

Вятский государственный университет  
Цифровые кафедры  
Разработка прикладного программного обеспечения

# Говорим с компьютером на ОДНОМ ЯЗЫКЕ

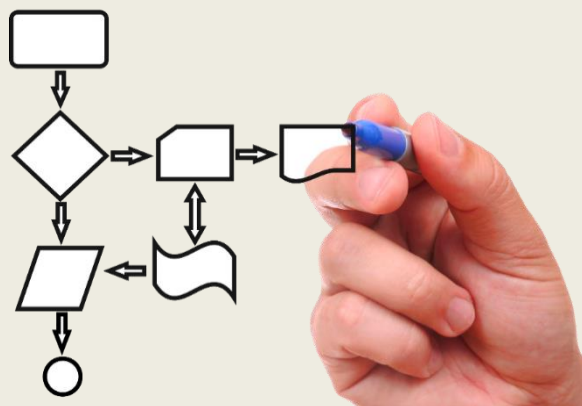
Долженкова Мария Львовна,  
кандидат технических наук, заведующий кафедрой ЭВМ  
Нижегородова Маргарита Владимировна,  
кандидат педагогических наук, доцент кафедры САУ  
Шмакова Наталья Александровна,  
старший преподаватель кафедры САУ  
Белиц Александр Борисович,  
специалист второй категории АО НПП Знак

Дата 17.10.2022



# Понятие «алгоритм»

Алгоритм – конечная совокупность точно заданных правил или набор инструкций, описывающих порядок действий исполнителя для решения определённой задачи.



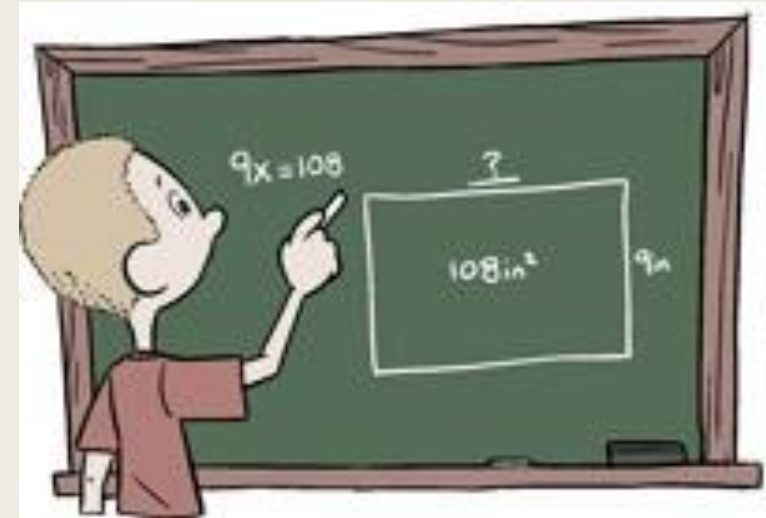
# Пример из математики

Узнать (измерить) длину одной из сторон прямоугольника и запомнить (или записать) ее;

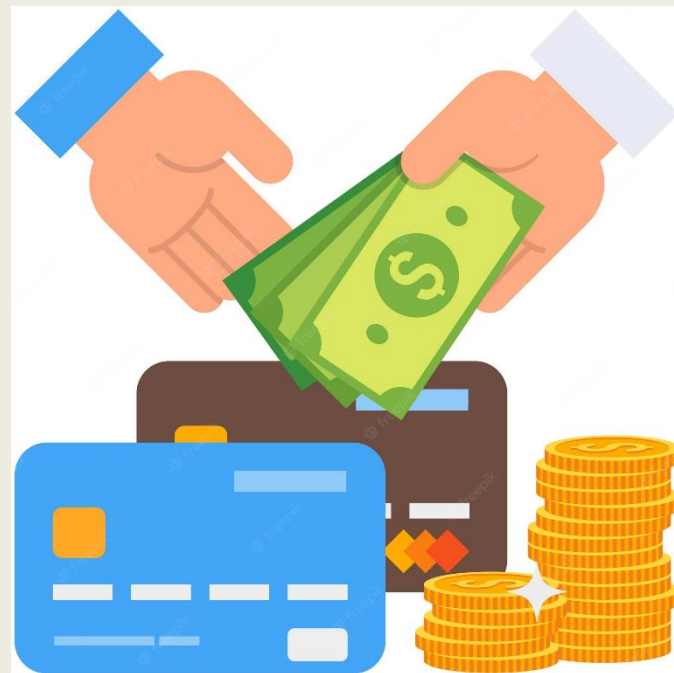
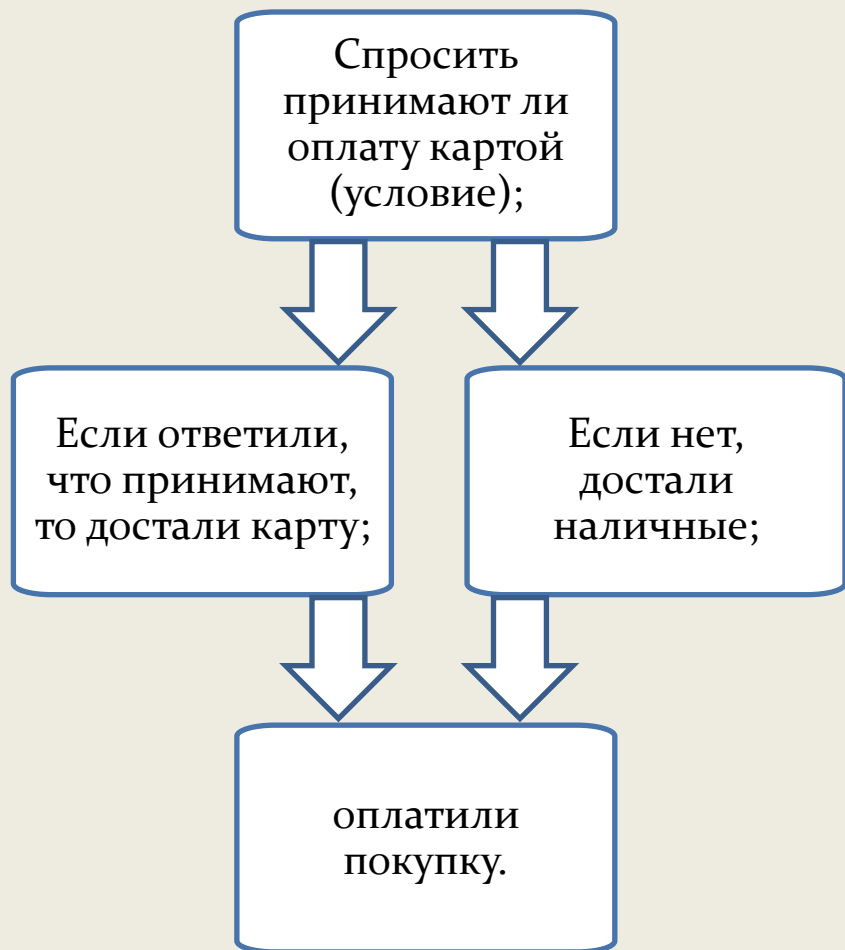
Узнать (измерить) длину второй стороны фигуры и запомнить (или записать) ее;

Рассчитать площадь, перемножив два найденных значения;

Сообщить результат товарищу.



