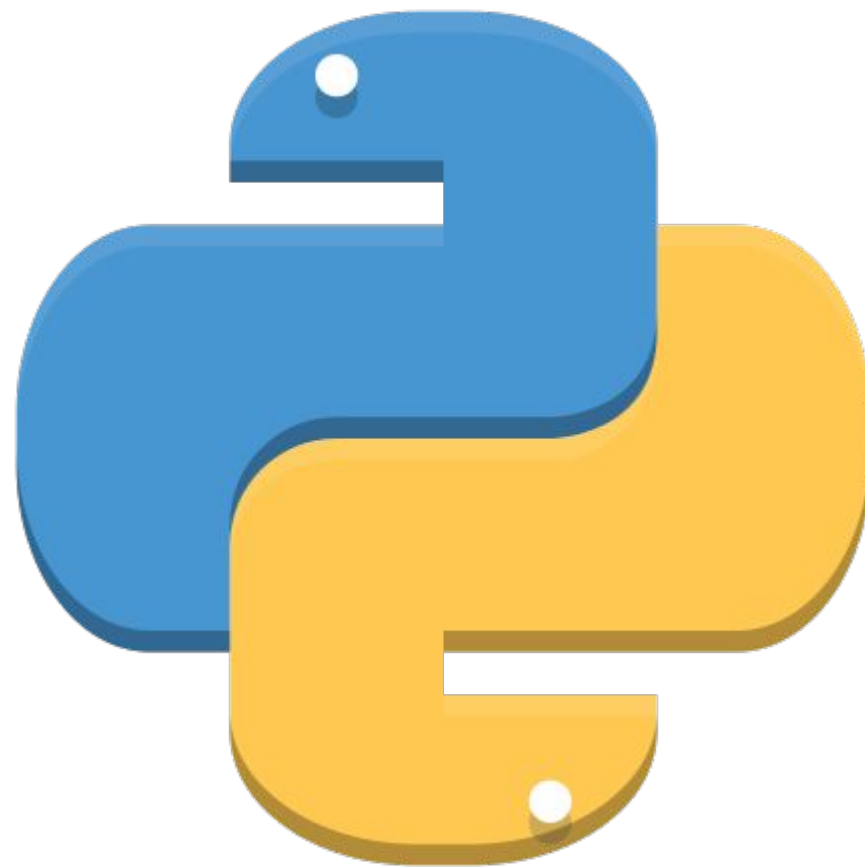


***Моя Школа***

сертифицированные IT курсы

# Введение



**Программист** — специалист, занимающийся созданием компьютерных программ.

Программистов можно условно разделить на три категории в зависимости от специализации:

**Прикладные программисты** занимаются в основном разработкой программного обеспечения прикладного характера — игры, бухгалтерские программы, редакторы, мессенджеры и т. п.

**Системные программисты** разрабатывают операционные системы, работают с сетями, пишут интерфейсы к различным распределенным базам данных.

**Веб-программисты** также работают с сетями, но уже с глобальными - Интернет. Они пишут программную составляющую сайтов, создают динамические веб-страницы, веб-интерфейсы для работы с базами данных. Они создают сайты, сервисы и веб-приложения — все те, которыми мы пользуемся ежедневно. Специалисты работают над видимой и серверной частями, чтобы мы могли полистать ленту с утра, отправить деньги другу, выучить язык или просто развлечься.

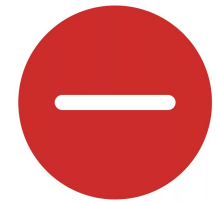


## Плюсы:

- ✓ востребованность разработчиков на рынке труда;
- ✓ высокая зарплата;
- ✓ возможность самостоятельного обучения.

## Минусы:

- ✓ относительно быстрое уменьшение актуальности технологий;
- ✓ неравномерное распределение работы.



