

# Market Structure - Pseudocode

🕒 Date Created	@October 26, 2022 2:56 PM
📌 Status	Feito
# Questões	

## ▼ Rules for Market Structure

### ▼ Basic function definitions

▼ **candlesNumber**: number of candles elapsed since the previous range high/low

```
int candlesNumber = index(currentCandle) - index(previousHigh)  
// previousLow is used in case of a bearish BOS
```

▼ **pivotGaps**: number of candle gaps before a break of high/low

```
for (each candle between currentCandle and previousHigh/previousLow) {  
    int pivotGaps = sum(currentCandleHigh < previousHigh ? 1 ; 0)  
    // currentCandleLow > previousLow in case of a bearish BOS  
}
```

▼ **diffColorBreaks (bool)**: different candle color required in case of a BOS

```
for (each candle between currentCandle and previousHigh/previousLow) {  
    if (bullishBOS) {  
        previousHighCandle or any other candle before the currentCandle (not  
        included) is a bearish candle? TRUE; FALSE;  
    } else if (bearishBOS) {  
        previousLowCandle or any other candle before the currentCandle  
        (current not included) is a bullish candle? TRUE; FALSE;  
    }  
}
```

▼ **Break of Structure (BoS):** crossover of a currentCandleHigh and previousHigh OR crossover of a currentCandleLow and previousLow

```
// each timeframe has its own rules of pivotGaps minimum number AND  
diffColorBreaks TRUE or FALSE;
```

```
// mark with a straight line from previousHigh/previousLow until  
currentCandle
```

▼ **Range Limits:** definition of where is to be positioned previousHigh or previousLow

```
// in case of a bullishBOS, previousLow is not the lowest price between  
index(previousHighCandle) and index(currentCandle)
```

```
// but the last (from right to left) valid internal BOS
```

```
int currIndex = index(currentCandle);
```

```
if (bullishBOS) {
```

```
    for (i; i++; i < candlesNumber) {
```

```
        if (high[currIndex - i] < high[currIndex - (1 + i)]) AND (i ≥ pivotGaps)  
        AND diffColorBreaks {
```

```
            previousLow (which will be the new Range Low, for future  
            calculations) = lowest( from low[currIndex - (1+i)] and  
            low[currIndex] );
```

```
            break;
```

```
        }
```

```
    }
```

```
} else if (bearishBOS) {
```

```
    for (i; i++; i < candlesNumber) {
```

```
        if (low[currIndex - i] > low[currIndex - (1 + i)]) AND (i ≥ pivotGaps) AND  
        diffColorBreaks {
```

```
            previousHigh (which will be the new Range High, for future  
            calculations) = highest( from high[currIndex - (1+i)] and  
            high[currIndex] );
```

```

        break;
    }
}
}

```

- ▼ **noiseCandle:** breaks previousHigh AND previousLow by the same candle  
 wait for candle closure, don't mark as a BOS yet  
 if the next candle continues the previous trend, previousHigh/Low becomes the noiseCandle high/low  
 if the next candle is opposed to the previous trend, there has to be valid BOS to both sides (following above BOS rules)

- ▼ **OrderBlock type 1:** engulfed candles

this is a nice example from mql5 community itself:

<https://www.mql5.com/en/articles/1946>

but I don't only use a single candle that was engulfed, but all of the candles that got engulfed by the current candle body (see strategy images for examples, or ask me, I have tons)