# Пример из жизни

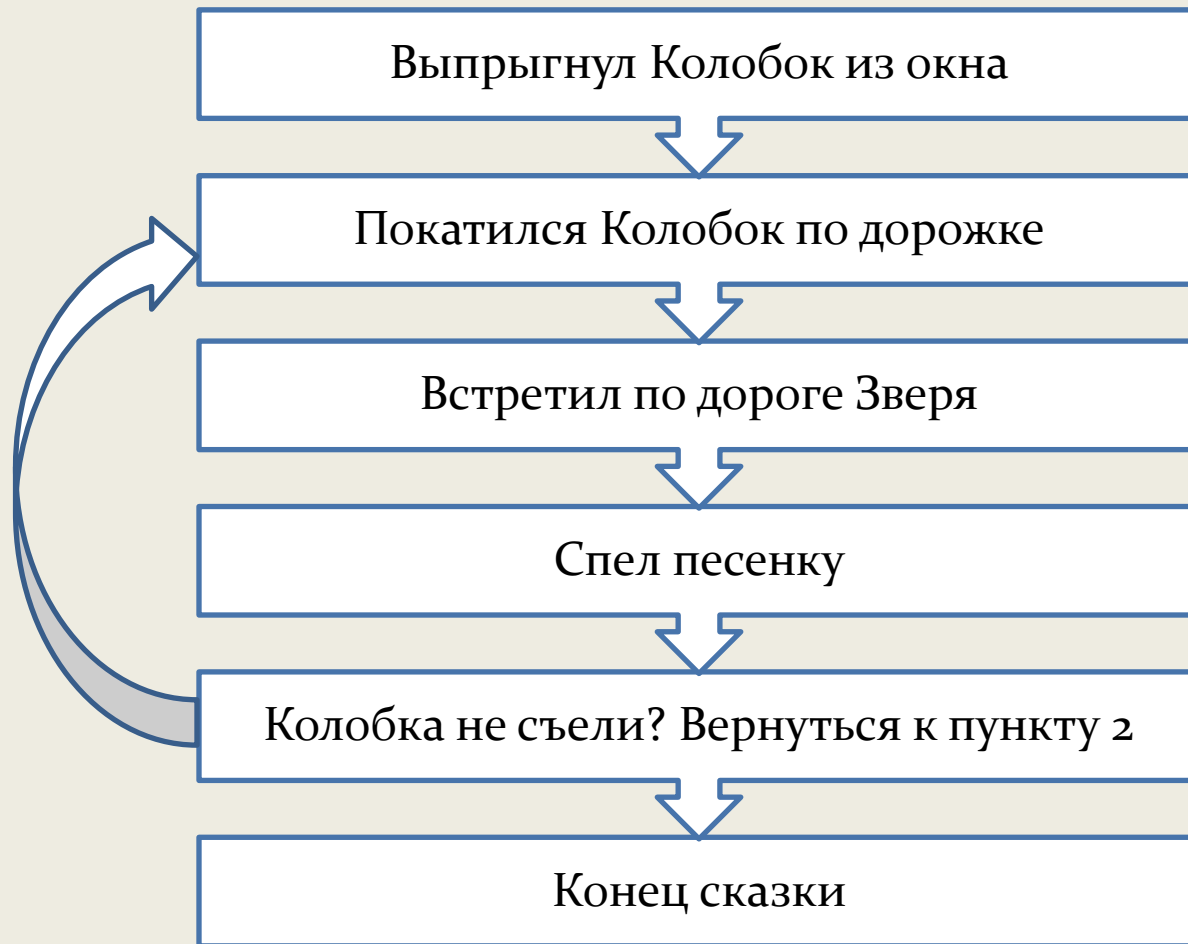


# Алгоритм

Алгоритм называется циклическим, если в нем имеются действия или наборы действий, которые необходимо выполнить более одного раза.

- Повторяющиеся алгоритмические действия являются телом цикла. Цикл имеет условие, по которому выполнение циклического алгоритма заканчивается.

# Пример из литературы. Сказка Колобок



Выпрыгнул колобок из окна

? **покатился** по дорожке

? **встретил** Зайца 🐰

Заяц 🐰 **хочет съесть** ?

? **поёт свою песенку** 🎤

? **покатился** по дорожке

? **встретил** Медведя 🐻

Медведь 🐻 **хочет съесть** ?

? **поёт свою песенку** 🎤

? **покатился** по дорожке

? **встретил** лису ?

Лиса ? **хочет съесть** ?

? **поёт свою песенку** 🎤

? поёт у Лисы на носу

? поёт у лисы на языке

Конец сказки

# Программа

Программа – это описание на формальном языке, «понятном» компьютеру, последовательности действий, которые необходимо выполнить над данными для решения поставленной задачи.



# А что такое — «язык программирования»?

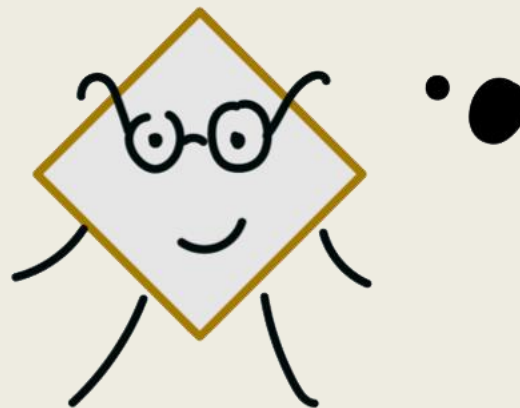




# Алфавит языка

Алфавит языка – набор символов, используемых в данном языке, который создан для «общения» человека с компьютером.

```
Var x =  
(2 x 2)  
+ 5;  
—
```



исходный код

# Символы алфавита

- Служебные (зарезервированные) слова.
- Имена (идентификаторы).
- Числа и символьные строки.
- Знаки операций и разделители. Комментарии.

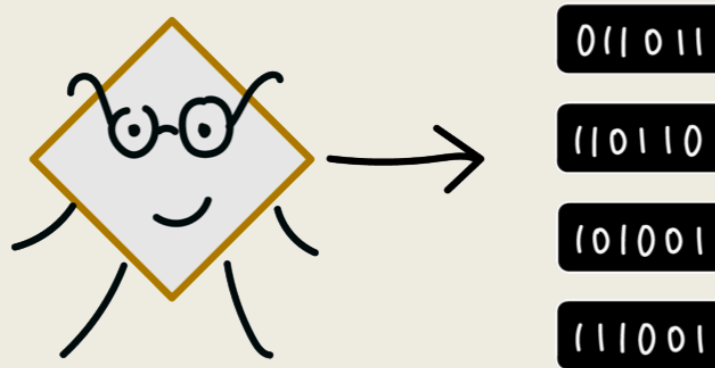
# Синтаксис и семантика языка

Синтаксис языка – это набор правил, определяющий допустимые конструкции языка.