## Языки программирования

```
graph TD; A[Языки программирования] --> B[Низкоуровневый язык программирования]; A --> C[Высокоуровневый язык программирования];
```

### **Низкоуровневый язык**

программирования  
Это такой язык, в котором каждая команда, написанная программистом, соответствует одной команде процессора.

### **Высокоуровневый язык**

программирования  
Это такой язык, в котором команды, написанная программистом, приближены к естественному (английскому) языку, легче воспринимаются человеком, не зависят от конкретного компьютера

## Языки программирования

```
graph TD; A[Языки программирования] --> B[Интерпретируемый язык программирования]; A --> C[Компилируемый язык программирования];
```

### **Интерпретируемый язык программирования**

Если программа написана на интерпретируемом языке, то интерпретатор непосредственно выполняет (интерпретирует) ее текст без предварительного перевода. При этом программа остается на исходном языке и не может быть запущена без интерпретатора. Можно сказать, что процессор компьютера — это интерпретатор машинного кода. Кратко говоря, интерпретатор переводит на машинный язык прямо во время исполнения программы.

### **Компилируемый язык программирования**

Программа на компилируемом языке при помощи специальной программы компилятора преобразуется (компилируется) в набор инструкций для данного типа процессора (машинный код) и далее записывается в исполняемый файл, который может быть запущен на выполнение как отдельная программа. Другими словами, компилятор переводит программу с языка высокого уровня на низкоуровневый язык, понятный процессору сразу и целиком, создавая при этом отдельную программу



1

Особенности  
языка

2

Работа со  
средой  
разработки

3

Основные типы  
данных



**Создатель языка Python** — нидерландский программист Гвидо Ван Россум. Он был участником проекта по написанию ABC, языка для обучения программированию. В конце 1989 года Гвидо приступил к разработке нового языка и задумал его как потомка ABC, способного к обработке исключений и взаимодействию с операционной системой Amoeba. Так и получился Python.

**Откуда такое название?** Многие разработчики считают, что язык назван в честь семейства змей, но это не так. Когда Гвидо работал над проектом, он любил смотреть комедийное шоу «Летающий цирк Монти Пайтона», поэтому и нарёк своё творение в честь британской комик-группы. Так что правильно произносить «Пайтон».

Python свободно распространялся через интернет и со временем у него появились последователи — люди, заинтересованные в развитии этого языка программирования. Первая публикация Python состоялась в феврале 1991 года — это была версия 0.9.0. В 1994 году Гвидо опубликовал Python 1.0, а потом одна за другой выпустились и другие версии: до 2.0 язык обновился в октябре 2000, до 3.0 — в декабре 2008. В октябре 2021 мир увидела самая свежая версия — Python 3.10.0.

Более подробную информацию о всех релизах, версиях и изменениях языка, а также собственно интерпретаторы и необходимые утилиты для работы и прочую полезную информацию можно найти на официальном сайте <https://www.python.org>

## Насколько Python популярен?

По данным GitHub на июнь 2021 года, Python входит в топ-5 самых популярных языков, уступая только JSX, JavaScript и Java.

В чём секрет такой популярности языка? Python подходит для решения широкого круга задач и применяться на всех популярных платформах. Росту популярности Python способствовала и его эффективность в стремительно развивающихся сферах Machine Learning и Data Science. Ещё одна причина — техническая. С развитием технологий увеличились вычислительные мощности персональных компьютеров и серверов, как результат — вырос спрос на интерпретируемые языки программирования, в том числе и на Python. Дело в том, что в таких языках помимо запуска самой программы необходим ещё запуск интерпретатора, а это требует дополнительных ресурсов. Раньше с этим были проблемы, но теперь они решены.

