



# Алгоритмический трейдинг с Python

Денис Колодин  
ген. директор AlgoMarkets LLC



## «Скрытые» проблемы разработки торговых роботов

- **Скорость разработки**

Чем ниже уровень абстракции языка программирования, тем дольше длится разработка торгового робота, так как разработчик должен много внимания уделять деталям.

- **Стоимость доработки**

Торговый робот это программа с высокой долей отладочных изменений, так как его надёжность очень важна. Внесение изменения бывает дороже новой разработки.

- **Юридическое бремя**

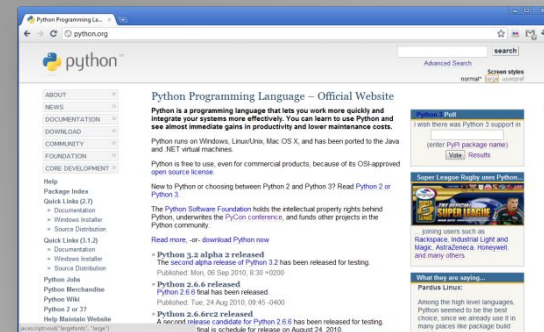
Пользователь торгового робота должен иметь, по большей части, то же программное окружение, что и разработчик. ПО не всегда предоставляется бесплатно.

# Интерпретатор Python


**Python** ([ˈpaɪθən]; пайтон, питон) — высокоуровневый язык программирования общего назначения с акцентом на производительность разработчика и читаемость кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Поддерживает множество парадигм программирования:

- Структурное
- Объектно-ориентированное
- Функциональное
- Императивное
- Аспектно-ориентированное



<http://www.python.org>



# Принцип «батарейки в комплекте»

Python имеет множество многофункциональных библиотек для решения любых задач, от распараллеливания вычислений до формирования трёхмерной графики.

**NumPy** – библиотека высокоуровневых математических функций

**SciPy** – научные вычисления (интегрирование, оптимизация, и т. п.)

**Matplotlib** – библиотека визуализации данных

**PyWin** – интеграции с программами для Windows

**wxPython** – графический интерфейс пользователя

**MarketWizard** – программирование торговых алгоритмов

и ещё более 11000 библиотек!

Всё библиотеки легко подгружаются из **Python Package Index**

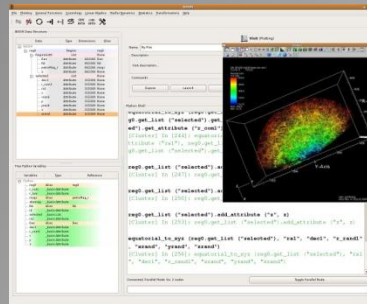
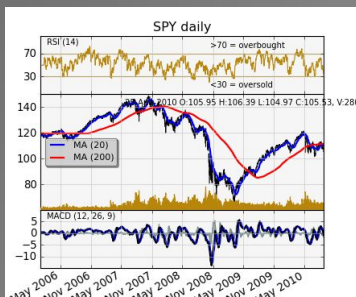
# Кто использует Python?

## Winton Capital Management

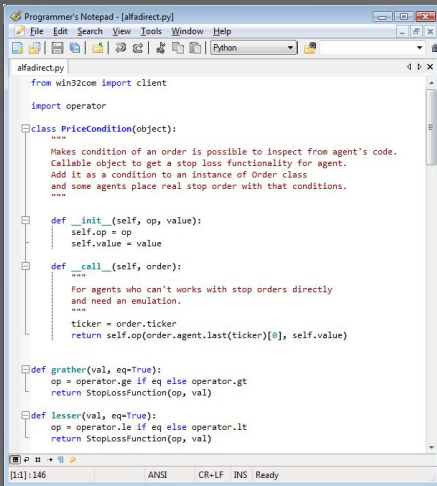
в управлении 14 миллиардов долларов!

## D. E. Shaw & Co

в управлении 21 миллиард долларов!