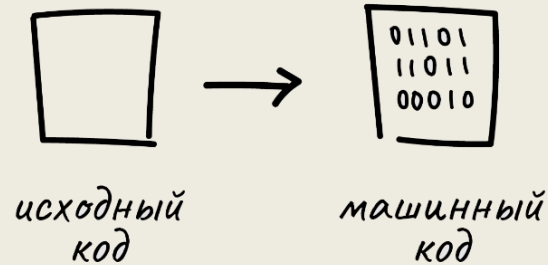
Семантика языка – задает значения для всех допустимых цепочек языка, служит для определения смысла и однозначного толкования языковых конструкций.

# Транслятор программы

- Программы на языках программирования высокого уровня оформляются в привычном для человека виде.
- Человеку-программисту удобно разрабатывать и читать программу, но компьютер такую программу не «поймет»!
- Для решения данной проблемы служит транслятор.

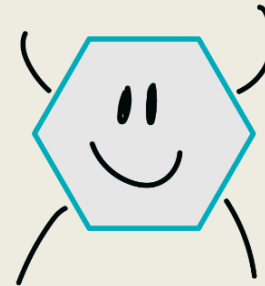


# Компилятор и интерпретатор

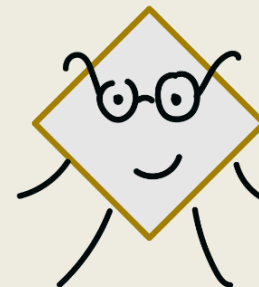


Как мы превращаем  
наш **исходный код**  
во что-то, что могут  
прочитать компьютеры?

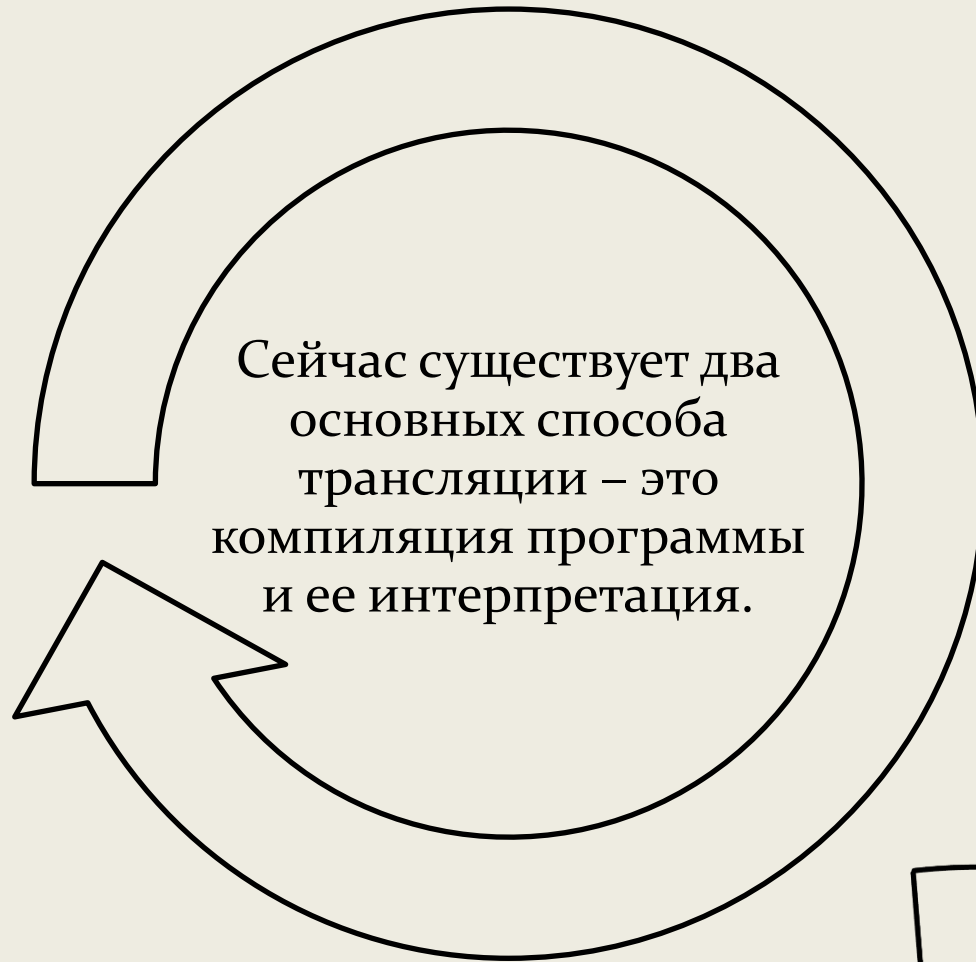
- > Совсем как в реальной жизни для перевода одного языка в другой мы обращаемся к **переводчикам!**



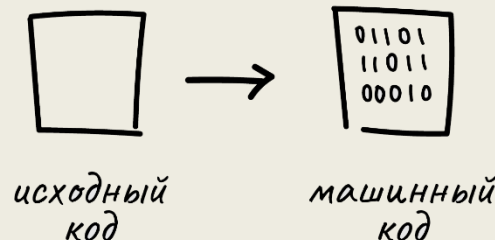
- > У нас есть два переводчика **Интерпретатор** и **Компилятор**, оба делают перевод исходного кода в машинный, но делают это по-разному!



# Компилятор и интерпретатор



- **Компилятор:** исходный код программы целиком переводится в машинный, создается исполняемый файл.
- **Интерпретатор:** считывает команды программы строка за строкой, последовательно переводятся в машинный код и исполняются на текущей программно-аппаратной платформе.



# Знакомьтесь — Python

- Первый прототип языка Python появился в 1989 году, он состоял из простой виртуальной машины, «парсера» и среды выполнения.
- Главная особенность заключалась в том, что будущий Python предлагал гибкую модель расширяемости.



# Знакомьтесь – Python

- На Python полностью или частично написаны многие компьютерные игры.
- Язык Python – незаменимый помощник системных администраторов и сетевых инженеров в автоматизации управления серверами и дата-центрами.
- На Python разрабатывают встраиваемые системы для различных устройств.





# Знакомьтесь – Python

- Задачи по работе с данными востребованы во всех областях. Python – один из наиболее часто используемых в Data Science и работе с естественным языком.
- Python – стандарт для создания и обучения искусственных нейронных сетей.
- С помощью него можно смоделировать любой объём данных для обучения модели.