#	language	repos count
1	JSX	54 711 718
2	JavaScript	14 240 090
3	Java	9 827 505
4	Python	6 909 587
5	HTML	5 367 639
6	C#	2 893 002
7	C++	2 569 474
8	PHP	2 385 827
9	Jupyter Notebook	2 239 515
10	CSS	2 039 268

Топ популярных языков программирования летом 2021





## Установка компонентов

1. Для установки интерпретатора заходим на официальный сайт и разделе Downloads выбираем последнюю версию

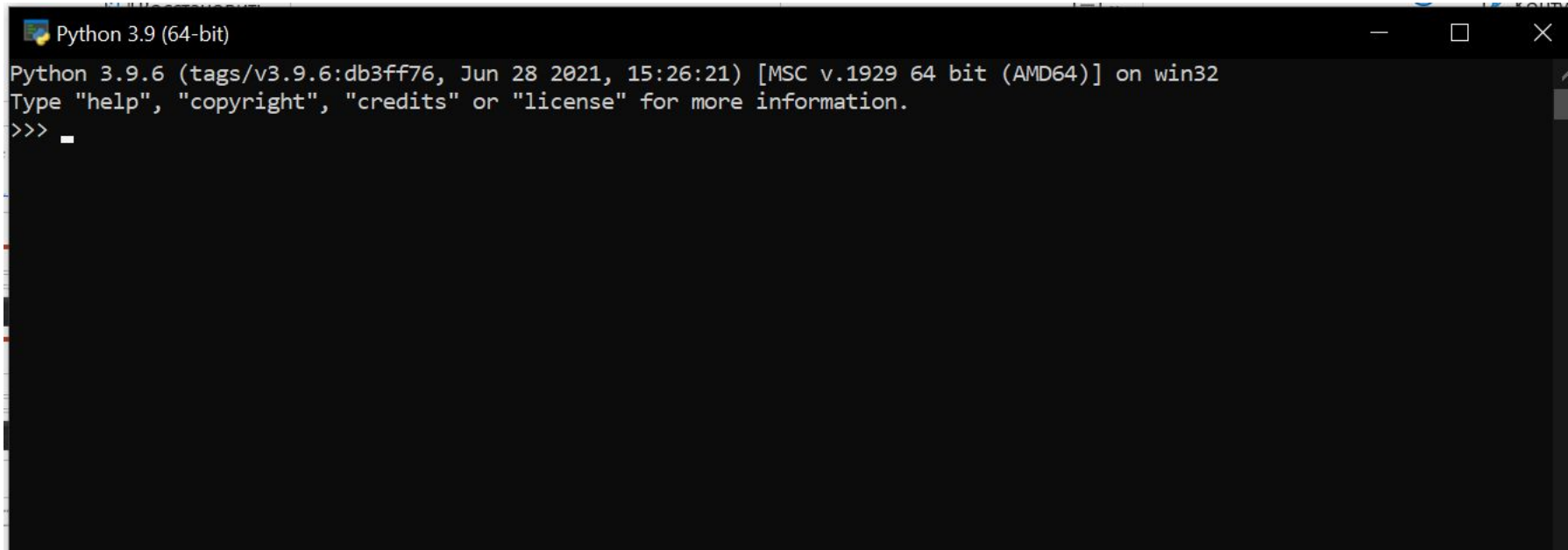
<https://www.python.org/downloads/>

2. Для установки среды разработки заходим на официальный сайт PyCharm -> нажимаем кнопку Download -> выбираем версию Community

<https://www.jetbrains.com/pycharm/download/#section=windows>



**Консоль (также называемая Shell)** — это, по сути, интерпретатор командной строки, который принимает данные от пользователя, то есть одну команду за раз, и интерпретирует их. Если он не содержит ошибок, он запускает команду и выдает требуемые выходные данные, в противном случае отображается сообщение об ошибке.

