
| __le__(self, value, /),【返回值】: self<=value.
|
| __len__(self, /),【返回值】: len(self).
|
| __lt__(self, value, /),【返回值】: self<value.
|
| __ne__(self, value, /),【返回值】: self!=value.
|
| __new__(*args, **kwargs) from builtins.type
| Create and return a new object. See help(type) for accurate signature.
|
| __repr__(self, /),【返回值】: repr(self).
|
| __setitem__(self, key, value, /)
| Set self[key] to value.
|
| __sizeof__(...)
| D.__sizeof__() -> size of D in memory, in bytes
|
| clear(...)
| D.clear() -> None. Remove all items from D.
|
| copy(...)

```
|       D.copy() -> a shallow copy of D
|
|   fromkeys(iterable, value=None, /) from builtins.type
|       Returns a new dict with keys from iterable and values equal to value.
|
|   get(...)
|       D.get(k[,d]) -> D[k] if k in D, else d.    d defaults to None.
|
|   items(...)
|       D.items() -> a set-like object providing a view on D's items
|
|   keys(...)
|       D.keys() -> a set-like object providing a view on D's keys
|
|   pop(...)
|       D.pop(k[,d]) -> v, remove specified key and return the corresponding value.
|       If key is not found, d is returned if given, otherwise KeyError is raised
|
|   popitem(...)
|       D.popitem() -> (k, v), remove and return some (key, value) pair as a
|       2-tuple; but raise KeyError if D is empty.
|
|   setdefault(...)
|       D.setdefault(k[,d]) -> D.get(k,d), also set D[k]=d if k not in D
|
|   update(...)
|       D.update([E, ]**F) -> None.    Update D from dict/iterable E and F.
|       If E is present and has a .keys() method, then does:    for k in E: D[k] = E[k]
|       If E is present and lacks a .keys() method, then does:    for k, v in E: D[k] = v
|       In either case, this is followed by: for k in F:    D[k] = F[k]
|
|   values(...)
|       D.values() -> an object providing a view on D's values
|
|   ----------------------------------------------------------------------
|   【其他数据&属性定义】
|
|   __hash__ = None
```

__loader__

SourceFileLoader in module importlib._bootstrap_external 对象：
【类定义】
SourceFileLoader(FileLoader, SourceLoader)
| Concrete implementation of SourceLoader using the file system.
|
| 【方法定义顺序】:
| SourceFileLoader
| FileLoader
| SourceLoader
| _LoaderBasics
| builtins.object
|
| 【方法定义】
|
| path_stats(self, path),【返回值】: the metadata for the path.
|
| set_data(self, path, data, *, _mode=438)
| Write bytes data to a file.
|
| ----------
| 方法继承： FileLoader:
|
| __eq__(self, other),【返回值】: self==value.
|
| __hash__(self),【返回值】: hash(self).
|
| __init__(self, fullname, path)
| Cache the module name and the path to the file found by the finder.
|
| get_data(self, path),【返回值】: the data from path as raw bytes.
|
| get_filename(self, name=None, *args, **kwargs),【返回值】: the path to the source file as found by the finder.
|
| load_module(self, name=None, *args, **kwargs)
| Load a module from a file.
|
| This method is deprecated. Use exec_module() instead.
|
| ----------
| Data descriptors inherited from FileLoader:
|

```
|  __dict__
|      dictionary for instance variables (if defined)
|
|  __weakref__
|      list of weak references to the object (if defined)
|
|  ----------------------------------------------------------------------
|  方法继承：  SourceLoader:
|
|  get_code(self, fullname)
|      Concrete implementation of InspectLoader.get_code.
|
|      Reading of bytecode requires path_stats to be implemented. To write
|      bytecode, set_data must also be implemented.
|
|  get_source(self, fullname)
|      Concrete implementation of InspectLoader.get_source.
|
|  path_mtime(self, path)
|      Optional method that returns the modification time (an int) for the
|      specified path, where path is a str.
|
|      Raises IOError when the path cannot be handled.
|
|  source_to_code(self, data, path, *, _optimize=-1),【返回值】：the code object compiled from source.
|
|      The 'data' argument can be any object type that compile() supports.
|
|  ----------------------------------------------------------------------
|  方法继承：  _LoaderBasics:
|
|  create_module(self, spec)
|      Use default semantics for module creation.
|
|  exec_module(self, module)
|      Execute the module.
|
|  is_package(self, fullname)
|      Concrete implementation of InspectLoader.is_package by checking if
|      the path returned by get_filename has a filename of '__init__.py'.
```

__name__

package talib:

【名称】
talib

【模块包内容】
abstract
common
deprecated
func
test_abstract
test_data
test_func

【函数】
get_function_groups()
Returns a dict with keys of function-group names and values of lists
of function names ie {'group_names': ['function_names']}

get_functions()
Returns a list of all the functions supported by TALIB

【数据】
__all__ = ['get_functions', 'get_function_groups']
__function_groups__ = {'Cycle Indicators': ['HT_DCPERIOD', 'HT_DCPHASE...
__ta_version__ = b'0.4.0 (Sep 10 2015 14:20:56)'

【版本】: 0.4.9

【文件】: \talib__init__.py

__package__

【名称】 talib

【模块包内容】
abstract
common

deprecated
func
test_abstract
test_data
test_func

【函数】
get_function_groups()
Returns a dict with keys of function-group names and values of lists
of function names ie {'group_names': ['function_names']}

get_functions()
Returns a list of all the functions supported by TALIB

【数据】
__all__ = ['get_functions', 'get_function_groups']
__function_groups__ = {'Cycle Indicators': ['HT_DCPERIOD', 'HT_DCPHASE...
__ta_version__ = b'0.4.0 (Sep 10 2015 14:20:56)'

【版本】: 0.4.9

【文件】: \talib__init__.py

__path__

list 对象:
【类定义】
list(object)
| list() -> new empty list
| list(iterable) -> new list initialized from iterable's items
|
| 【方法定义】
|
| __add__(self, value, /),【返回值】: self+value.
|
| __contains__(self, key, /),【返回值】: key in self.
|
| __delitem__(self, key, /)
| Delete self[key].
|
| __eq__(self, value, /),【返回值】: self==value.
|