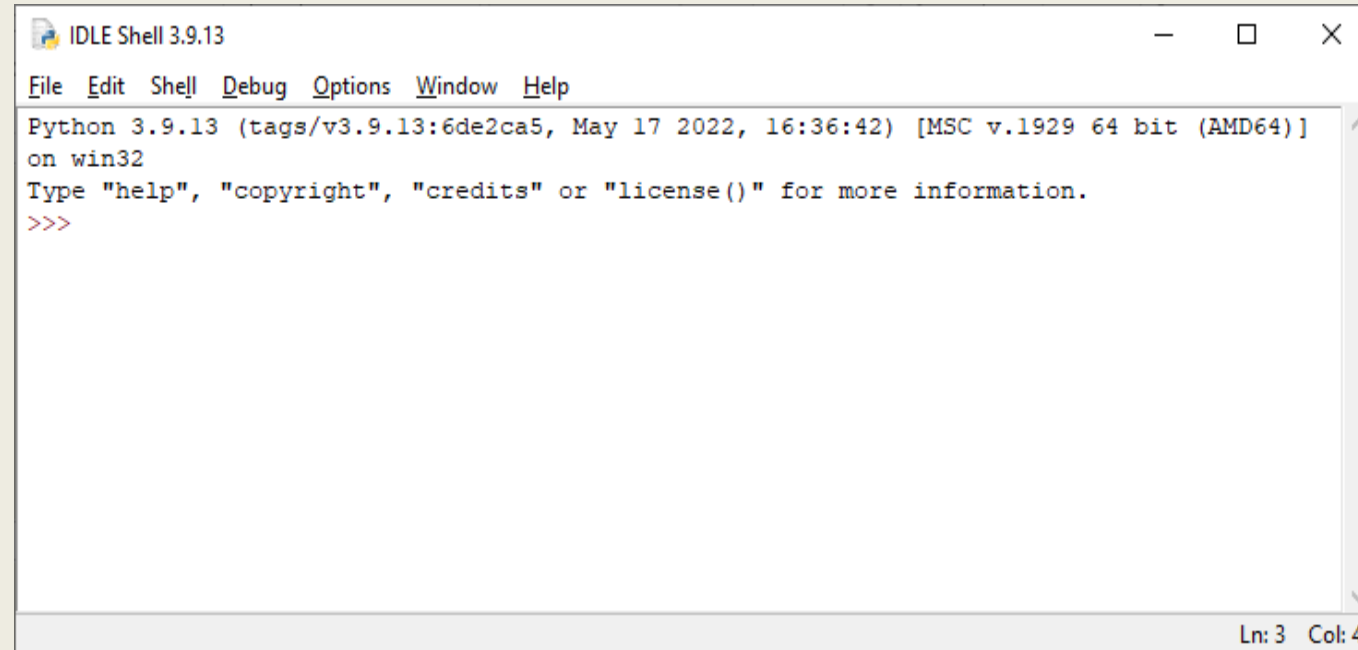


# Как и где писать на Python



- IDLE – редактор, поставляемый вместе с Python.
- Sublime Text – свободное программное обеспечение.
- Visual Studio Code – бесплатный редактор кода от Microsoft для Windows, Linux и MacOS.
- Jupyter Notebook – это веб-приложение с открытым исходным кодом.
- PyCharm – это интегрированная среда разработки специально для Python.
- Spyder – это мощная научная интегрированная среда программирования, написанная на Python, для Python.

# IDLE Python



+: Легкий, подходит для начинающих

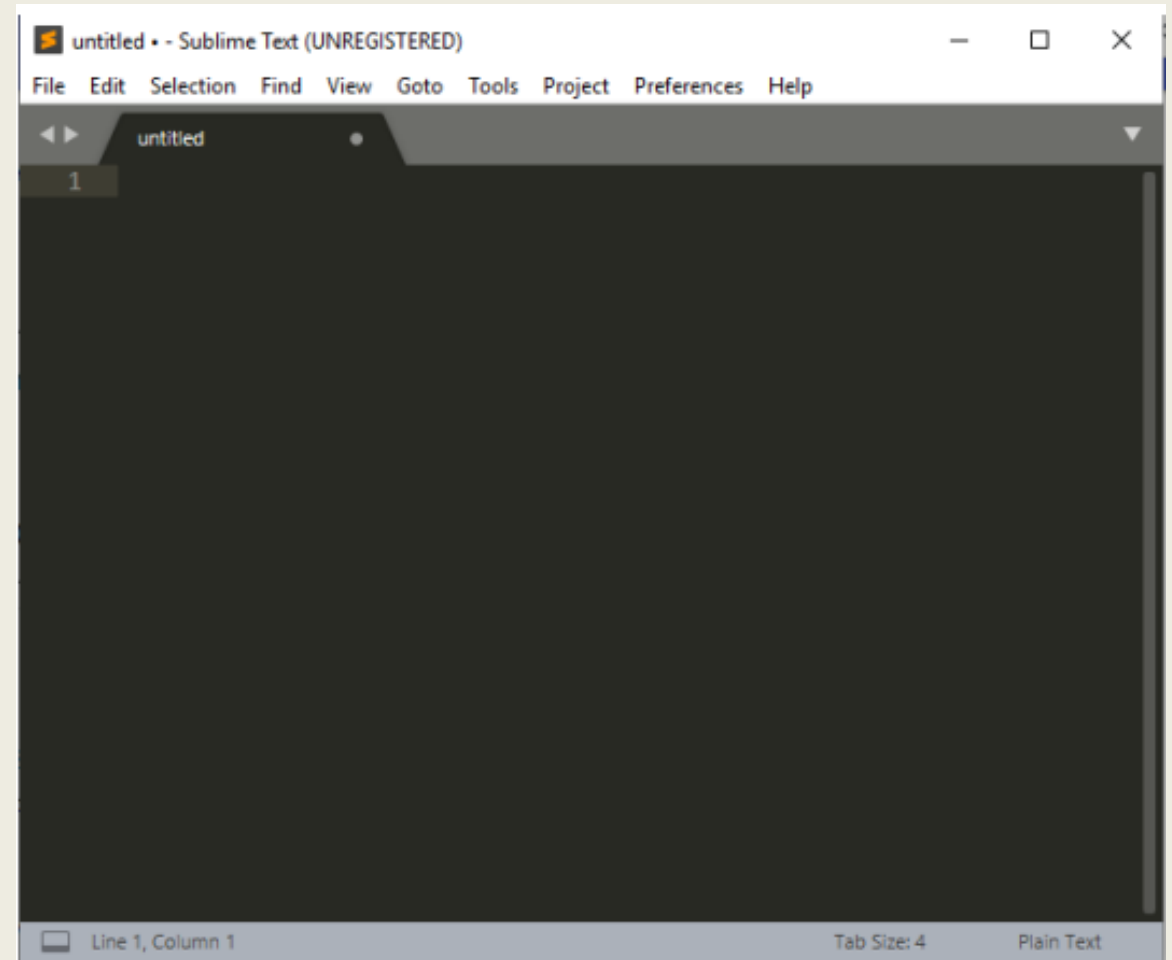
-: Не подходит для сложных проектов, не хватает продвинутых функций.



# Sublime Text

+: Простой и по большей части бесплатный, тонко настраивается, компактный и эффективный.

-: Для удобства требует дополнительных пакетов.