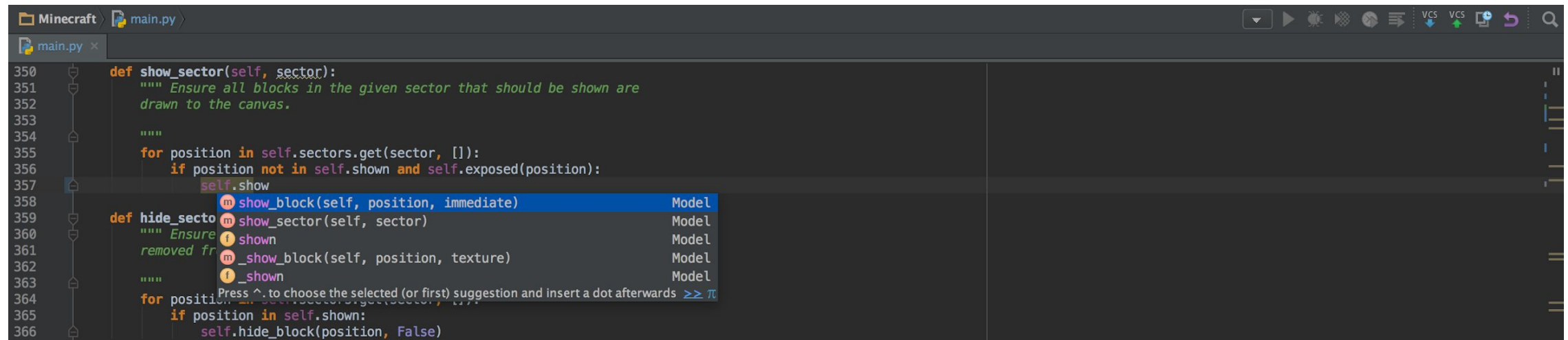


```
Python 3.9 (64-bit)
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

## Возможности PyCharm

PyCharm делает разработку максимально продуктивной благодаря функциям автодополнения и анализа кода, мгновенной подсветке ошибок и быстрым исправлениям. Автоматические рефакторинги помогают эффективно редактировать код, а удобная навигация позволяет мгновенно перемещаться по проекту.

PyCharm предоставляет большой набор инструментов из коробки: встроенный отладчик и инструмент запуска тестов, профилировщик Python, полнофункциональный встроенный терминал, инструменты для работы с базами данных.



```
def show_sector(self, sector):
    """ Ensure all blocks in the given sector that should be shown are
    drawn to the canvas.

