

Горячее

Лучшее

Свежее

Подписки



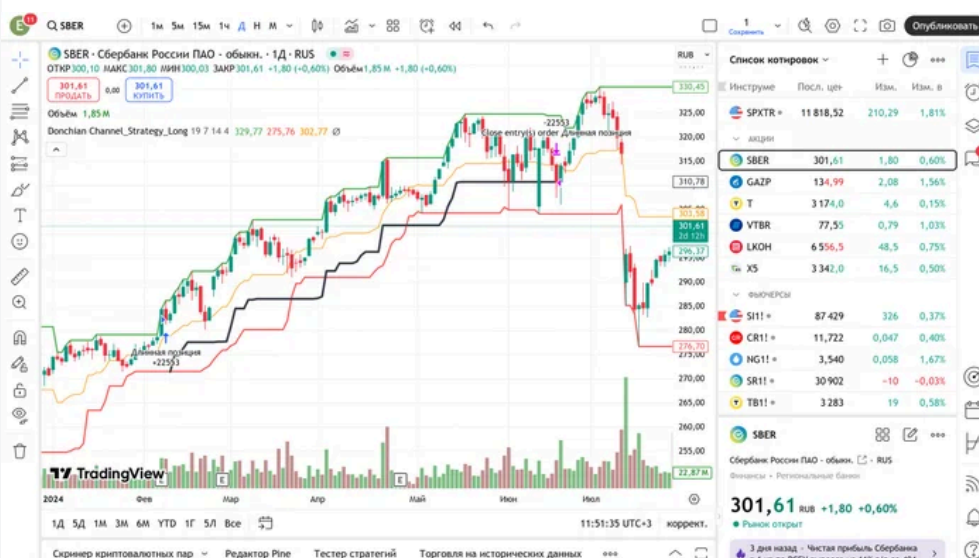
Войти

empenoso 1 месяц назад Лига Инвесторов

## Что за Pine Script и как он помогает рисовать свои штуки на графиках TradingView?

Pine Script - это язык программирования, разработанный командой TradingView как Domain Specific Language, то есть специализированный язык для решения конкретной задачи - анализа и визуализации финансовых данных. Он создан для тех, кто хочет строить собственные индикаторы, тестировать торговые стратегии и делать всё это прямо в интерфейсе графика - без установки Python, без импорта исторических котировок и без настройки среды разработки.

Pine Script предельно прост по синтаксису, но в то же время достаточно мощный, чтобы покрыть 95% потребностей розничного трейдера. В нём предусмотрены ключевые блоки: работа с таймсериями, доступ к фундаментальным данным, рисование на графике и даже поддержка таблиц.



### Базовый синтаксис: как читать и писать код

Pine Script создан с акцентом на простоту: даже если вы раньше не писали код на нём, освоить базовые конструкции можно за вечер. У каждого скрипта есть чёткая структура, и разобраться в ней - первый шаг к созданию собственного инструмента на TradingView.

Начинается скрипт с указания версии языка:

```
//@version=6
```

### Войти

Логин

Пароль

Войти

Создать аккаунт

Забыли пароль?

или продолжите с

Войти с Яндекс ID

Войти через VK ID

Промокоды

Работа

Курсы

Реклама

Игры

Пополнение Steam

Это строка обязательна: она сообщает интерпретатору TradingView, какие правила синтаксиса использовать. Шестая версия — самая новая.

Далее идёт определение типа скрипта:

- `indicator()` — если вы хотите просто визуализировать данные (например, нарисовать скользящую среднюю, уровни, стрелки).
- `strategy()` — если ваша цель — бэктест торговых решений с расчётом сделок, прибыли и убытков.

Чтобы скрипт был гибким, в него добавляют параметры через функцию `input()`. Например:

```
lengthMA = input.int(20, title="Период скользящих средних")
```

Теперь пользователь может сам менять значение прямо из панели настроек на графике.