# Этапы создания программы на Python



```
Programmer's Notepad - [alfadirect.py]
File Edit Search View Tools Window Help
Python
alfadirect.py
from win32com import client
import operator

class PriceCondition(object):
    """
    Makes condition of an order is possible to inspect from agent's code.
    Callable object to get a stop loss functionality for agent.
    Add it as a condition to an instance of Order class
    and some agents place real stop order with that conditions.
    """
    def __init__(self, op, value):
        self.op = op
        self.value = value
    def __call__(self, order):
        """
        For agents who can't work with stop orders directly
        and need an emulation.
        """
        ticker = order.ticker
        return self.op(order.agent.last(ticker)[0], self.value)

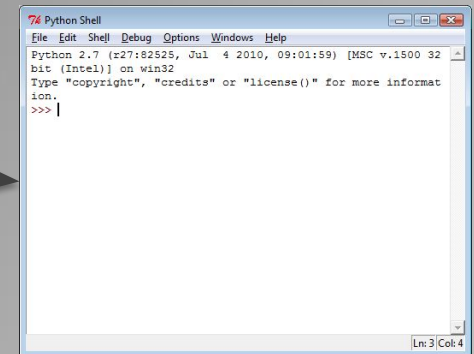
def gather(val, eq=True):
    op = operator.ge if eq else operator.gt
    return StoplossFunction(op, val)

def lesser(val, eq=True):
    op = operator.le if eq else operator.lt
    return StoplossFunction(op, val)
```

сохранить  
в файл



запустить  
из файла



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7 (r27:82525, Jul 4 2010, 09:01:59) [MSC v.1500 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informat
ion.
>>> |
```


Текстовый редактор

Этап  
компиляции  
отсутствует!

Интерпретатор



# Преимущества

- Нет компиляции
  - Syntax Sugar + строгие правила оформления кода
  - Динамическая (утиная) типизация
  - Лёгкая интеграция с кодом на C/C++
  - Легко масштабируется
  - Кроссплатформенность
  - Не требует IDE
  - Огромное количество качественных библиотек
  - Освоить может каждый
  - Интерпретатор распространяется бесплатно!
- 



# Библиотека MarketWizard

Фреймворк на Python, создан, чтобы ускорить разработку торгового робота, упростить его перенос между торговыми платформами и уменьшить стоимость его доработки.

- **Последовательная обработка данных** (нет постдиктивных ошибок + высокая скорость обновления)
- **Слабо связанные элементы** (лёгкая замена)
- **Универсальные провайдеры данных**
- **Универсальный интерфейс для разных торговых систем** (лёгкий перенос)
- **Открытый исходный код** (Open Source)

Реализован торговый робот для сайта [algoritmus.ru](http://algoritmus.ru)

Автоматически загружает состав индекса ММВБ, загружает данные по инструментам, входящим в индекс, производит анализ и выводит результат в шаблон отчёта. Полностью автономный код (нужен только Python) занимает всего 50 строк кода.

Библиотека на официальном  
сайте:

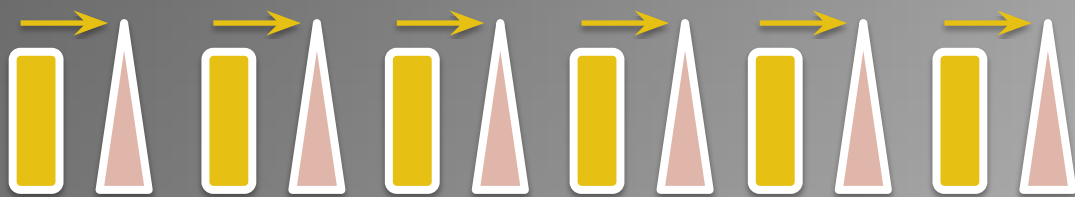
[pypi.python.org/pypi/marketwizard](http://pypi.python.org/pypi/marketwizard)



# Поточный расчёт индикаторов



Традиционный алгоритм



Поточный алгоритм в MarketWizard

# Слабо связанные элементы



Виртуальное табло  
с индикаторами



Новое виртуальное  
табло с  
индикаторами

Контейнер оптимизатора



Виртуальный  
трейдер

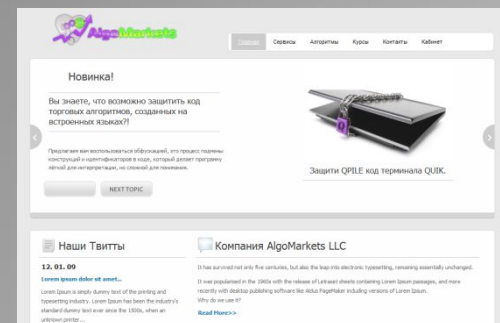
# Информация



Журнал F&O  
[www.fomag.ru](http://www.fomag.ru)



Сайт MarketWizard  
[marketwizard.algomarkets.ru](http://marketwizard.algomarkets.ru)



Сайт  
АлгоМаркетс  
[www.algomarkets.ru](http://www.algomarkets.ru)



[denis@algomarkets.ru](mailto:denis@algomarkets.ru)



[www.twitter.com/deniskolodin](https://www.twitter.com/deniskolodin)



**Спасибо за внимание!**