| __ge__(self, value, /),【返回值】: self>=value.
|
| __getattribute__(self, name, /),【返回值】: getattr(self, name).
|
| __getitem__(...)
| x.__getitem__(y) <==> x[y]
|
| __gt__(self, value, /),【返回值】: self>value.
|
| __iadd__(self, value, /)
| Implement self+=value.
|
| __imul__(self, value, /)
| Implement self*=value.
|
| __init__(self, /, *args, **kwargs)
| Initialize self. See help(type(self)) for accurate signature.
|
| __iter__(self, /)
| Implement iter(self).
|
| __le__(self, value, /),【返回值】: self<=value.
|
| __len__(self, /),【返回值】: len(self).
|
| __lt__(self, value, /),【返回值】: self<value.
|
| __mul__(self, value, /),【返回值】: self*value.n
|
| __ne__(self, value, /),【返回值】: self!=value.
|
| __new__(*args, **kwargs) from builtins.type
| Create and return a new object. See help(type) for accurate signature.
|
| __repr__(self, /),【返回值】: repr(self).
|
| __reversed__(...)
| L.__reversed__() -- return a reverse iterator over the list
|
| __rmul__(self, value, /),【返回值】: self*value.
|
| __setitem__(self, key, value, /)
| Set self[key] to value.
|
| __sizeof__(...)

```
|       L.__sizeof__() -- size of L in memory, in bytes
|
|   append(...)
|       L.append(object) -> None -- append object to end
|
|   clear(...)
|       L.clear() -> None -- remove all items from L
|
|   copy(...)
|       L.copy() -> list -- a shallow copy of L
|
|   count(...)
|       L.count(value) -> integer -- return number of occurrences of value
|
|   extend(...)
|       L.extend(iterable) -> None -- extend list by appending elements from the iterable
|
|   index(...)
|       L.index(value, [start, [stop]]) -> integer -- return first index of value.
|       Raises ValueError if the value is not present.
|
|   insert(...)
|       L.insert(index, object) -- insert object before index
|
|   pop(...)
|       L.pop([index]) -> item -- remove and return item at index (default last).
|       Raises IndexError if list is empty or index is out of range.
|
|   remove(...)
|       L.remove(value) -> None -- remove first occurrence of value.
|       Raises ValueError if the value is not present.
|
|   reverse(...)
|       L.reverse() -- reverse *IN PLACE*
|
|   sort(...)
|       L.sort(key=None, reverse=False) -> None -- stable sort *IN PLACE*
|
|   ----------------------------------------------------------------------
|   【其他数据&属性定义】
|
|   __hash__ = None
```

__spec__

ModuleSpec in module importlib._bootstrap 对象：
【类定义】
ModuleSpec(builtins.object)
| The specification for a module, used for loading.
|
| A module's spec is the source for information about the module. For
| data associated with the module, including source, use the spec's
| loader.
|
| `name` is the absolute name of the module. `loader` is the loader
| to use when loading the module. `parent` is the name of the
| package the module is in. The parent is derived from the name.
|
| `is_package` determines if the module is considered a package or
| not. On modules this is reflected by the `__path__` attribute.
|
| `origin` is the specific location used by the loader from which to
| load the module, if that information is available. When filename is
| set, origin will match.
|
| `has_location` indicates that a spec's "origin" reflects a location.
| When this is True, `__file__` attribute of the module is set.
|
| `cached` is the location of the cached bytecode file, if any. It
| corresponds to the `__cached__` attribute.
|
| `submodule_search_locations` is the sequence of path entries to
| search when importing submodules. If set, is_package should be
| True--and False otherwise.
|
| Packages are simply modules that (may) have submodules. If a spec
| has a non-None value in `submodule_search_locations`, the import
| system will consider modules loaded from the spec as packages.
|
| Only finders (see importlib.abc.MetaPathFinder and
| importlib.abc.PathEntryFinder) should modify ModuleSpec instances.
|
| 【方法定义】
|
| __eq__(self, other),【返回值】: self==value.
|

```
|  __init__(self, name, loader, *, origin=None, loader_state=None, is_package=None)
|      Initialize self.  See help(type(self)) for accurate signature.
|
|  __repr__(self),【返回值】: repr(self).
|
|  ----------------------------------------------------------------------
|  【数据说明】
|
|  __dict__
|      dictionary for instance variables (if defined)
|
|  __weakref__
|      list of weak references to the object (if defined)
|
|  cached
|
|  has_location
|
|  parent
|      The name of the module's parent.
|
|  ----------------------------------------------------------------------
|  【其他数据&属性定义】
|
|  __hash__ = None
```

__ta_version__

bytes 对象：

【类定义】

```
bytes(object)
|  bytes(iterable_of_ints) -> bytes
|  bytes(string, encoding[, errors]) -> bytes
|  bytes(bytes_or_buffer) -> immutable copy of bytes_or_buffer
|  bytes(int) -> bytes object of size given by the parameter initialized with null bytes
|  bytes() -> empty bytes object
|
|  Construct an immutable array of bytes from:
|    - an iterable yielding integers in range(256)
|    - a text string encoded using the specified encoding
|    - any object implementing the buffer API.
|    - an integer
```

|
| 【方法定义】
|
| __add__(self, value, /),【返回值】: self+value.
|
| __contains__(self, key, /),【返回值】: key in self.
|
| __eq__(self, value, /),【返回值】: self==value.
|
| __ge__(self, value, /),【返回值】: self>=value.
|
| __getattribute__(self, name, /),【返回值】: getattr(self, name).
|
| __getitem__(self, key, /),【返回值】: self[key].
|
| __getnewargs__(...)
|
| __gt__(self, value, /),【返回值】: self>value.
|
| __hash__(self, /),【返回值】: hash(self).
|
| __iter__(self, /)
| Implement iter(self).
|
| __le__(self, value, /),【返回值】: self<=value.
|
| __len__(self, /),【返回值】: len(self).
|
| __lt__(self, value, /),【返回值】: self<value.
|
| __mod__(self, value, /),【返回值】: self%value.
|
| __mul__(self, value, /),【返回值】: self*value.n
|
| __ne__(self, value, /),【返回值】: self!=value.
|
| __new__(*args, **kwargs) from builtins.type
| Create and return a new object. See help(type) for accurate signature.
|
| __repr__(self, /),【返回值】: repr(self).
|
| __rmod__(self, value, /),【返回值】: value%self.
|
| __rmul__(self, value, /),【返回值】: self*value.
|

```
|  __str__(self, /),【返回值】: str(self).
|
|  capitalize(...)
|      B.capitalize() -> copy of B
|      ,【返回值】: a copy of B with only its first character capitalized (ASCII)
|      and the rest lower-cased.
|
|  center(...)
|      B.center(width[, fillchar]) -> copy of B
|      ,【返回值】: B centered in a string of length width.  Padding is
|      done using the specified fill character (default is a space).
|
|  count(...)
|      B.count(sub[, start[, end]]) -> int
|      ,【返回值】: the number of non-overlapping occurrences of substring sub in
|      string B[start:end].  Optional arguments start and end are interpreted
|      as in slice notation.
|
|  decode(self, /, encoding='utf-8', errors='strict')
|      Decode the bytes using the codec registered for encoding.
|
|      encoding
|        The encoding with which to decode the bytes.
|      errors
|        The error handling scheme to use for the handling of decoding errors.
|        The default is 'strict' meaning that decoding errors raise a
|        UnicodeDecodeError. Other possible values are 'ignore' and 'replace'
|        as well as any other name registered with codecs.register_error that
|        can handle UnicodeDecodeErrors.
|
|  endswith(...)
|      B.endswith(suffix[, start[, end]]) -> bool
|      ,【返回值】: True if B ends with the specified suffix, False otherwise.
|      With optional start, test B beginning at that position.
|      With optional end, stop comparing B at that position.
|      suffix can also be a tuple of bytes to try.
|
|  expandtabs(...)
|      B.expandtabs(tabsize=8) -> copy of B
|      ,【返回值】: a copy of B where all tab characters are expanded using spaces.
|      If tabsize is not given, a tab size of 8 characters is assumed.
|
|  find(...)
|      B.find(sub[, start[, end]]) -> int
|      ,【返回值】: the lowest index in B where substring sub is found,
```

```
|       such that sub is contained within B[start:end].    Optional
|       arguments start and end are interpreted as in slice notation.
|       ,【返回值】: -1 on failure.
|
|   fromhex(string, /) from builtins.type
|       Create a bytes object from a string of hexadecimal numbers.
|
|       Spaces between two numbers are accepted.
|       Example: bytes.fromhex('B9 01EF') -> b'\xb9\x01\xef'.
|
|   hex(...)
|       B.hex() -> string
|
|       Create a string of hexadecimal numbers from a bytes object.
|       Example: b'\xb9\x01\xef'.hex() -> 'b901ef'.
|
|   index(...)
|       B.index(sub[, start[, end]]) -> int
|
|       Like B.find() but raise ValueError when the substring is not found.
|
|   isalnum(...)
|       B.isalnum() -> bool
|       ,【返回值】: True if all characters in B are alphanumeric
|       and there is at least one character in B, False otherwise.
|
|   isalpha(...)
|       B.isalpha() -> bool
|       ,【返回值】: True if all characters in B are alphabetic
|       and there is at least one character in B, False otherwise.
|
|   isdigit(...)
|       B.isdigit() -> bool
|       ,【返回值】: True if all characters in B are digits
|       and there is at least one character in B, False otherwise.
|
|   islower(...)
|       B.islower() -> bool
|       ,【返回值】: True if all cased characters in B are lowercase and there is
|       at least one cased character in B, False otherwise.
|
|   isspace(...)
|       B.isspace() -> bool
|       ,【返回值】: True if all characters in B are whitespace
|       and there is at least one character in B, False otherwise.
```

```
|
|  istitle(...)
|      B.istitle() -> bool
|      ,【返回值】: True if B is a titlecased string and there is at least one
|      character in B, i.e. uppercase characters may only follow uncased
|      characters and lowercase characters only cased ones. Return False
|      otherwise.
|
|  isupper(...)
|      B.isupper() -> bool
|      ,【返回值】: True if all cased characters in B are uppercase and there is
|      at least one cased character in B, False otherwise.
|
|  join(self, iterable_of_bytes, /)
|      Concatenate any number of bytes objects.
|
|      The bytes whose method is called is inserted in between each pair.
|
|      The result is returned as a new bytes object.
|
|      Example: b'.'.join([b'ab', b'pq', b'rs']) -> b'ab.pq.rs'.
|
|  ljust(...)
|      B.ljust(width[, fillchar]) -> copy of B
|      ,【返回值】: B left justified in a string of length width. Padding is
|      done using the specified fill character (default is a space).
|
|  lower(...)
|      B.lower() -> copy of B
|      ,【返回值】: a copy of B with all ASCII characters converted to lowercase.
|
|  lstrip(self, bytes=None, /)
|      Strip leading bytes contained in the argument.
|
|      If the argument is omitted or None, strip leading  ASCII whitespace.
|
|  partition(self, sep, /)
|      Partition the bytes into three parts using the given separator.
|
|      This will search for the separator sep in the bytes. If the separator is found,
|      returns a 3-tuple containing the part before the separator, the separator
|      itself, and the part after it.
|
|      If the separator is not found, returns a 3-tuple containing the original bytes
|      object and two empty bytes objects.
```