Рассмотрим базовый пример — аналог "Hello, World" в Pine Script:

```
// This Pine Script® code is subject to the terms of the Mozilla Public License 2.0 at
// https://mozilla.org/MPL/2.0/
// Михаил Шардин, https://shardin.name/?utm_source=tradingview
// 12.04.2025

//@version=6
indicator("Пример скользящих средних", overlay=true)

// Входные параметры
lengthMA = input.int(20, title="Период скользящих средних")
src = input(close, title="Источник данных")

// 1. Простая скользящая средняя (SMA)
```



#### Топ прошлой недели

- Oskanov  
10 постов
- AlexKud  
40 постов
- MeniazovutMariia  
14 постов

[Посмотреть весь топ](#)

РЕКЛАМА · Я.БИЗНЕС



i2hard.shop

**Готовые игровые компьютеры для AAA-игр и киберспорта**

[Узнать больше](#)

```

smaValue = ta.sma(src, lengthMA)
plot(smaValue, "SMA", color=color.blue, linewidth=2)

// 2. Экспоненциальная скользящая средняя (EMA)
emaValue = ta.ema(src, lengthMA)
plot(emaValue, "EMA", color=color.red, linewidth=2)

// 3. Взвешенная скользящая средняя (WMA)
wmaValue = ta.wma(src, lengthMA)
plot(wmaValue, "WMA", color=color.green, linewidth=2)

// 4. Сглаженная скользящая средняя (SMMA)
smmaValue = ta.rma(src, lengthMA)
plot(smmaValue, "SMMA", color=color.purple, linewidth=2)

// Отображение цены для сравнения
plot(close, "Цена", color=color.black, linewidth=1)

// Легенда для отображения текущих значений
var table legendTable = table.new(position.top_right, 1, 6)
if barstate.islast
    table.cell(legendTable, 0, 0, "Тип MA", bgcolor=color.gray)
    table.cell(legendTable, 0, 1, "SMA: " + str.toString(smaValue,
    format.mintick), bgcolor=color.blue)
    table.cell(legendTable, 0, 2, "EMA: " + str.toString(emaValue,
    format.mintick), bgcolor=color.red)
    table.cell(legendTable, 0, 3, "WMA: " + str.toString(wmaValue,
    format.mintick), bgcolor=color.green)
    table.cell(legendTable, 0, 4, "SMMA: " + str.toString(smmaValue,
    format.mintick), bgcolor=color.purple)

```

Этот пример Pine Script демонстрирует сразу четыре типа скользящих средних (MA) и сравнивает их между собой на графике. Скрипт устанавливает параметры:

- lengthMA: пользователь может выбрать период скользящих средних (по умолчанию 20).
- src: выбор источника данных (по умолчанию цена закрытия close).

Рассчитывает и отображает четыре типа MA

- SMA (Simple Moving Average) - простая скользящая, синяя линия.
- EMA (Exponential Moving Average) - экспоненциальная, красная линия.
- WMA (Weighted Moving Average) - взвешенная, зелёная линия.
- SMMA (Smoothed Moving Average) - сглаженная, фиолетовая линия.

Все линии накладываются поверх графика цены благодаря overlay=true.

## Практические примеры

### Пример 1: Индикатор для внутридневной торговли

#### Лучшие посты недели

Рассылка Пикабу:  
отправляем самые  
рейтинговые материалы за 7  
дней 🔥

Укажите

Подписаться

Нажимая кнопку  
«Подписаться на рассылку»,  
я соглашаюсь с [Правилами](#)  
[Пикабу](#) и даю согласие на  
обработку персональных  
данных.

ФИН УСЛУГИ

Реклама. Рекламодатель: ПАО Московская Биржа ИНН 7702077840 ООО «МБ Мэрикет» (ИНН 7727618949)

promo.finuslugi.ru РЕКЛАМА

**Вклад под 30%**  
по ставке по вкладу 30  
процентов на «Финуслугах»

Перейти на сайт

Новости Пикабу Команда Пикабу  
Помощь Награды  
Кодекс Пикабу Контакты  
Реклама О проекте  
О компании Зал славы

Промокоды Купоны Мегамаркет  
Скидки Купоны AliExpress  
Работа Купоны М.Видео  
Курсы Купоны YandexTravel  
Блоги Купоны Lamoda