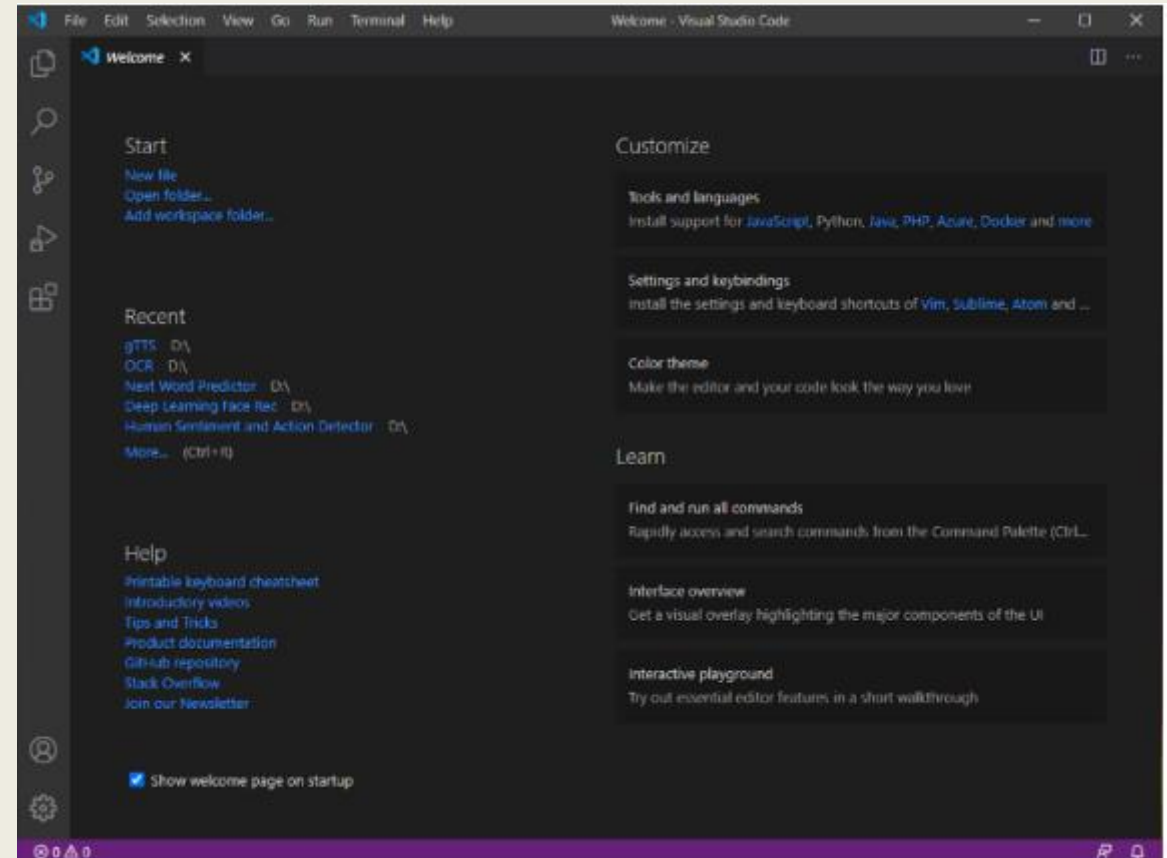




# Visual Studio Code

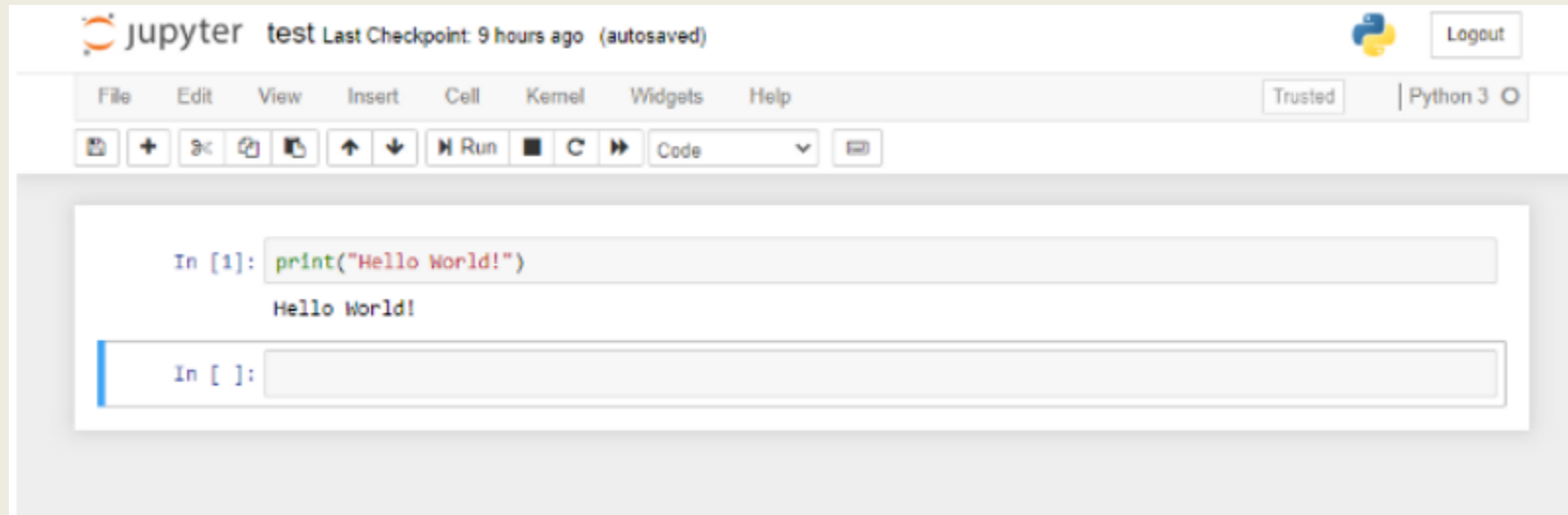
+: потребляет немного памяти по сравнению с другими громоздкими инструментами разработки, имеет встроенный терминал и прост в использовании.

-: Иногда терминал работает не так, как хотелось бы.