replace(self, old, new, count=-1, /),【返回值】: a copy with all occurrences of substring old replaced by new.

count
Maximum number of occurrences to replace.
-1 (the default value) means replace all occurrences.

If the optional argument count is given, only the first count occurrences are
replaced.

rfind(...)
B.rfind(sub[, start[, end]]) -> int
,【返回值】: the highest index in B where substring sub is found,
such that sub is contained within B[start:end]. Optional
arguments start and end are interpreted as in slice notation.
,【返回值】: -1 on failure.

rindex(...)
B.rindex(sub[, start[, end]]) -> int

Like B.rfind() but raise ValueError when the substring is not found.

rjust(...)
B.rjust(width[, fillchar]) -> copy of B
,【返回值】: B right justified in a string of length width. Padding is
done using the specified fill character (default is a space)

rpartition(self, sep, /)
Partition the bytes into three parts using the given separator.

This will search for the separator sep in the bytes, starting and the end. If
the separator is found, returns a 3-tuple containing the part before the
separator, the separator itself, and the part after it.

If the separator is not found, returns a 3-tuple containing two empty bytes
objects and the original bytes object.

rsplit(self, /, sep=None, maxsplit=-1),【返回值】: a list of the sections in the bytes, using sep as the delimiter.

sep
The delimiter according which to split the bytes.
None (the default value) means split on ASCII whitespace characters
(space, tab, return, newline, formfeed, vertical tab).
maxsplit
Maximum number of splits to do.

```
|               -1 (the default value) means no limit.
|
|       Splitting is done starting at the end of the bytes and working to the front.
|
|   rstrip(self, bytes=None, /)
|       Strip trailing bytes contained in the argument.
|
|       If the argument is omitted or None, strip trailing ASCII whitespace.
|
|   split(self, /, sep=None, maxsplit=-1),【返回值】: a list of the sections in the bytes, using sep as the delimiter.
|
|       sep
|         The delimiter according which to split the bytes.
|         None (the default value) means split on ASCII whitespace characters
|         (space, tab, return, newline, formfeed, vertical tab).
|       maxsplit
|         Maximum number of splits to do.
|         -1 (the default value) means no limit.
|
|   splitlines(self, /, keepends=False),【返回值】: a list of the lines in the bytes, breaking at line boundaries.
|
|       Line breaks are not included in the resulting list unless keepends is given and
|       true.
|
|   startswith(...)
|       B.startswith(prefix[, start[, end]]) -> bool
|       ,【返回值】: True if B starts with the specified prefix, False otherwise.
|       With optional start, test B beginning at that position.
|       With optional end, stop comparing B at that position.
|       prefix can also be a tuple of bytes to try.
|
|   strip(self, bytes=None, /)
|       Strip leading and trailing bytes contained in the argument.
|
|       If the argument is omitted or None, strip leading and trailing ASCII whitespace.
|
|   swapcase(...)
|       B.swapcase() -> copy of B
|       ,【返回值】: a copy of B with uppercase ASCII characters converted
|       to lowercase ASCII and vice versa.
|
|   title(...)
|       B.title() -> copy of B
|       ,【返回值】: a titlecased version of B, i.e. ASCII words start with uppercase
|       characters, all remaining cased characters have lowercase.
```

```
|
|  translate(...)
|      translate(table, [deletechars]),【返回值】: a copy with each character mapped by the given translation table.
|
|          table
|              Translation table, which must be a bytes object of length 256.
|
|      All characters occurring in the optional argument deletechars are removed.
|      The remaining characters are mapped through the given translation table.
|
|  upper(...)
|      B.upper() -> copy of B
|      ,【返回值】: a copy of B with all ASCII characters converted to uppercase.
|
|  zfill(...)
|      B.zfill(width) -> copy of B
|
|      Pad a numeric string B with zeros on the left, to fill a field
|      of the specified width.  B is never truncated.
|
|  ----------------------------------------------------------------------
|  Static methods defined here:
|
|  maketrans(frm, to, /),【返回值】: a translation table useable for the bytes or bytearray translate method.
|
|      The returned table will be one where each byte in frm is mapped to the byte at
|      the same position in to.
|
|      The bytes objects frm and to must be of the same length.
```

__version__

未找到文档：'0.4.9'.

abstract 抽象修饰符

module talib.abstract in talib:

【名称】

talib.abstract - This file Copyright (c) 2013 Brian A Cappello <briancappello at gmail>