Мобильное приложение



РЕКЛАМА



Этот мой скрипт под названием *Morning Rage Indicator* предназначен для анализа первых часов торговой сессии. В нём задаётся утренний ценовой диапазон на 15-минутном таймфрейме и отображают уровни High и Low в заданный период — по умолчанию с 8:45 до 10:45. Эти уровни часто служат ориентирами для пробойных стратегий.

```
// Михаил Шардин, https://shardin.name/?utm\_source=tradingview
// 21.03.2025

//@version=6
indicator("Morning Rage Indicator", overlay=true)

// Параметры временных диапазонов
rangeStartHour = input.int(8, "Начало диапазона - час")
rangeStartMinute = input.int(45, "Начало диапазона - минута")
rangeEndHour = input.int(10, "Конец диапазона - час")
rangeEndMinute = input.int(45, "Конец диапазона - минута")

// Источники данных для расчета High/Low диапазона
rangeHighSource = input.source(high, "Источник для Range High")
rangeLowSource = input.source(low, "Источник для Range Low")

// Проверка нового дня
isNewDay = year != year[1] or month != month[1] or dayofmonth != dayofmonth[1]

// Функция для проверки времени
isTime(h, m) =>
hour == h and minute == m

// Определение периода формирования диапазона
```

K2 CLOUD

0+

## Бесплатный перенос 1С в K2 Cloud

Готовое облако для размещения систем 1С от 50+ пользователей



```

isRangePeriod = (hour > rangeStartHour or (hour == rangeStartHour and minute >=
rangeStartMinute)) and (hour < rangeEndHour or (hour == rangeEndHour and
minute <= rangeEndMinute))

// Расчет максимума и минимума диапазона
var float rangeHigh = 0.0
var float rangeLow = 10e10
var bool rangeCalculated = false

// Сброс диапазона в начале нового дня
if isNewDay
rangeHigh := rangeHighSource
rangeLow := rangeLowSource
rangeCalculated := false

// Обновление диапазона только в указанный период
if isRangePeriod and not rangeCalculated
rangeHigh := math.max(rangeHigh, rangeHighSource)
rangeLow := math.min(rangeLow, rangeLowSource)

// Помечаем диапазон как рассчитанный после окончания периода
if hour == rangeEndHour and minute == rangeEndMinute
rangeCalculated := true

// Получаем цену закрытия в момент окончания диапазона
var float rangeClose = na
if hour == rangeEndHour and minute == rangeEndMinute
rangeClose := close

// Визуализация на графике
bgcolor(isRangePeriod and not rangeCalculated ? color.new(color.blue, 90) : na)

// Рисуем уровни диапазона
plot(rangeCalculated ? rangeHigh : na, "Range High", color.green,
2, plot.style_circles)
plot(rangeCalculated ? rangeLow : na, "Range Low", color.red, 2, plot.style_circles)

// Выводим метки с информацией
var table infoTable = table.new(position.top_right, 1, 1)
if barstate.islast
table.cell(infoTable, 0, 0, "Range High: " + str.tostring(rangeHigh) +
"\nRange Low: " + str.tostring(rangeLow),
bgcolor=color.new(color.gray, 80))

```

#### Ключевые функции:

- `isTime()` и `isRangePeriod` определяют нужный временной интервал.
- `rangeHigh` и `rangeLow` обновляются только в рамках этого окна.
- `plot()` визуализирует границы диапазона.

- table показывает итоговые значения прямо на графике.

Вы можете адаптировать код под любые рынки и таймфреймы, изменив время.

## Пример 2: Фундаментальный анализ

Pine Script позволяет не только анализировать технические индикаторы, но и загружать фундаментальные данные компаний прямо на график. Это открывает возможности для оценки бизнеса без переключения между платформами и сайтами. Один из таких примеров — анализ показателя **Net Profit Margin** (чистой рентабельности).

### Что это за показатель?

Net Profit Margin (NPM) показывает, какой процент от выручки остаётся у компании после всех расходов — налогов, процентов, административных издержек и прочего. Если NPM выше 10%, это сигнализирует о финансовом здоровье бизнеса. Выше 15% — компания не просто эффективна, а конкурентоспособна даже в кризисные периоды.