# Jupyter Notebook



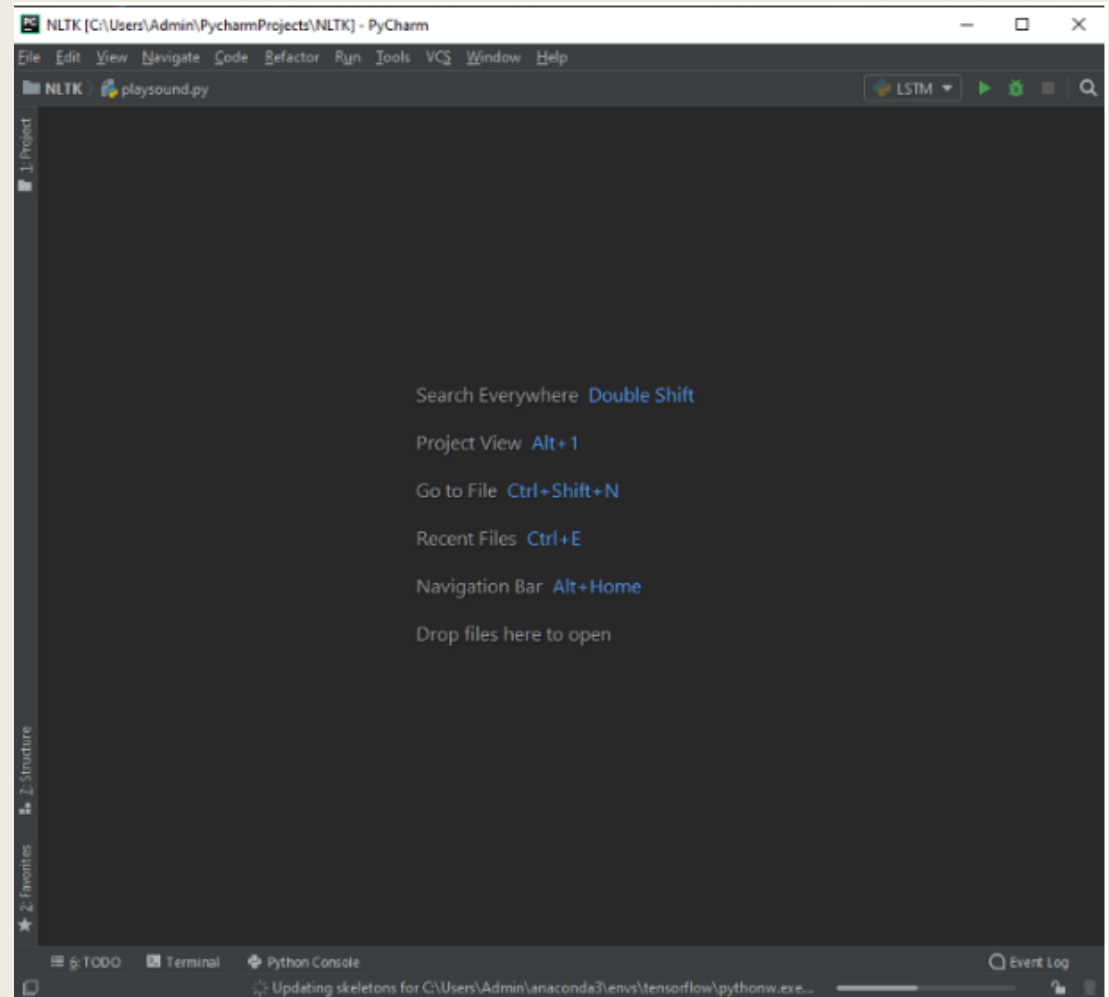
- + : Лучшая платформа для начала работы с наукой о данных, легко делиться файлами и визуализациями, разметка и другие дополнительные функции.
- : Нет мощных функций из некоторых IDE.



# PyCharm

+: разработана специально для Python, поддерживает виртуальные среды.

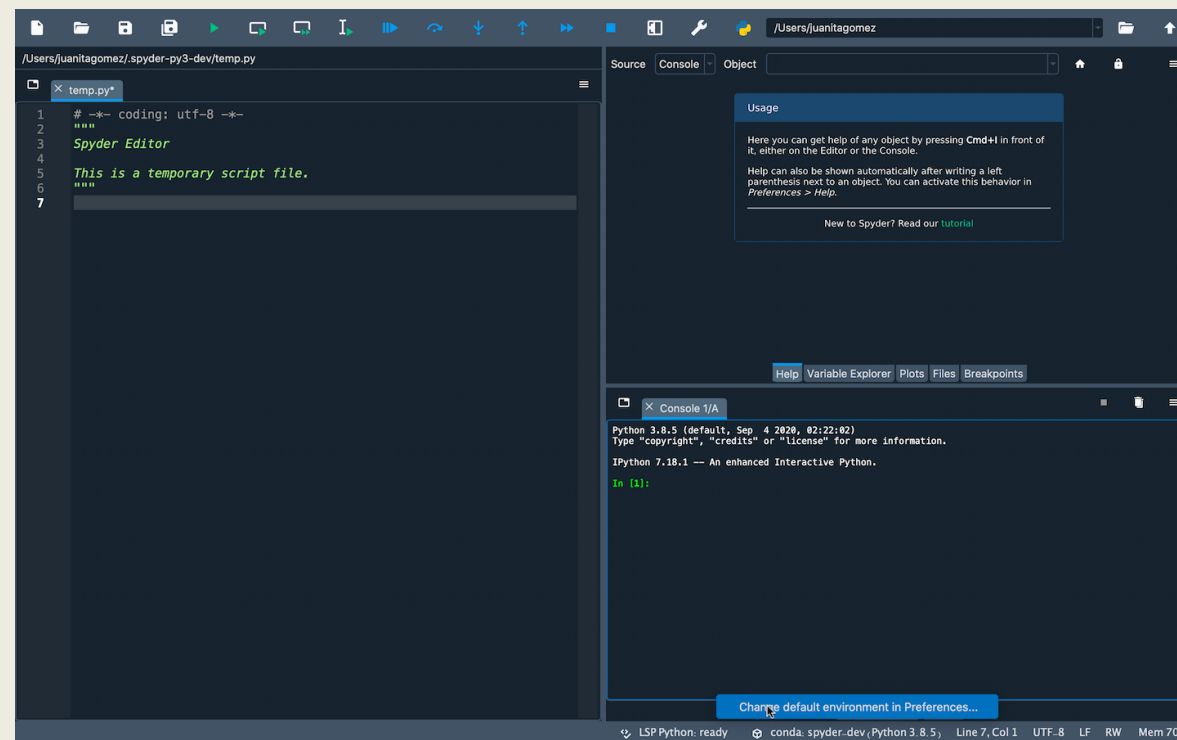
-: Требуется минимум 8 Гб оперативной памяти для стабильной работы





# Spyder

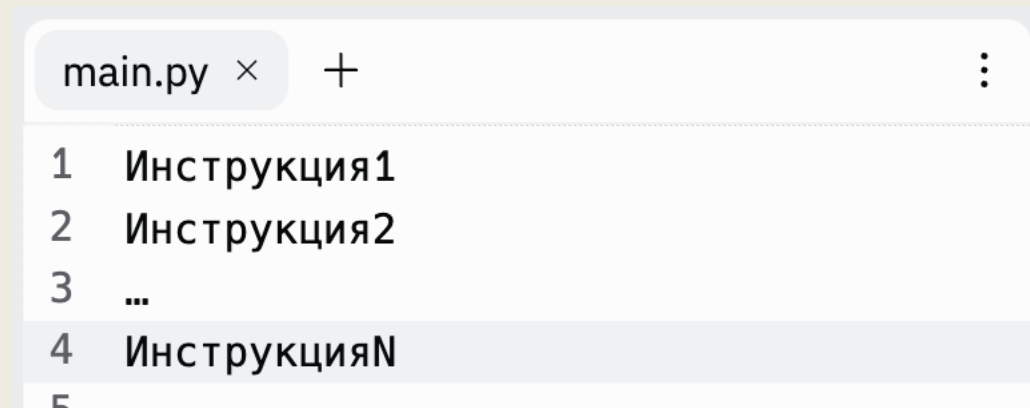
+ : хорошее рабочее окружение  
для просмотра интерпретаций  
и кодирования в два окна,  
широкий выбор опций  
исключительно для Python.  
- : Устаревший интерфейс.