【类定义】

builtins.object

Function

class Function(builtins.object)

| This is a pythonic wrapper around TALIB's abstract interface. It is
| intended to simplify using individual TALIB functions by providing a
| unified interface for setting/controlling input data, setting function
| parameters and retrieving results. Input data consists of a ``dict`` of
| ``numpy`` arrays (or a ``pandas.DataFrame``), one array for each of open,
| high, low, close and volume. This can be set with the set_input_arrays()
| method. Which keyed array(s) are used as inputs when calling the function
| is controlled using the input_names property.
|
| This class gets initialized with a TALIB function name and optionally an
| input_arrays object. It provides the following primary functions for
| setting inputs and retrieving results:
|
| ---- input_array/TA-function-parameter set-only functions -----
| - set_input_arrays(input_arrays)
| - set_function_args([input_arrays,] [param_args_andor_kwargs])
|
| Documentation for param_args_andor_kwargs can be seen by printing the
| Function instance or programatically via the info, input_names and
| parameters properties.
|
| ----- result-returning functions -----
| - the outputs property wraps a method which ensures results are always valid
| - run([input_arrays]) # calls set_input_arrays and returns self.outputs
| - FunctionInstance([input_arrays,] [param_args_andor_kwargs]) # calls set_function_args and returns self.outputs
|
| 【方法定义】
|
| __call__(self, *args, **kwargs)
| func_instance([input_arrays,] [parameter_args,] [input_price_series_kwargs,] [parameter_kwargs])
|
| This is a shortcut to the outputs property that also allows setting
| the input_arrays dict and function parameters.
|
| __init__(self, function_name, *args, **kwargs)
|
| __repr__(self)

```
|
|   __str__(self)
|
|   __unicode__(self)
|
|   get_input_arrays(self)
|       Returns a copy of the dict of input arrays in use.
|
|   get_input_names(self)
|       Returns the dict of input price series names that specifies which
|       of the ndarrays in input_arrays will be used to calculate the function.
|
|   get_parameters(self)
|       Returns the function's optional parameters and their default values.
|
|   run(self, input_arrays=None)
|       run([input_arrays=None])
|
|       This is a shortcut to the outputs property that also allows setting
|       the input_arrays dict.
|
|   set_function_args(self, *args, **kwargs)
|       optionl args:[input_arrays,] [parameter_args,] [input_price_series_kwargs,] [parameter_kwargs]
|
|   set_input_arrays(self, input_arrays)
|       Sets the dict of input_arrays to use. Returns True/False for
|       subclasses:
|
|       If input_arrays is a dict with the keys open, high, low, close and
|       volume, it is assigned as the input_array to use and this function
|       returns True, returning False otherwise. If you implement your own
|       data type and wish to subclass Function, you should wrap this function
|       with an if-statement:
|
|       class CustomFunction(Function):
|           def __init__(self, function_name):
|               Function.__init__(self, function_name)
|
|           def set_input_arrays(self, input_data):
|               if Function.set_input_arrays(self, input_data):
|                   return True
|               elif isinstance(input_data, some_module.CustomDataType):
|                   input_arrays = Function.get_input_arrays(self)
|                   # convert input_data to input_arrays and then call the super
|                   Function.set_input_arrays(self, input_arrays)
```

```
|                         return True
|                     return False
|
|  set_input_names(self, input_names)
|      Sets the input price series names to use.
|
|  set_parameters(self, parameters)
|      Sets the function parameter values.
|
|  ----------------------------------------------------------------------
|  【数据说明】
|
|  __dict__
|      dictionary for instance variables (if defined)
|
|  __weakref__
|      list of weak references to the object (if defined)
|
|  function_flags
|      Returns any function flags defined for this indicator function.
|
|  info
|      Returns a copy of the function's info dict.
|
|  input_arrays
|      Returns a copy of the dict of input arrays in use.
|
|  input_names
|      Returns the dict of input price series names that specifies which
|      of the ndarrays in input_arrays will be used to calculate the function.
|
|  lookback
|      Returns the lookback window size for the function with the parameter
|      values that are currently set.
|
|  output_flags
|      Returns the flags for each output for this indicator function.
|
|  output_names
|      Returns a list of the output names returned by this function.
|
|  outputs
|      Returns the TA function values for the currently set input_arrays and
|      parameters. Returned values are a ndarray if there is only one output
|      or a list of ndarrays for more than one output.
```

|
| parameters
| Returns the function's optional parameters and their default values.

【数据】

ACOS = {'parameters': OrderedDict(), 'display_name': 'V...'output_name...
AD = {'parameters': OrderedDict(), 'display_name': 'C...'output_names'...
ADD = {'parameters': OrderedDict(), 'display_name': 'V...'output_names...
ADOSC = {'parameters': OrderedDict([('fastperiod', 3), (...'output_nam...
ADX = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flags': ...
ADXR = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flags':...
APO = {'parameters': OrderedDict([('fastperiod', 12), ...'output_names...
AROON = {'parameters': OrderedDict([('timeperiod', 14)])...['aroondown...
AROONOSC = {'parameters': OrderedDict([('timeperiod', 14)])...'output_...
ASIN = {'parameters': OrderedDict(), 'display_name': 'V...'output_name...
ATAN = {'parameters': OrderedDict(), 'display_name': 'V...'output_name...
ATR = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flags': ...
AVGPRICE = {'parameters': OrderedDict(), 'display_name': 'A...'functio...
BBANDS = {'parameters': OrderedDict([('timeperiod', 5), (...'function_...
BETA = {'parameters': OrderedDict([('timeperiod', 5)]),...'output_name...
BOP = {'parameters': OrderedDict(), 'display_name': 'B...'output_names...
CCI = {'parameters': OrderedDict([('timeperiod', 14)])...'output_names...
CDL2CROWS = {'parameters': OrderedDict(), 'display_name': 'T...], 'fun...
CDL3BLACKCROWS = {'parameters': OrderedDict(), 'display_name': 'T...],...
CDL3INSIDE = {'parameters': OrderedDict(), 'display_name': 'T...], 'fu...
CDL3LINESTRIKE = {'parameters': OrderedDict(), 'display_name': 'T...],...
CDL3OUTSIDE = {'parameters': OrderedDict(), 'display_name': 'T...], 'f...
CDL3STARSINSOUTH = {'parameters': OrderedDict(), 'display_name': 'T.....
CDL3WHITESOLDIERS = {'parameters': OrderedDict(), 'display_name': 'T.....
CDLABANDONEDBABY = {'parameters': OrderedDict([('penetration', 0.3).....
CDLADVANCEBLOCK = {'parameters': OrderedDict(), 'display_name': 'A...]...
CDLBELTHOLD = {'parameters': OrderedDict(), 'display_name': 'B...], 'f...
CDLBREAKAWAY = {'parameters': OrderedDict(), 'display_name': 'B...], '...
CDLCLOSINGMARUBOZU = {'parameters': OrderedDict(), 'display_name': 'C....
CDLCONCEALBABYSWALL = {'parameters': OrderedDict(), 'display_name': 'C...
CDLCOUNTERATTACK = {'parameters': OrderedDict(), 'display_name': 'C.....
CDLDARKCLOUDCOVER = {'parameters': OrderedDict([('penetration', 0.5).....
CDLDOJI = {'parameters': OrderedDict(), 'display_name': 'D...], 'funct...
CDLDOJISTAR = {'parameters': OrderedDict(), 'display_name': 'D...], 'f...
CDLDRAGONFLYDOJI = {'parameters': OrderedDict(), 'display_name': 'D.....
CDLENGULFING = {'parameters': OrderedDict(), 'display_name': 'E...], '...
CDLEVENINGDOJISTAR = {'parameters': OrderedDict([('penetration', 0.3)....
CDLEVENINGSTAR = {'parameters': OrderedDict([('penetration', 0.3)...],...
CDLGAPSIDESIDEWHITE = {'parameters': OrderedDict(), 'display_name': 'U...
CDLGRAVESTONEDOJI = {'parameters': OrderedDict(), 'display_name': 'G.....