### Фундаментальный анализ акций в РФ и США

#### Разбор кода:

```
//@version=6
indicator("Net Profit Margin: чистая рентабельность", overlay=false)
// Получение квартального значения чистой рентабельности
NPM_quarterly = request.financial(syminfo.tickerid, 'NET_MARGIN', 'FQ')
```

Функция `request.financial()` позволяет подгрузить фундаментальные показатели. В данном случае мы берём NET\_MARGIN за последний **финансовый квартал** ('FQ').

```
// Цвет фона: зелёный — хорошая рентабельность, красный — слабая
bg_color = NPM_quarterly >= 10 ? color.new(color.green, 90) :
color.new(color.red, 90)
bgcolor(bg_color, title="Фон рентабельности")
```

Здесь используется наглядная визуализация: фон графика окрашивается в зависимости от уровня рентабельности. Если маржа  $\geq 10\%$  — всё в порядке, если меньше — стоит быть осторожным.

```
// График и контрольные линии
plot(NPM_quarterly, title="Net Profit Margin", color=color.yellow, linewidth=2)
hline(10, "10% уровень", color=color.silver, linestyle=hline.style_dotted)
hline(15, "15% уровень", color=color.silver, linestyle=hline.style_dotted)
```

На графике рисуется линия с текущими значениями чистой маржи и добавлены два горизонтальных уровня — для отслеживания контрольных границ эффективности.

Коды полностью представлены на Гитхаб.

### Зачем использовать это в реальной торговле?

Представьте, что вы выбираете между двумя акциями. Обе растут на графике, но только у одной стабильная чистая рентабельность выше 15%. Это даёт дополнительное подтверждение: бизнес не просто на подъёме, а действительно зарабатывает. Сильная маржа — защитный буфер в трудные времена.

В Pine Script такие оценки можно строить прямо на графике, комбинируя с техническим анализом. И это делает язык особенно ценным — он объединяет технику и фундаментал в одной среде, [прямо как в книге Виктора Сперандео](#).

### Где брать готовые скрипты?

Если не хочется писать с нуля, начните с изучения чужих наработок. На TradingView есть каталоги авторов, где можно найти десятки полезных скриптов. Мне нравятся два источника:

- [BigBeluga](#) - популярные стратегии, часто с понятным описанием и открытым кодом.
- [PineCodersTASC](#) - официальные скрипты по материалам Traders' Tips, публикуются каждый месяц.

Чтобы использовать скрипт, просто откройте его, нажмите «Добавить на график», а затем — «Скопировать и изменить», если хотите доработать под себя.

Если вам нужен совет по адаптации или написанию скрипта с нуля - напишите мне. Я не так давно начал изучать Pine Script, но уже получил позитивный опыт быстрого прототипирования стратегий под конкретные задачи.

### Заключение



Pine Script – идеальный инструмент для быстрых экспериментов и визуализации торговых идей прямо на графиках TradingView. Он позволяет за 15 минут проверять гипотезы, недоступные стандартным индикаторам.

Не бойтесь начать! Попробуйте изменить примеры из статьи в Pine Editor и добавьте их на график – результат увидите сразу. Удачи!

**Автор:** Михаил Шардин


 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

15 апреля 2025 г.



 0  3.6K 

 [Поддержать](#)

[Эмоции](#)

Больше постов читайте по тегу [«Программирование»](#). А если хотите изучить новую профессию, посмотрите актуальные курсы от проверенных школ с реальными отзывами на сайте [Пикабу Курсы](#).



**Лига Инвесторов**

9.8K постов • 7.7K подписчик

[Добавить пост](#)

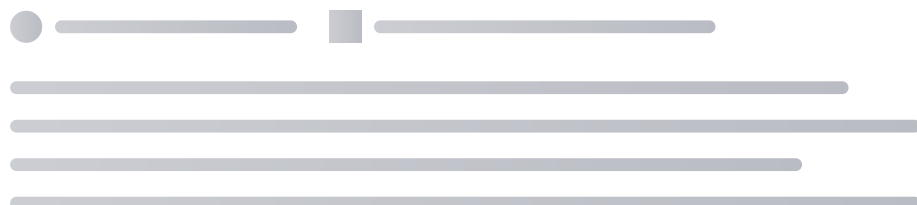
[Подписаться](#)



#### Правила сообщества

1. Необходимо соблюдать правила Пикабу
2. Запрещены посты, не относящиеся к тематике сообщества
3. Запрещается откровенная реклама
4. Нельзя оскорблять участников сообщества.

Чтобы оставить комментарий, необходимо [зарегистрироваться](#) или [войти](#)





\_\_\_\_\_

\_\_\_\_\_

● \_\_\_\_\_ ■ \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

● \_\_\_\_\_ ■ \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