    """
    for position in self.sectors.get(sector, []):
        if position not in self.shown and self.exposed(position):
            self.show
            self.show_block(self, position, immediate)
def hide_sector(self, sector):
    """ Ensure
    removed fr
    self.show_block(self, position, texture)
    self.show
    """
    for position in self.sectors.get(sector, []):
        if position in self.shown:
            self.hide_block(position, False)
```

Model

Model

Model

Model

Press ^ to choose the selected (or first) suggestion and insert a dot afterwards >>> π

## Что такое программа и из чего она состоит?

**Программа** – это алгоритм, записанный на каком-либо языке программирования, или же набор команд для компьютера

команда 1

команда 2

---

команда N

**Команда** – это описание действий, которые должен выполнить компьютер. (Пример команды: сложение двух чисел или запись данных в файл и т.п.)

## Разберем из чего состоит программа более детально

1. **Константа** – постоянная величина, имеющая имя, в питоне нет КОНСТАНТ.

Внимание!!!

В Python констант нет!

2. **Переменная** – изменяющаяся величина, имеющая имя (ячейка памяти).

3. **Функция** – вспомогательный алгоритм для выполнения вычислений (вычисление квадратного корня, сложение, вывод текста на экран).

Переменная — это именованная область памяти. После того как вы дали имя области, появляется возможность обращаться к данным, что в ней хранятся.

Каждый элемент данных в Python является объектом определенного типа или класса. Когда, в процессе выполнения программного кода, появляется новое значение, интерпретатор выделяет для него область памяти — то есть создаёт объект определенного типа (число, строка и т.д.). После этого Python записывает в свой внутренний список адрес этого объекта.

Чтобы создать новую переменную в Python, ее не нужно заранее инициализировать — достаточно придумать ей имя и присвоить значение через оператор =.

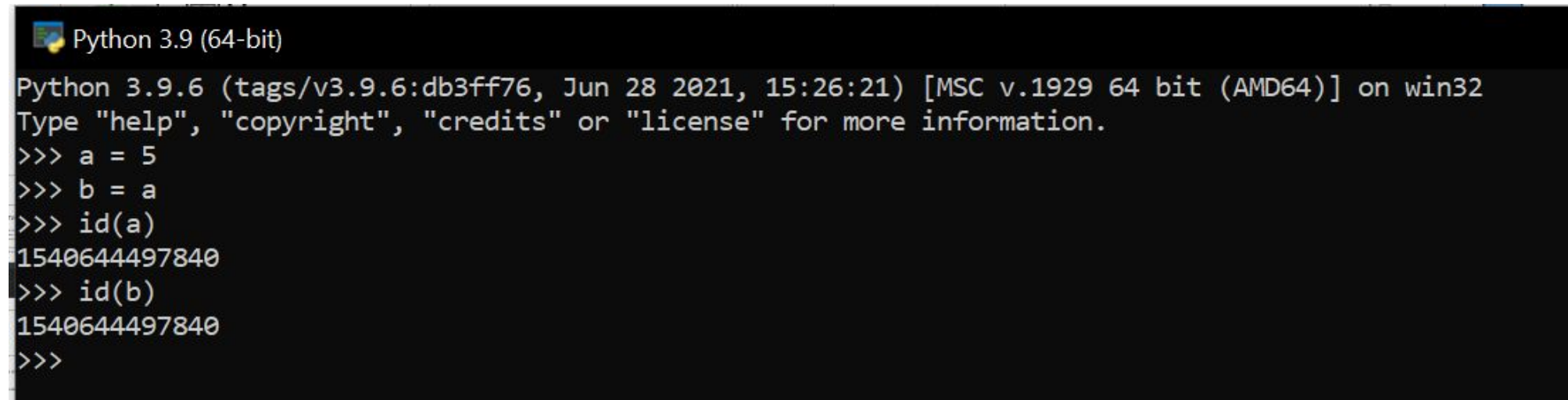
```
Python 3.9 (64-bit)
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 123
>>> print(a)
123
>>> print(type(a))
<class 'int'>
>>> print(id(a))
2817909217456
>>>
```



## Идентификатор объекта (Object Identity)

Идентификатор объекта — это адрес объекта в памяти.

В предыдущем примере мы вызывали функцию `id()`. Эта функция возвращает число, которое является неизменным и уникальным для каждого объекта на протяжении его жизненного периода:




```
Python 3.9 (64-bit)
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 5
>>> b = a
>>> id(a)
1540644497840
>>> id(b)
1540644497840
>>>
```

Видно, что объект здесь всего один. А `a` и `b` — по-разному названные переменные, которые на него ссылаются.

## Именованные переменные

В языке Python имя переменной должно состоять только из цифр, букв и знаков подчеркивания. И не должно начинаться с цифры.

Это жёсткий внутренний закон языка, помимо которого есть свод более мягких, но не менее важных правил, и они говорят нам:

- 
- давайте переменным имена, которые описывают суть объекта;
  - используйте единый стиль именования в рамках каждого проекта;
  - старайтесь не создавать наименования длиннее пятнадцати символов;

В каждом языке есть зарезервированные слова. Такие слова имеют специальное значение, и поэтому запрещены для использования в качестве имён переменных. Вот список зарезервированных слов для Python:

False, class, finally, is, return, None, continue, for, lambda, try, True, def, from, nonlocal, while and, del, global, not, with, as, elif, if, or, yield, assert, else, import, pass, break, except, in, raise.






## Основные типы данных

Когда мы имеем дело со значениями в программе, например, присваивая их для хранения переменной, нам приходится манипулировать различными типами этих значений (типами данных) такими, например, как числа или строки. В Python, как и в других языках программирования, имеется как встроенный комплект стандартных типов данных, так и возможность определять собственные пользовательские типы данных.


Давайте для первичного ознакомления перечислим некоторые основные типы данных встроенные в Python.

- 
1. Числа – это целые (-8, 0, 47), вещественные (-0.54 или -.54, 0.0, 2.475). Все они относятся к неизменяемым типам данных.
  2. Строки – это текст произвольной длины, взятый в одинарные, двойные или тройные кавычки. Например, '25 яблок', "Ok, Python!", ""Строка в тройных кавычках так"", ""или так.""". Строки также являются неизменяемым типом данных, хотя получить доступ к символу строки можно по его индексу.
  3. Логические значения – этот тип данных содержит всего две булевых константы: True и False.
  4. NoneType – специальное значение None, которое означает отсутствие значения.

Чтобы узнать тип объекта, используемого в программе, можно воспользоваться встроенной функцией `type(obj)`, которой в качестве аргумента нужно передать интересующий нас объект. Функция является встроенной в интерпретатор Python.



## Напишем свою первую программу



```
a = 5                # переменная типа int
pi = 3.14            # переменная типа float
s = "Hello, world!"  # переменная типа string
t = True             # переменная типа bool
```





## Комментарии

В Python признаком комментария является символ «решетки» #

При этом, интерпретатор Python игнорирует все символы, которые находятся после # и до конца строки.

Обратите внимание, что можно не только оставлять полезные заметки, поясняющие работу программу, но и временно «блокировать» выполнение той или иной команды.

Как сразу закомментировать много строк кода?

В PyCharm достаточно выделить нужный блок кода и нажать сочетание клавиш CTRL + ?

Если нужно раскомментировать много строк кода, то опять выделяем нужные строки и снова нажимаем сочетание клавиш CTRL + ?.



## Знакомство со встроенными функциями

Встроенная функция print

print() - служит для вывода информации на экран, в круглые скобки мы можем подавать либо переменную, либо текст

Теперь с помощью print выведем наши переменные:

```
print(a)
print(pi)
print(s)
print(t)
print("передал текст")
```

## Результат работы первой программы

```
5  
3.14  
Hello, world!  
True  
передал текст
```

# Динамическая типизация в Python

Динамическая типизация – это прием, при котором переменная связывается с типом в момент присваивания значения, а не в момент объявления переменной.

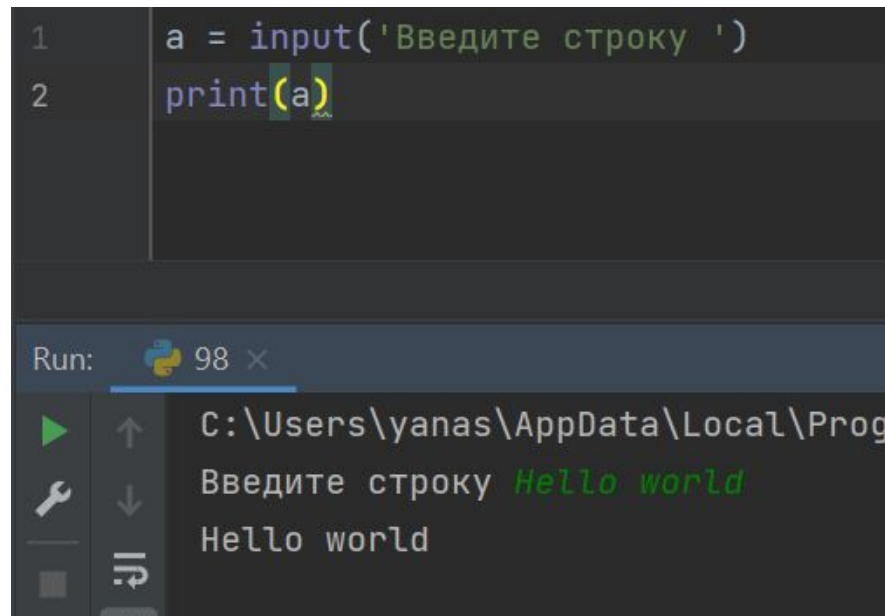
Это значит, что в Python программисту не нужно объявлять тип переменной в исходном коде. Интерпретатор делает это автоматически во время присвоения переменной нового значения. При этом он самостоятельно контролирует объем памяти, распределяемый под вновь создаваемый объект