CDLHAMMER = {'parameters': OrderedDict(), 'display_name': 'H...], 'fun...
CDLHANGINGMAN = {'parameters': OrderedDict(), 'display_name': 'H...], ...
CDLHARAMI = {'parameters': OrderedDict(), 'display_name': 'H...], 'fun...
CDLHARAMICROSS = {'parameters': OrderedDict(), 'display_name': 'H...],...
CDLHIGHWAVE = {'parameters': OrderedDict(), 'display_name': 'H...], 'f...
CDLHIKKAKE = {'parameters': OrderedDict(), 'display_name': 'H...], 'fu...
CDLHIKKAKEMOD = {'parameters': OrderedDict(), 'display_name': 'M...], ...
CDLHOMINGPIGEON = {'parameters': OrderedDict(), 'display_name': 'H...]...
CDLIDENTICAL3CROWS = {'parameters': OrderedDict(), 'display_name': 'I....
CDLINNECK = {'parameters': OrderedDict(), 'display_name': 'I...], 'fun...
CDLINVERTEDHAMMER = {'parameters': OrderedDict(), 'display_name': 'I.....
CDLKICKING = {'parameters': OrderedDict(), 'display_name': 'K...], 'fu...
CDLKICKINGBYLENGTH = {'parameters': OrderedDict(), 'display_name': 'K....
CDLLADDERBOTTOM = {'parameters': OrderedDict(), 'display_name': 'L...]...
CDLLONGLEGGEDDOJI = {'parameters': OrderedDict(), 'display_name': 'L.....
CDLLONGLINE = {'parameters': OrderedDict(), 'display_name': 'L...], 'f...
CDLMARUBOZU = {'parameters': OrderedDict(), 'display_name': 'M...], 'f...
CDLMATCHINGLOW = {'parameters': OrderedDict(), 'display_name': 'M...],...
CDLMATHOLD = {'parameters': OrderedDict([('penetration', 0.5)...], 'fu...
CDLMORNINGDOJISTAR = {'parameters': OrderedDict([('penetration', 0.3)....
CDLMORNINGSTAR = {'parameters': OrderedDict([('penetration', 0.3)...],...
CDLONNECK = {'parameters': OrderedDict(), 'display_name': 'O...], 'fun...
CDLPIERCING = {'parameters': OrderedDict(), 'display_name': 'P...], 'f...
CDLRICKSHAWMAN = {'parameters': OrderedDict(), 'display_name': 'R...],...
CDLRISEFALL3METHODS = {'parameters': OrderedDict(), 'display_name': 'R...
CDLSEPARATINGLINES = {'parameters': OrderedDict(), 'display_name': 'S....
CDLSHOOTINGSTAR = {'parameters': OrderedDict(), 'display_name': 'S...]...
CDLSHORTLINE = {'parameters': OrderedDict(), 'display_name': 'S...], '...
CDLSPINNINGTOP = {'parameters': OrderedDict(), 'display_name': 'S...],...
CDLSTALLEDPATTERN = {'parameters': OrderedDict(), 'display_name': 'S.....
CDLSTICKSANDWICH = {'parameters': OrderedDict(), 'display_name': 'S......
CDLTAKURI = {'parameters': OrderedDict(), 'display_name': 'T...], 'fun...
CDLTASUKIGAP = {'parameters': OrderedDict(), 'display_name': 'T...], '...
CDLTHRUSTING = {'parameters': OrderedDict(), 'display_name': 'T...], '...
CDLTRISTAR = {'parameters': OrderedDict(), 'display_name': 'T...], 'fu...
CDLUNIQUE3RIVER = {'parameters': OrderedDict(), 'display_name': 'U...]...
CDLUPSIDEGAP2CROWS = {'parameters': OrderedDict(), 'display_name': 'U....
CDLXSIDEGAP3METHODS = {'parameters': OrderedDict(), 'display_name': 'U...
CEIL = {'parameters': OrderedDict(), 'display_name': 'V...'output_name...
CMO = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flags': ...
CORREL = {'parameters': OrderedDict([('timeperiod', 30)])...'output_na...
COS = {'parameters': OrderedDict(), 'display_name': 'V...'output_names...
COSH = {'parameters': OrderedDict(), 'display_name': 'V...'output_name...
DEMA = {'parameters': OrderedDict([('timeperiod', 30)])...'function_fl...
DIV = {'parameters': OrderedDict(), 'display_name': 'V...'output_names...

DX = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flags': [...
EMA = {'parameters': OrderedDict([('timeperiod', 30)])...me as input',...
EXP = {'parameters': OrderedDict(), 'display_name': 'V...'output_names...
FLOOR = {'parameters': OrderedDict(), 'display_name': 'V...'output_nam...
HT_DCPERIOD = {'parameters': OrderedDict(), 'display_name': 'H...tion_...
HT_DCPHASE = {'parameters': OrderedDict(), 'display_name': 'H...tion_f...
HT_PHASOR = {'parameters': OrderedDict(), 'display_name': 'H...tion_fl...
HT_SINE = {'parameters': OrderedDict(), 'display_name': 'H...tion_flag...
HT_TRENDLINE = {'parameters': OrderedDict(), 'display_name': 'H...me a...
HT_TRENDMODE = {'parameters': OrderedDict(), 'display_name': 'H...tion...
KAMA = {'parameters': OrderedDict([('timeperiod', 30)])...me as input'...
LINEARREG = {'parameters': OrderedDict([('timeperiod', 14)])...'functi...
LINEARREG_ANGLE = {'parameters': OrderedDict([('timeperiod', 14)])...'...
LINEARREG_INTERCEPT = {'parameters': OrderedDict([('timeperiod', 14)])...
LINEARREG_SLOPE = {'parameters': OrderedDict([('timeperiod', 14)])...'...
LN = {'parameters': OrderedDict(), 'display_name': 'V...'output_names'...
LOG10 = {'parameters': OrderedDict(), 'display_name': 'V...'output_nam...
MA = {'parameters': OrderedDict([('timeperiod', 30), ...'function_flag...
MACD = {'parameters': OrderedDict([('fastperiod', 12), ...macdsignal',...
MACDEXT = {'parameters': OrderedDict([('fastperiod', 12), ...macdsigna...
MACDFIX = {'parameters': OrderedDict([('signalperiod', 9)]...macdsigna...
MAMA = {'parameters': OrderedDict([('fastlimit', 0.5), ...me as input'...
MAVP = {'parameters': OrderedDict([('minperiod', 2), ('...'function_fl...
MAX = {'parameters': OrderedDict([('timeperiod', 30)])...'function_fla...
MAXINDEX = {'parameters': OrderedDict([('timeperiod', 30)])...tput_nam...
MEDPRICE = {'parameters': OrderedDict(), 'display_name': 'M...'functio...
MFI = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flags': ...
MIDPOINT = {'parameters': OrderedDict([('timeperiod', 14)])...'functio...
MIDPRICE = {'parameters': OrderedDict([('timeperiod', 14)])...'functio...
MIN = {'parameters': OrderedDict([('timeperiod', 30)])...'function_fla...
MININDEX = {'parameters': OrderedDict([('timeperiod', 30)])...tput_nam...
MINMAX = {'parameters': OrderedDict([('timeperiod', 30)])...'function_...
MINMAXINDEX = {'parameters': OrderedDict([('timeperiod', 30)])...s': [...
MINUS_DI = {'parameters': OrderedDict([('timeperiod', 14)])...tion_fla...
MINUS_DM = {'parameters': OrderedDict([('timeperiod', 14)])...tion_fla...
MOM = {'parameters': OrderedDict([('timeperiod', 10)])...'output_names...
MULT = {'parameters': OrderedDict(), 'display_name': 'V...'output_name...
NATR = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flags':...
OBV = {'parameters': OrderedDict(), 'display_name': 'O...'output_names...
PLUS_DI = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flag...
PLUS_DM = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flag...
PPO = {'parameters': OrderedDict([('fastperiod', 12), ...'output_names...
ROC = {'parameters': OrderedDict([('timeperiod', 10)])...'output_names...
ROCP = {'parameters': OrderedDict([('timeperiod', 10)])...'output_name...
ROCR = {'parameters': OrderedDict([('timeperiod', 10)])...'output_name...