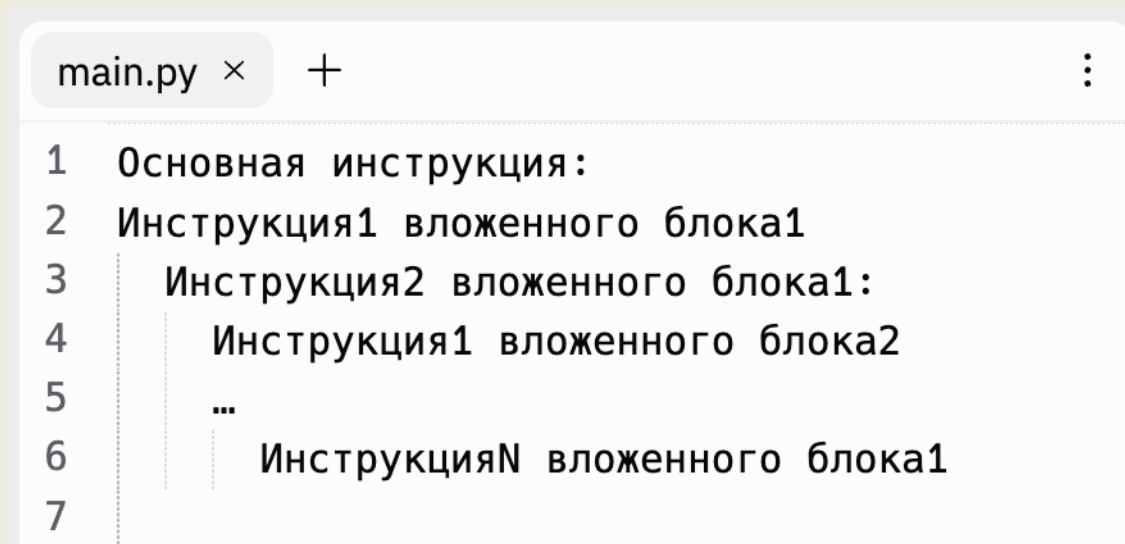
# Начинаем разговаривать

- Синтаксис языка Python, как и сам язык, очень прост.
- Конец строки является концом инструкции (точка с запятой не требуется).



# Вложенные инструкции

- Вложенные инструкции в Python записываются после двоеточия.
- Вложенные инструкции объединяются в блоки по величине отступов.
- Отступ может быть любым, главное, чтобы в пределах одного вложенного блока он был одинаков.



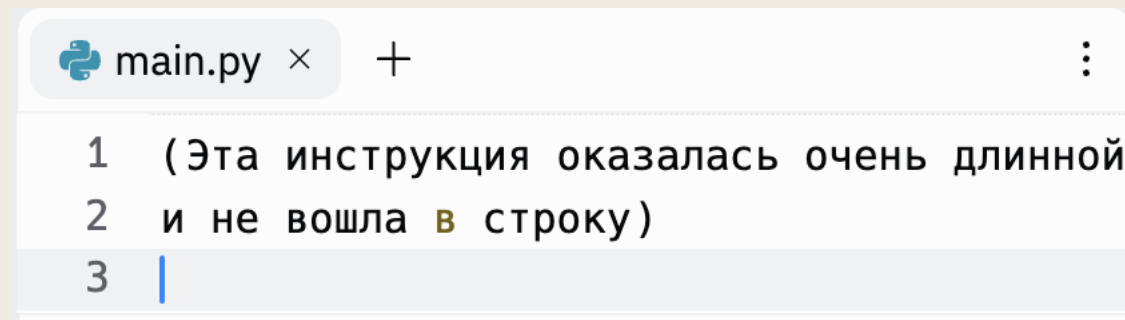
The screenshot shows a code editor window titled 'main.py' with a close button (x) and a plus sign (+) for additional tabs. The editor contains the following Python code:

```
1 Основная инструкция:
2 Инструкция1 вложенного блока1
3     Инструкция2 вложенного блока1:
4         Инструкция1 вложенного блока2
5         ...
6         ИнструкцияN вложенного блока1
7
```

The code demonstrates nested blocks using indentation. Line 1 is the main instruction. Lines 2-6 are indented under line 1, forming a block. Line 3 is further indented under line 2, forming a sub-block. Lines 4-6 are indented under line 3, forming a sub-sub-block. Line 7 is at the same level as line 1.

## Несколько специальных случаев

- Иногда возможно записать несколько инструкций в одной строке, разделяя их точкой с запятой.
- Допустимо записывать одну инструкцию в нескольких строках.



The screenshot shows a code editor window with a tab labeled 'main.py'. The code is written in Python and consists of a single long instruction split across three lines. Line 1 contains '(Эта инструкция оказалась очень длинной', line 2 contains 'и не вошла в строку)', and line 3 is empty with a cursor at the start. The lines are numbered 1, 2, and 3 on the left margin.

```
1 (Эта инструкция оказалась очень длинной
2 и не вошла в строку)
3 |
```

## Вывод информации на экран

- `print()` – это такая команда языка Python, которая выводит то, что в ее скобках на экран.
- Именно скобки `()` указывают, что `print` – это функция.

# Типы данных

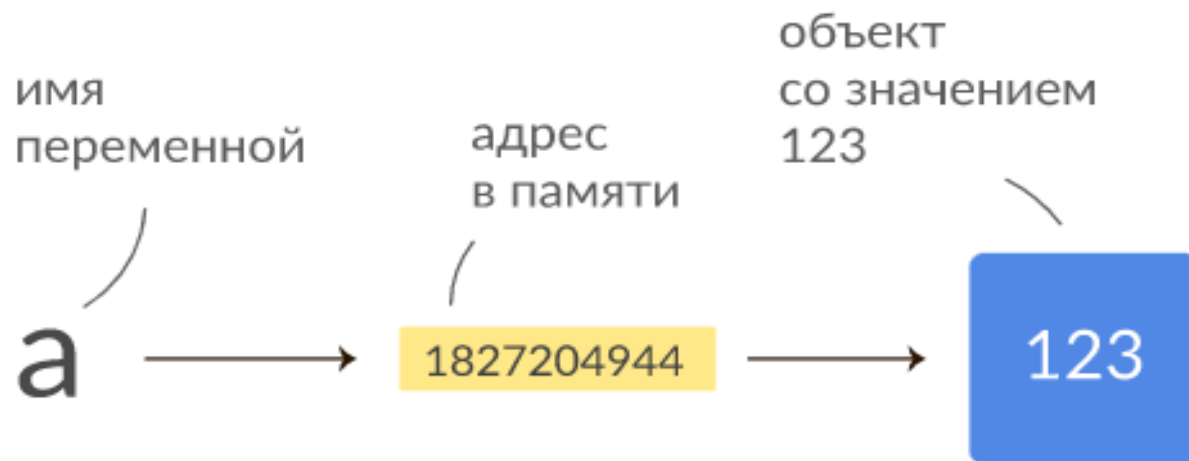
- Целые числа (тип `int`) – положительные и отрицательные целые числа, а также 0 (например, 4, 687, -45, 0).
- Числа с плавающей точкой (тип `float`) – дробные числа (например, 1.45, -3.789654, 0.00453).
- Строки (тип `str`) – набор символов, заключенных в кавычки (например, "ball", "What is your name?", 'dkfjUUv', '6589').

# Переменные

- Переменная – имя для доступа к области памяти компьютера, выделенной для хранения информации определенного типа.
- В именах переменных можно использовать буквы, цифры (но не начинаться с цифры) и знак подчеркивания «\_».
- Строчные и заглавные буквы различаются, то есть переменные с именами `dlina` и `Dlina` – это две разные переменные.
- Есть набор слов, которые нельзя использовать в качестве имен переменных, так как эти слова «зарезервированы»

# Переменные Python

Важно: переменная в Python не хранит значение напрямую – она хранит лишь ссылку на объект (адрес)!

