```mq4
//In-Yo-Ashi.mq4

#property indicator_chart_window

#property indicator_buffers 6

#property indicator_color1 Aqua
#property indicator_color2 Magenta
#property indicator_color3 DodgerBlue
#property indicator_color4 Red
#property indicator_color5 SpringGreen
#property indicator_color6 Gold

#property indicator_width1 5
#property indicator_width2 5
#property indicator_width3 5
#property indicator_width4 5
#property indicator_width5 5
#property indicator_width6 5
```
← #property命令を記述

```mq4
//インジケーターバッファーの宣言
double Open_S[];
double Close_S[];
double Open_M[];
double Close_M[];
double Open_L[];
double Close_L[];
```
← インジケーターバッファーを宣言

```mq4
//変数の宣言
extern int Average_Period_S = 3;
extern int Average_Period_M = 25;
extern int Average_Period_L = 75;
```
← 変数を宣言

```mq4
int init()
{
//インジケーターバッファーのインデックス
SetIndexBuffer(0,Open_S);
SetIndexBuffer(1,Close_S);
SetIndexBuffer(2,Open_M);
SetIndexBuffer(3,Close_M);
SetIndexBuffer(4,Open_L);
SetIndexBuffer(5,Close_L);

//インジケーターのラベル
SetIndexLabel(0,"Open("+Average_Period_S+")");
SetIndexLabel(1,"Close("+Average_Period_S+")");
SetIndexLabel(2,"Open("+Average_Period_M+")");
SetIndexLabel(3,"Close("+Average_Period_M+")");
SetIndexLabel(4,"Open("+Average_Period_L+")");
SetIndexLabel(5,"Close("+Average_Period_L+")");

//インジケーターのスタイル
SetIndexStyle(0,DRAW_HISTOGRAM,STYLE_SOLID);
SetIndexStyle(1,DRAW_HISTOGRAM,STYLE_SOLID);
SetIndexStyle(2,DRAW_HISTOGRAM,STYLE_SOLID);
SetIndexStyle(3,DRAW_HISTOGRAM,STYLE_SOLID);
SetIndexStyle(4,DRAW_HISTOGRAM,STYLE_SOLID);
SetIndexStyle(5,DRAW_HISTOGRAM,STYLE_SOLID);

//インジケーターの描画開始時点
SetIndexDrawBegin(0,Average_Period_S + 1);
SetIndexDrawBegin(1,Average_Period_S + 1);
SetIndexDrawBegin(2,Average_Period_M + 1);
SetIndexDrawBegin(3,Average_Period_M + 1);
SetIndexDrawBegin(4,Average_Period_L + 1);
SetIndexDrawBegin(5,Average_Period_L + 1);

return(0);
}
```
← 基本設定を記述

```mq4
int start()
{
//短期陰陽足の計算
//短期陰陽足の終値の計算
int limit = Bars - IndicatorCounted();

for(int i = 0; i < limit; i++)
  {
    double Close_Sum = 0;

    for(int j = i; j < i + Average_Period_S; j++)
      {
        Close_Sum += Close[j];
      }

    Close_S[i] = Close_Sum / Average_Period_S;
    Close_S[i] = NormalizeDouble(Close_S[i],MarketInfo(Symbol(),MODE_DIGITS));
  }

//短期陰陽足の始値の計算
for(i = 0; i < limit; i++)
  {
    double Open_Sum = 0;

    for(j = i; j < i + Average_Period_S; j++)
      {
        Open_Sum += Close[j+1];
      }

    Open_S[i] = Open_Sum / Average_Period_S;
    Open_S[i] = NormalizeDouble(Open_S[i],MarketInfo(Symbol(),MODE_DIGITS));
  }

//中期陰陽足の計算
//中期陰陽足の終値の計算
for(i = 0; i < limit; i++)
  {
    Close_Sum = 0;

    for(j = i; j < i + Average_Period_M; j++)
      {
        Close_Sum += Close[j];
      }

    Close_M[i] = Close_Sum / Average_Period_M;
    Close_M[i] = NormalizeDouble(Close_M[i],MarketInfo(Symbol(),MODE_DIGITS));
  }
//中期陰陽足の始値の計算
for(i = 0; i < limit; i++)
  {
    Open_Sum = 0;

    for(j = i; j < i + Average_Period_M; j++)
      {
        Open_Sum += Close[j+1];
      }

    Open_M[i] = Open_Sum / Average_Period_M;
    Open_M[i] = NormalizeDouble(Open_M[i],MarketInfo(Symbol(),MODE_DIGITS));
  }

//長期陰陽足の計算
//長期陰陽足の終値の計算
for(i = 0; i < limit; i++)
  {
    Close_Sum = 0;

    for(j = i; j < i + Average_Period_L; j++)
      {
        Close_Sum += Close[j];
      }

    Close_L[i] = Close_Sum / Average_Period_L;
    Close_L[i] = NormalizeDouble(Close_L[i],MarketInfo(Symbol(),MODE_DIGITS));
  }
//長期陰陽足の始値の計算
for(i = 0; i < limit; i++)
  {
    Open_Sum = 0;

    for(j = i; j < i + Average_Period_L; j++)
      {
        Open_Sum += Close[j+1];
      }

    Open_L[i] = Open_Sum / Average_Period_L;
    Open_L[i] = NormalizeDouble(Open_L[i],MarketInfo(Symbol(),MODE_DIGITS));
  }

return(0);
}
```
← 具体的な処理内容を記述