```
Python 3.9 (64-bit)
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 23
>>> print(type(a))
<class 'int'>
>>> a = 'hello world'
>>> print(type(a))
<class 'str'>
>>> _
```

# Встроенная функция input

Для передачи в программу данных, вводимых пользователем с клавиатуры, используется встроенная функция `input()`, которая останавливает текущий поток программы до тех пор, пока пользователь не введет все данные и не нажмет Enter. При этом функции можно передавать в качестве аргумента строку с сообщением, которое будет выведено пользователю на экран. Таким сообщением обычно бывает подсказка о типе вводимых данных или призыв к какому-то действию.

Например:



```
1 a = input('Введите строку ')
2 print(a)
```

Run: 98 x

C:\Users\yanas\AppData\Local\Prog

Введите строку *Hello world*

Hello world

## Приведение типов

Довольно часто в программе требуется преобразовать значение одного типа данных в другой. Например, когда пользователь вводит данные для вычисления в строковом виде, перед вычислением возникает необходимость преобразовать строки в числа и только потом проводить вычисления. Если этого не сделать, то мы получим либо ошибку, либо совсем неожиданный результат.

Для того, чтобы мы могли считать число или переменную типа `bool`, `float`, `int` Нам нужно функцию `input()`, обернуть в `int()`, либо `float()`, либо `bool()`

```
a = int(input("Введите целое число"))      # переменная типа int
pi = float(input("Введите дробное число"))  # переменная типа float
t = bool(input("Введите значение True | False")) # переменная типа bool
```

```
print(a)
print(pi)
print(t)
```

```
Введите целое число 1
Введите дробное число 3.14
Введите значение True | False True
1
3.14
True
```





## Задание №1

Необходимо написать программу, которая требует у пользователя ввести два целых числа, строку и одно дробное число, далее выводит на экран строку и два целых числа и дробное число





## Решение

```
1 a = int(input("Введите первое целое число: "))
2 b = int(input("Введите второе целое число: "))
3 c = input("Введите строку: ")
4 d = float(input("Введите дробное число: "))
5
6 print(c)
7 print(a, b)
8 print(d)
```



## Задание №2

Создать 4 переменных. С типом данных целых чисел, чисел с плавающей точкой, логическим и строковым. Преобразовать числа в строку.





## Решение

```
a1 = 1
a2 = 2.0
a3 = True
a4 = "Hello"

a1 = str(a1)
a2 = str(a2)
```



## Задание №3

Запросить у пользователя имя, фамилию, отчество, возраст и город проживания.

Вывести на экран ФИО в одну строку.

Возраст и город в отдельных.



## Решение

```
name = input("Введите имя: ")
surname = input("Введите фамилию: ")
patronymic = input("Введите отчество: ")
age = int(input("Введите возраст: "))
city = input("Введите город: ")

print(surname, name, patronymic)
print(age)
print(city)
```



## Домашнее задание

Настроить PyCharm под себя (выбрать удобную тему, поставить заставку, нужный шрифт)



Почитать про встроенные инструменты для разработчиков

**Встроенные инструменты для разработчиков -  
Функциональность | PyCharm ([jetbrains.com](https://jetbrains.com))**