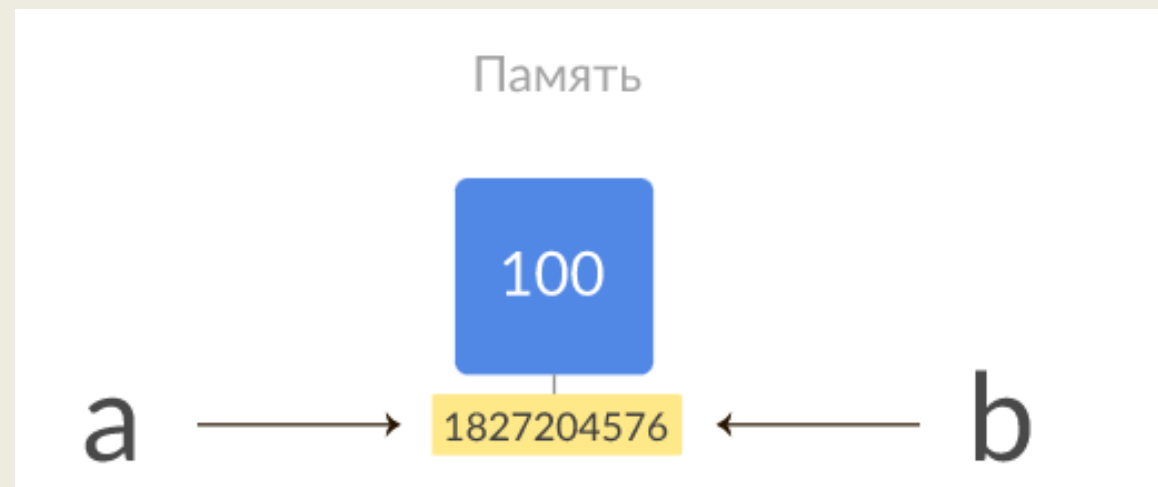


# Каскадное присваивание

- Запись  $a = b = 100$  называется каскадным присваиванием и равносильна паре инструкций.

$b = 100$

$a = b$

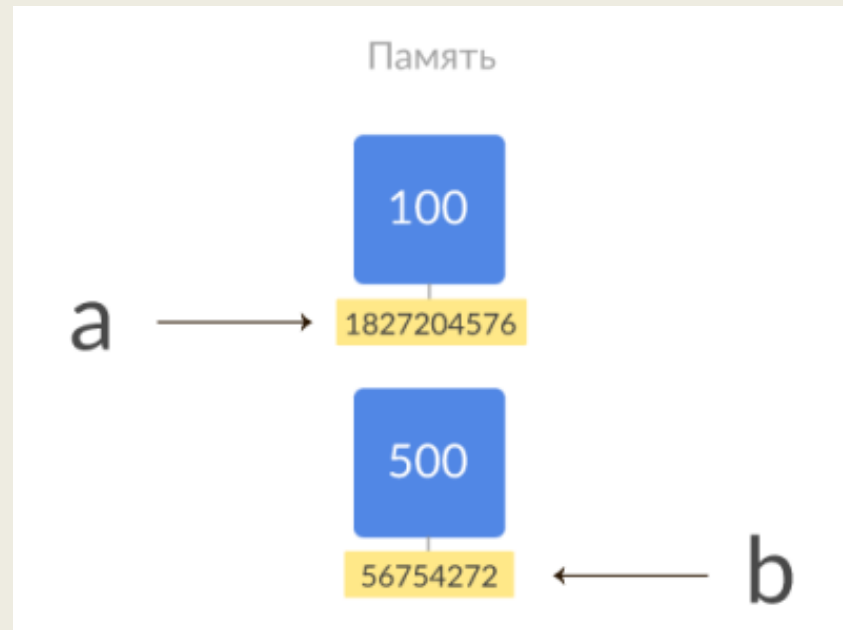




# Переменные Python

- Пусть в какой-то момент вы захотели поменять значение переменной `b`.

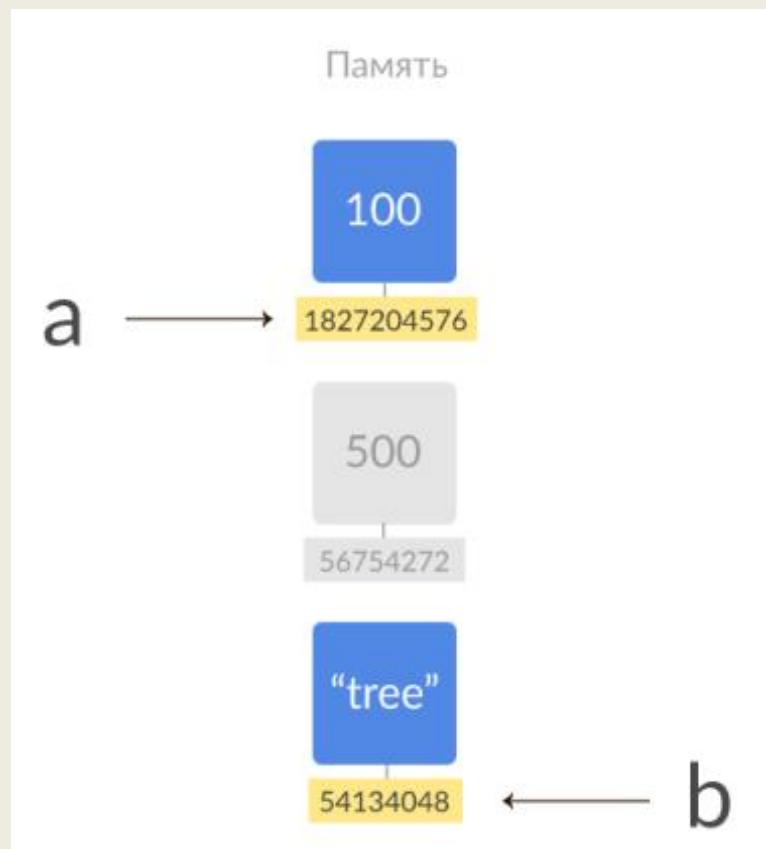
`b = 500`



# Переменные Python

- Если далее в программе вы присвоили `b` новое значение.

`b = 'tree'`



# Ввод данных в программу

- За ввод в программу данных с клавиатуры в Python отвечает функция `input()`.
- При вызове функции `input()` выполнение программы приостанавливается до тех пор, пока пользователь не введет текст на клавиатуре.

# Практическая работа

- Напишите программу, которая предлагала бы пользователю решить пример  $4.25 * 100 - 50/2$ . Потом выводила бы на экран правильный ответ и ответ пользователя. Подумайте, нужно ли здесь преобразовывать строку в число?
- Запросите у пользователя четыре числа. Отдельно сложите первые два и отдельно вторые два. Разделите первую сумму на вторую. Выведите результат на экран так, чтобы ответ содержал две цифры после запятой.