

# Введение в Python



**Язык программирования** – это язык для записи алгоритмов и программ, близкий к естественному, исполнителем которых является компьютер

---

## Языки



### Естественные

(исторически созданные языки национальной речи, имеющие письменную и устную формы);

### Формальные

(искусственно созданные языки, имеющие только письменную форму).

**Программа** – это алгоритм, записанный на языке программирования.

Как и люди, компьютеры «говорят» на разных языках,  
только языки эти — компьютерные.

Компьютерный язык служит для того, чтобы  
переговариваться с компьютером, используя команды,  
понятные и компьютеру, и человеку.

Некоторые языки программирования названы в честь  
людей (например, Ада и Паскаль),

другие названия являются простыми акронимами, то  
есть аббревиатурой (к примеру, BASIC — от англ.  
Beginner's All-purpose Symbolic Instruction Code,  
универсальный код символических инструкций для  
начинающих), и уж совсем немногие языки названы в  
честь телевизионных шоу — как Python.

О да, язык программирования Python  
(произносится «Пайтон», с ударением на  
первый слог, хотя имейте в виду, что в  
России многие называют язык просто  
«питон»)

получил свое имя благодаря  
телешоу «Летающий цирк  
Монти Пайтона»,

так что змея питон здесь вовсе ни при чем

**Python** – язык программирования высокого уровня, предназначенный для самого широкого круга задач.

---

С его помощью можно обрабатывать различные данные, создавать изображения, работать с базами данных, разрабатывать Web-сайты.

Разработка языка Python была начата в конце 1980-х годов сотрудником голландского института CWI Гвидо ван Россумом.

Опубликован исходный текст в феврале 1991 года.

# Синтаксис



## Основные принципы синтаксиса языка Python

1. Конец строки является концом инструкции (";" не требуется).

```
a = 5
b = 3
print(a + b)
```

2. Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков.

```
if a == 5:
    print('yes')
    a += 1
```

3. Вложенные инструкции в Python записываются в соответствии с одним и тем же шаблоном, когда основная инструкция завершается двоеточием, вслед за которым располагается вложенный блок кода, обычно с отступом под строкой основной инструкции.

# Алфавит и словарь языка

Основой языка программирования является алфавит — набор допустимых символов, которые можно использовать для записи программы:

---

- ❑ латинские прописные и строчные буквы (A, B, C, ..., X, Y, Z, a, b, c, ..., x, y, z);
- ❑ русские прописные и строчные буквы (можно использовать, но это является очень плохим стилем);
- ❑ арабские цифры (0, 1, 2, ..., 7, 8, 9); специальные символы (знак подчёркивания; знаки препинания; круглые, квадратные скобки; знаки арифметических операций, # - знак комментариев и др.).

В качестве неделимых элементов (составных символов) рассматриваются следующие последовательности символов:

- ❑  $\geq$  и  $\leq$  (знаки  $\geq$  и  $\leq$ );
- ❑ «!», =,  $\neq$
- ❑ """" или ' ' ' и "" или # — утроенные кавычки или апострофы (начало и конец комментария)

# Идентификатор

— имя некоторого объекта в программе, являющееся его уникальным признаком, позволяющим отличать его от других объектов.

## Идентификатор:

- обязательно есть у **каждой** переменной, функции, объекта и т.п.;
- не ограничен по длине;
- чувствителен к регистру (*то есть **A** и **a** — это разные имена переменных или функций*).

# Идентификатор

## Допустимые символы:

- только символы латинского языка (от "A" до "Z")
- подчеркивание "\_"
- цифры от "0" до "9"

## НО:

- идентификатор **не может** начинаться с **цифры**
- должен быть **НЕ** ключевым словом

Служебные слова – это единые смысловые элементы с фиксированным значением.

# Имена=идентификаторы

---

Прописные и строчные буквы в именах различаются, например,

- f и F – две разные переменные.
- Длина имени может быть любой. Для удобства мы будем использовать имена, передающие смысл объект, состоящие не более, чем из 15 символов.
- Для обозначения переменных, программ и других объектов используются **имена (идентификаторы)** - любые отличные от служебных слов последовательности букв, цифр и символа подчёркивания, начинающиеся с буквы или символа

**НЕЛЬЗЯ** использовать в имени:

• ~~скобки~~

• ~~знаки +, =, !, ? и др.~~

# Служебные слова

Служебное слово языка Python	Значение служебного слова
and	и
break	прервать
else	иначе
False	ложь
float	вещественный (с плавающей точкой)
for	для
if	если
input	ввод
integer	целый
list	список
or	или
print	печать
string	строковый (цепочка символов)
True	истина
while	пока

# Переменные

**Переменная** — это простейшая именованная структура данных, в которой может быть сохранён промежуточный или конечный результат работы программы.

Чтобы создать переменную, нужно присвоить некоторому идентификатору значение при помощи оператора присваивания — знака "=".

**Пример:**

```
a = 10
b = 3.1415926
c = "Hello"
d = [1, 2, 3]
e = true
```

# Типы переменных



```
a = 10
b = 3.1415926
c = "Hello"
d = [1, 2, 3]
e = true
```

- переменная **a** хранит значение типа **int** (целое число),
- переменная **b** — типа **float** (число с плавающей точкой, т.е. вещественное число),
- переменная **c** — типа **str** (строка),
- переменная **d** — типа **list** (список, в данном случае из трех целых чисел),
- переменная **e** — типа **bool** (логический тип, где переменная может быть истиной (*true*) или ложью (*false*)).