ROCR100 = {'parameters': OrderedDict([('timeperiod', 10)])...'output_n...
RSI = {'parameters': OrderedDict([('timeperiod', 14)])...tion_flags': ...
SAR = {'parameters': OrderedDict([('acceleration', 0.0...'function_fla...
SAREXT = {'parameters': OrderedDict([('startvalue', 0), (...'function_...
SIN = {'parameters': OrderedDict(), 'display_name': 'V...'output_names...
SINH = {'parameters': OrderedDict(), 'display_name': 'V...'output_name...
SMA = {'parameters': OrderedDict([('timeperiod', 30)])...'function_fla...
SQRT = {'parameters': OrderedDict(), 'display_name': 'V...'output_name...
STDDEV = {'parameters': OrderedDict([('timeperiod', 5), (...'output_na...
STOCH = {'parameters': OrderedDict([('fastk_period', 5),...mes': ['slo...
STOCHF = {'parameters': OrderedDict([('fastk_period', 5),...mes': ['fa...
STOCHRSI = {'parameters': OrderedDict([('timeperiod', 14), ...tion_fla...
SUB = {'parameters': OrderedDict(), 'display_name': 'V...'output_names...
SUM = {'parameters': OrderedDict([('timeperiod', 30)])...'output_names...
T3 = {'parameters': OrderedDict([('timeperiod', 5), (...me as input', ...
TAN = {'parameters': OrderedDict(), 'display_name': 'V...'output_names...
TANH = {'parameters': OrderedDict(), 'display_name': 'V...'output_name...
TEMA = {'parameters': OrderedDict([('timeperiod', 30)])...'function_fl...
TRANGE = {'parameters': OrderedDict(), 'display_name': 'T...'output_na...
TRIMA = {'parameters': OrderedDict([('timeperiod', 30)])...'function_f...
TRIX = {'parameters': OrderedDict([('timeperiod', 30)])...'output_name...
TSF = {'parameters': OrderedDict([('timeperiod', 14)])...'function_fla...
TYPPRICE = {'parameters': OrderedDict(), 'display_name': 'T...'functio...
ULTOSC = {'parameters': OrderedDict([('timeperiod1', 7), ...'output_na...
VAR = {'parameters': OrderedDict([('timeperiod', 5), (...'output_names...
WCLPRICE = {'parameters': OrderedDict(), 'display_name': 'W...'functio...
WILLR = {'parameters': OrderedDict([('timeperiod', 14)])...'output_nam...
WMA = {'parameters': OrderedDict([('timeperiod', 30)])...'function_fla...
__all__ = ['Function', 'BBANDS', 'OBV', 'CDLHIGHWAVE', 'COS', 'MA', 'C...
__test__ = {}

【文件】: \talib\abstract.cp35-win_amd64.pyd

atexit

built-in module atexit:

【名称】
atexit - allow programmer to define multiple exit functions to be executedupon normal program termination.

【说明】
Two public functions, register and unregister, are defined.

【函数】
register(...)
register(func, *args, **kwargs) -> func

Register a function to be executed upon normal program termination

func - function to be called at exit
args - optional arguments to pass to func
kwargs - optional keyword arguments to pass to func

func is returned to facilitate usage as a decorator.

unregister(...)
unregister(func) -> None

Unregister an exit function which was previously registered using
atexit.register

func - function to be unregistered

【文件】: (built-in)

common 通用

module talib.common in talib:

【名称】
talib.common

【数据】
MA_Type = <talib.common.MA_Type object>
__pyx_capi__ = {'_ta_check_success': <capsule object "PyObject *(PyObj...
__ta_version__ = b'0.4.0 (Sep 10 2015 14:20:56)'
__test__ = {}

【文件】: \talib\common.cp35-win_amd64.pyd

get_function_groups

function get_function_groups in module talib:

get_function_groups()
    Returns a dict with keys of function-group names and values of lists
    of function names ie {'group_names': ['function_names']}

get_functions

function get_functions in module talib:

get_functions()
    Returns a list of all the functions supported by TALIB

CDL-k 线图相关函数

CDL2CROWS k 线图--两只乌鸦

CDL2CROWS　位于模块：　talib.func:

CDL2CROWS(...)
    CDL2CROWS(open, high, low, close)

    Two Crows (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDL3BLACKCROWS：k 线图--3 只黑乌鸦

CDL3BLACKCROWS　位于模块：　talib.func:

CDL3BLACKCROWS(...)
    CDL3BLACKCROWS(open, high, low, close)

    Three Black Crows (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDL3INSIDE k 线图：3 内上下震荡

CDL3INSIDE 位于模块： talib.func:

CDL3INSIDE(...)
    CDL3INSIDE(open, high, low, close)

    Three Inside Up/Down (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDL3LINESTRIKE k 线图：3 线震荡

CDL3LINESTRIKE 位于模块： talib.func:

CDL3LINESTRIKE(...)
    CDL3LINESTRIKE(open, high, low, close)

    Three-Line Strike (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDL3OUTSIDE k 线图：3 外下震荡

CDL3OUTSIDE 位于模块： talib.func:

CDL3OUTSIDE(...)
    CDL3OUTSIDE(open, high, low, close)

    Three Outside Up/Down (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDL3STARSINSOUTH k 线图：南方三星

CDL3STARSINSOUTH 位于模块： talib.func:

CDL3STARSINSOUTH(...)
CDL3STARSINSOUTH(open, high, low, close)

Three Stars In The South (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDL3WHITESOLDIERS k 线图：三白兵

CDL3WHITESOLDIERS 位于模块： talib.func:

CDL3WHITESOLDIERS(...)
CDL3WHITESOLDIERS(open, high, low, close)

Three Advancing White Soldiers (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLABANDONEDBABY k 线图：弃婴

CDLABANDONEDBABY 位于模块： talib.func:

CDLABANDONEDBABY(...)
CDLABANDONEDBABY(open, high, low, close[, penetration=?])