# Типы переменных

- **Никакого специального объявления переменных не требуется**, первое присваивание переменной значения и является ее объявлением. Идентификатор в Python является "ссылкой" на хранимые в памяти данные.
- **Python — язык с динамической типизацией:** каждая переменная в каждый момент времени имеет определенный тип, но этот тип может меняться по ходу выполнения программы, достаточно просто присвоить ей новое значение другого типа. Определить, какой тип имеет переменная, можно с помощью команды `type()`.

## Программа

```
x = 4.0  
print(type(x))
```

выведет

```
<class 'float'>
```

Это означает, что созданная переменная `x` относится к типу (классу) `float`, т. е. хранит вещественное число. Заметьте, что дробная часть этого числа равна нулю, но мы специально указали её наличие с помощью точки. Это был сигнал для транслятора: «нужно создать вещественную переменную».

# Типы данных

Название	Обозначение	Допустимые значения	Пример
Целочисленный	int (integer)	сколь угодно большие, размер ограничен оперативной памятью	325
Вещественный	float	Любые числа с дробной частью	9.23 0.0 -1.7e-6
Строковый	str (string)	Любые символы из таблицы Unicode	"hello! " "x="
Логический	bool (boolean)	False и True	

В вещественном числе целая часть от дробной отделяется точкой, при этом перед точкой и после неё должно быть, по крайней мере, по одной цифре.

Пробелы внутри числа недопустимы.

**Тип переменной определяется в тот момент, когда ей присваивается новое значение.**

# Оператор присваивания

Общий вид оператора:

---

**<имя переменной> = <значение  
или вычисляемое выражение>**

**Примеры:**

$a = 25$

$b = \text{"Привет"}$

$c = 1.4 + 5.7 * a$

$d = a < c$

$e = \text{"мир"} + b$

$f = x * (a + c) / 3$

---

Нельзя указывать в правой части выражения переменные, которые не были заранее созданы (определены).

Так для переменных  $s$  и  $d$  все входящие переменные были заданы выше.

**Следующая строка ошибочна**

$$f = x * (a + c) / 3$$

так как переменная  $x$  из правой части ранее не была создана.

В Python разрешено множественное присваивание.

Запись  $a = b = 0$

равносильна паре операторов

$b = 0$  и  $a = 0$

---

**$a, b = 5, 3$  ИЛИ  $a = 5$  и  $b = 3$**

Так же часто используют сокращенную запись арифметических операций:

*сокращенная запись*

*полная запись*

$a += b$

$a = a + b$

$a -= b$

$a = a - b$

$a *= b$

$a = a * b$

$a /= b$

$a = a / b$

---

# Выражения и операции

**Выражение** – это конструкция, возвращающая значение некоторого типа.

Простыми выражениями являются переменные и константы.

Сложные выражения строятся из простых с помощью операций, функций и скобок. Данные, к которым применяются операции, называются **операндами**.

Используется линейная форма записи выражений (в одну строку).

## Арифметические операции

Операция	Обозначение	Пример
Сложение	+	$3 + 4 = 7$
Вычитание	-	$7 - 2 = 5$
Умножение	*	$2 * 2 = 4$
Деление	/	$8 / 2 = 4$
Целочисленное деление	//	$9 // 2 = 4$
Остаток от деления	%	$9 \% 2 = 1$
Возведение в степень	**	$2 ** 3 = 8$

# Основные операции

Основные операции с целыми (int) и вещественными (float) числами:

---

- $A + B$  — сумма;
- $A - B$  — разность;
- $A * B$  — произведение;
- $A / B$  — частное, *(результатом этого действия является вещественное число, даже если  $A$  нацело делится на  $B$ );*
- $A \% B$  — взятие остатка от деления  $A$  на  $B$ ;
- $A // B$  — взятие целой части от деления  $A$  на  $B$
- $A ** B$  — возведение в степень.

# Выражения и операции

Логические выражения могут содержать величины или выражения, которые сравниваются между собой с помощью операций сравнения.

Логическое выражение может принимать лишь два значения: «истина» или «ложь».

## Операции сравнения

Операция	Символы	Пример
равно	<code>==</code>	<code>x == 0</code>
не равно	<code>!=</code>	<code>x != 0</code>
больше	<code>&gt;</code>	<code>x &gt; 0</code>
меньше	<code>&lt;</code>	<code>x &lt; 0</code>
больше или равно	<code>&gt;=</code>	<code>x &gt;= 0</code>
меньше или равно	<code>&lt;=</code>	<code>x &lt;= 0</code>

# Приоритеты операций

Приоритеты операций в Python совпадают с приоритетом операций в математике, а именно:

- 1) Выполняются возведения в степень справа налево, то есть  $3 ** 3 ** 3$  это  $3 ** (3 ** 3)$ .
- 2) Выполняются унарные минусы (отрицания).
- 3) Выполняются умножения и деления слева направо. Операции умножения и деления имеют одинаковый приоритет.
- 4) Выполняются сложения и вычитания слева направо. Операции сложения и вычитания имеют одинаковый приоритет.
- 5) Для изменения порядка действий нужно использовать скобки.

# Выражения и операции

*Приоритет выполнения операций:*

- 1) операции в скобках;
- 2) возведение в степень;
- 3) умножение и деление (в том числе // и %);
- 4) сложение и вычитание.

Операции одинакового приоритета выполняются в порядке записи слева направо.

Если выражение слишком длинное и не помещается в одной строке, необходимо заключить всё выражение в скобки (перенос внутри скобок разрешён).

*Например:*  $\frac{(a+b)h}{2}$    $(a+b)^1 * h^2 / 2^3$

$$v + \frac{at^2}{2} \quad \longrightarrow \quad v + a^4 * t^2 * 2^1 / 2^3$$