Abandoned Baby (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【参数】
penetration: 0.3
【输出】
integer (values are -100, 0 or 100)

CDLADVANCEBLOCK ,K 线图：推进

CDLADVANCEBLOCK 位于模块： talib.func:

CDLADVANCEBLOCK(...)
CDLADVANCEBLOCK(open, high, low, close)

Advance Block (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLBELTHOLD ,K 线图：带住

CDLBELTHOLD 位于模块： talib.func:

CDLBELTHOLD(...)
CDLBELTHOLD(open, high, low, close)

Belt-hold (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLBREAKAWAY,K 线图：分离

CDLBREAKAWAY 位于模块： talib.func:

CDLBREAKAWAY(...)
CDLBREAKAWAY(open, high, low, close)

Breakaway (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLCLOSINGMARUBOZU,K 线图：收盘光头光脚

CDLCLOSINGMARUBOZU 位于模块： talib.func:

CDLCLOSINGMARUBOZU(...)
CDLCLOSINGMARUBOZU(open, high, low, close)

Closing Marubozu (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLCONCEALBABYSWALL K 线图：藏婴吞没形态

CDLCONCEALBABYSWALL 位于模块： talib.func:

CDLCONCEALBABYSWALL(...)
CDLCONCEALBABYSWALL(open, high, low, close)

Concealing Baby Swallow (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLCOUNTERATTACK

,K 线图：反击

CDLCOUNTERATTACK 位于模块： talib.func:

CDLCOUNTERATTACK(...)
CDLCOUNTERATTACK(open, high, low, close)

Counterattack (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLDARKCLOUDCOVER,K 线图：乌云盖

CDLDARKCLOUDCOVER 位于模块： talib.func:

CDLDARKCLOUDCOVER(...)
CDLDARKCLOUDCOVER(open, high, low, close[, penetration=?])

Dark Cloud Cover (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【参数】

penetration: 0.5

【输出】

integer (values are -100, 0 or 100)

CDLDOJI,K 线图：十字星

CDLDOJI　位于模块：　talib.func:

CDLDOJI(...)

CDLDOJI(open, high, low, close)

Doji (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】

integer (values are -100, 0 or 100)

CDLDOJISTAR,K 线图：十字星

CDLDOJISTAR　位于模块：　talib.func:

CDLDOJISTAR(...)

CDLDOJISTAR(open, high, low, close)

Doji Star (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】

integer (values are -100, 0 or 100)

CDLDRAGONFLYDOJI,K 线图：蜻蜓十字星

CDLDRAGONFLYDOJI 位于模块： talib.func:

CDLDRAGONFLYDOJI(...)
    CDLDRAGONFLYDOJI(open, high, low, close)

    Dragonfly Doji (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLENGULFING,K 线图：吞没

CDLENGULFING 位于模块： talib.func:

CDLENGULFING(...)
    CDLENGULFING(open, high, low, close)

    Engulfing Pattern (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLEVENINGDOJISTAR,K 线图：黄昏十字星

CDLEVENINGDOJISTAR 位于模块： talib.func:

CDLEVENINGDOJISTAR(...)
    CDLEVENINGDOJISTAR(open, high, low, close[, penetration=?])

    Evening Doji Star (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【参数】
        penetration: 0.3
【输出】

integer (values are -100, 0 or 100)

CDLEVENINGSTAR,K 线图：黄昏之星

CDLEVENINGSTAR 位于模块： talib.func:

CDLEVENINGSTAR(...)
CDLEVENINGSTAR(open, high, low, close[, penetration=?])

Evening Star (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【参数】
penetration: 0.3
【输出】
integer (values are -100, 0 or 100)

CDLGAPSIDESIDEWHITE,K 线图：上/下间隙并排的白色线条

CDLGAPSIDESIDEWHITE 位于模块： talib.func:

CDLGAPSIDESIDEWHITE(...)
CDLGAPSIDESIDEWHITE(open, high, low, close)

Up/Down-gap side-by-side white lines (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLGRAVESTONEDOJI,K 线图：墓碑十字线

CDLGRAVESTONEDOJI 位于模块： talib.func:

CDLGRAVESTONEDOJI(...)
    CDLGRAVESTONEDOJI(open, high, low, close)

    Gravestone Doji (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLHAMMER,K 线图：锤

CDLHAMMER　位于模块：　talib.func:

CDLHAMMER(...)
    CDLHAMMER(open, high, low, close)

    Hammer (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLHANGINGMAN,K 线图：吊人

CDLHANGINGMAN　位于模块：　talib.func:

CDLHANGINGMAN(...)
    CDLHANGINGMAN(open, high, low, close)

    Hanging Man (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLHARAMI,K 线图：阴阳线

CDLHARAMI　位于模块：　talib.func:

CDLHARAMI(...)
    CDLHARAMI(open, high, low, close)

    Harami Pattern (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLHARAMICROSS,K 线图：交叉阴阳线

CDLHARAMICROSS　位于模块：　talib.func:

CDLHARAMICROSS(...)
    CDLHARAMICROSS(open, high, low, close)

    Harami Cross Pattern (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLHIGHWAVE,K 线图：长脚十字线

CDLHIGHWAVE　位于模块：　talib.func:

CDLHIGHWAVE(...)
    CDLHIGHWAVE(open, high, low, close)

    High-Wave Candle (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLHIKKAKE,K 线图：陷阱

CDLHIKKAKE 位于模块： talib.func:

CDLHIKKAKE(...)
CDLHIKKAKE(open, high, low, close)

Hikkake Pattern (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLHIKKAKEMOD,K 线图：改良的陷阱

CDLHIKKAKEMOD 位于模块： talib.func:

CDLHIKKAKEMOD(...)
CDLHIKKAKEMOD(open, high, low, close)

Modified Hikkake Pattern (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLHOMINGPIGEON,K 线图：信鸽

CDLHOMINGPIGEON 位于模块： talib.func:

CDLHOMINGPIGEON(...)
CDLHOMINGPIGEON(open, high, low, close)

Homing Pigeon (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLIDENTICAL3CROWS,K 线图：相同的三只乌鸦

CDLIDENTICAL3CROWS 位于模块： talib.func:

CDLIDENTICAL3CROWS(...)
CDLIDENTICAL3CROWS(open, high, low, close)

Identical Three Crows (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLINNECK,K 线图：颈纹

CDLINNECK 位于模块： talib.func:

CDLINNECK(...)
CDLINNECK(open, high, low, close)

In-Neck Pattern (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLINVERTEDHAMMER,K 线图：倒锤

CDLINVERTEDHAMMER 位于模块： talib.func:

CDLINVERTEDHAMMER(...)
CDLINVERTEDHAMMER(open, high, low, close)

Inverted Hammer (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLKICKING,K 线图：踢

CDLKICKING 位于模块： talib.func:

CDLKICKING(...)
CDLKICKING(open, high, low, close)

Kicking (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLKICKINGBYLENGTH,K 线图：踢牛/踢熊

由较长的光头光脚的确定

CDLKICKINGBYLENGTH　位于模块：　talib.func:

CDLKICKINGBYLENGTH(...)
    CDLKICKINGBYLENGTH(open, high, low, close)

    Kicking - bull/bear determined by the longer marubozu (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLLADDERBOTTOM,K 线图：梯底

CDLLADDERBOTTOM　位于模块：　talib.func:

CDLLADDERBOTTOM(...)
    CDLLADDERBOTTOM(open, high, low, close)

    Ladder Bottom (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLLONGLEGGEDDOJI,K 线图：长腿十字线

CDLLONGLEGGEDDOJI　位于模块：　talib.func:

CDLLONGLEGGEDDOJI(...)
    CDLLONGLEGGEDDOJI(open, high, low, close)

    Long Legged Doji (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLLONGLINE,K 线图：长线

CDLLONGLINE 位于模块： talib.func:

CDLLONGLINE(...)
CDLLONGLINE(open, high, low, close)

Long Line Candle (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLMARUBOZU,K 线图：光头光脚

CDLMARUBOZU 位于模块： talib.func:

CDLMARUBOZU(...)
CDLMARUBOZU(open, high, low, close)

Marubozu (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLMATCHINGLOW,K 线图：匹配低

CDLMATCHINGLOW 位于模块： talib.func:

CDLMATCHINGLOW(...)
CDLMATCHINGLOW(open, high, low, close)

Matching Low (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLMATHOLD,K 线图：垫住

CDLMATHOLD 位于模块： talib.func:

CDLMATHOLD(...)
CDLMATHOLD(open, high, low, close[, penetration=?])

Mat Hold (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【参数】
penetration: 0.5
【输出】
integer (values are -100, 0 or 100)

CDLMORNINGDOJISTAR,K 线图：早晨十字星

CDLMORNINGDOJISTAR 位于模块： talib.func:

CDLMORNINGDOJISTAR(...)
CDLMORNINGDOJISTAR(open, high, low, close[, penetration=?])

Morning Doji Star (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【参数】

penetration: 0.3

【输出】

integer (values are -100, 0 or 100)

CDLMORNINGSTAR,K 线图：晨星

CDLMORNINGSTAR 位于模块： talib.func:

CDLMORNINGSTAR(...)

CDLMORNINGSTAR(open, high, low, close[, penetration=?])

Morning Star (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【参数】

penetration: 0.3

【输出】

integer (values are -100, 0 or 100)

CDLONNECK,K 线图：颈型

CDLONNECK 位于模块： talib.func:

CDLONNECK(...)

CDLONNECK(open, high, low, close)

On-Neck Pattern (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】

integer (values are -100, 0 or 100)

CDLPIERCING,K 线图：穿孔模式

CDLPIERCING 位于模块： talib.func:

CDLPIERCING(...)
    CDLPIERCING(open, high, low, close)

    Piercing Pattern (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLRICKSHAWMAN,K 线图：车夫

CDLRICKSHAWMAN　位于模块：　talib.func:

CDLRICKSHAWMAN(...)
    CDLRICKSHAWMAN(open, high, low, close)

    Rickshaw Man (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLRISEFALL3METHODS,K 线图：上升/下降三法

CDLRISEFALL3METHODS　位于模块：　talib.func:

CDLRISEFALL3METHODS(...)
    CDLRISEFALL3METHODS(open, high, low, close)

    Rising/Falling Three Methods (Pattern Recognition)

【输入】
        prices: ['open', 'high', 'low', 'close']
【输出】
        integer (values are -100, 0 or 100)

CDLSEPARATINGLINES,K 线图：分割线

CDLSEPARATINGLINES　位于模块：　talib.func:

CDLSEPARATINGLINES(...)
　　CDLSEPARATINGLINES(open, high, low, close)

　　Separating Lines (Pattern Recognition)

【输入】
　　prices: ['open', 'high', 'low', 'close']
【输出】
　　integer (values are -100, 0 or 100)

CDLSHOOTINGSTAR,K 线图：流星

CDLSHOOTINGSTAR　位于模块：　talib.func:

CDLSHOOTINGSTAR(...)
　　CDLSHOOTINGSTAR(open, high, low, close)

　　Shooting Star (Pattern Recognition)

【输入】
　　prices: ['open', 'high', 'low', 'close']
【输出】
　　integer (values are -100, 0 or 100)

CDLSHORTLINE,K 线图：短线

CDLSHORTLINE　位于模块：　talib.func:

CDLSHORTLINE(...)
　　CDLSHORTLINE(open, high, low, close)

　　Short Line Candle (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】

integer (values are -100, 0 or 100)

CDLSPINNINGTOP,K 线图：陀螺

CDLSPINNINGTOP 位于模块： talib.func:

CDLSPINNINGTOP(...)

CDLSPINNINGTOP(open, high, low, close)

Spinning Top (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】

integer (values are -100, 0 or 100)

CDLSTALLEDPATTERN,K 线图：停滞模式

CDLSTALLEDPATTERN 位于模块： talib.func:

CDLSTALLEDPATTERN(...)

CDLSTALLEDPATTERN(open, high, low, close)

Stalled Pattern (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】

integer (values are -100, 0 or 100)

CDLSTICKSANDWICH,K 线图：棍子三明治

CDLSTICKSANDWICH 位于模块： talib.func:

CDLSTICKSANDWICH(...)

CDLSTICKSANDWICH(open, high, low, close)

Stick Sandwich (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLTAKURI,K 线图：托里

（蜻蜓十字星,具有很长的下影线）

CDLTAKURI 位于模块： talib.func:

CDLTAKURI(...)
CDLTAKURI(open, high, low, close)

Takuri (Dragonfly Doji with very long lower shadow) (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLTASUKIGAP,K 线图：翼隙

CDLTASUKIGAP 位于模块： talib.func:

CDLTASUKIGAP(...)
CDLTASUKIGAP(open, high, low, close)

Tasuki Gap (Pattern Recognition)

【输入】
prices: ['open', 'high', 'low', 'close']
【输出】
integer (values are -100, 0 or 100)

CDLTHRUSTING,K 线图：推模式

CDLTHRUSTING　位于模块：　talib.func:

CDLTHRUSTING(...)
　　CDLTHRUSTING(open, high, low, close)

　　Thrusting Pattern (Pattern Recognition)

【输入】
　　prices: ['open', 'high', 'low', 'close']
【输出】
　　integer (values are -100, 0 or 100)

CDLTRISTAR,K 线图：三星模式

CDLTRISTAR　位于模块：　talib.func:

CDLTRISTAR(...)
　　CDLTRISTAR(open, high, low, close)

　　Tristar Pattern (Pattern Recognition)

【输入】
　　prices: ['open', 'high', 'low', 'close']
【输出】
　　integer (values are -100, 0 or 100)

CDLUNIQUE3RIVER,K 线图：独特的 3 河

CDLUNIQUE3RIVER　位于模块：　talib.func:

CDLUNIQUE3RIVER(...)
　　CDLUNIQUE3RIVER(open, high, low, close)

　　Unique 3 River (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】

integer (values are -100, 0 or 100)

CDLUPSIDEGAP2CROWS,K 线图：双飞乌鸦

CDLUPSIDEGAP2CROWS 位于模块： talib.func:

CDLUPSIDEGAP2CROWS(...)

CDLUPSIDEGAP2CROWS(open, high, low, close)

Upside Gap Two Crows (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】

integer (values are -100, 0 or 100)

CDLXSIDEGAP3METHODS,K 线图：上行/下行缺口三方法

CDLXSIDEGAP3METHODS 位于模块： talib.func:

CDLXSIDEGAP3METHODS(...)

CDLXSIDEGAP3METHODS(open, high, low, close)

Upside/Downside Gap Three Methods (Pattern Recognition)

【输入】

prices: ['open', 'high', 'low', 'close']

【输出】

integer (values are -100, 0 or 100)