# Встроенные функции

Язык программирования предназначен для записи компьютерных программ.

У любого языка программирования есть некоторый неизменный словарь, в первую очередь, состоящий из ключевых слов и названий встроенных функций.

**Встроенные функции – это функции или подпрограммы, имеющиеся в данной реализации языка программирования**. То есть, данный компилятор или интерпретатор точно будет обрабатывать известные ему встроенные функции.

У разных реализаций языка программирования может быть разный набор встроенных функций.

Когда мы говорим о языке программирования Python версии 3, мы имеем в виду **CPython** — наиболее распространённую, де-факто эталонную реализацию языка программирования Python. Она свободно доступна для скачивания по адресу <https://www.python.org/>

**Ключевые слова и названия встроенных функций нельзя использовать в качестве названий переменных!**

Список встроенных в язык Python 3 функций ([ссылка](https://docs.python.org/3/library/functions.html)):



Обратите внимание: **у функций в конце всегда стоит пара круглых скобок, открывающая и закрывающая.**

# Типы данных

В любом языке программирования есть встроенные средства для работы с данными. **В высокоуровневом языке программирования у данных всегда есть типы**.

Тип данных характеризует одновременно:

* **множество допустимых значений,** которые могут принимать данные, принадлежащие к этому типу;
* **набор операций**, которые можно осуществлять над данными, принадлежащими к этому типу.

Например, самый простой тип данных – целые числа. Они могут быть отрицательными и положительными. Максимальное и минимальное значение целого числа определяется конкретной реализацией конкретного языка программирования.

В CPython **версии 2** максимальное значение целого числа – 2147483647

В CPython **версии 3** максимальное значение целого числа ограничено лишь объемами доступной памяти.

Целые числа, как минимум, имеют стандартный набор арифметических операций: сложение, вычитание, умножение и деление.

*Если мы говорим о низкоуровневых языках, то используется* ***низкоуровневое определение типа****— как заданных размерных и структурных характеристик ячейки памяти, в которую можно поместить некое значение, соответствующее этим характеристикам.*

Типы могут назначаться «раз и навсегда» ([**сильная, строгая типизация**](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F)) или позволять себя изменять ([**слабая типизация**](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%B0%D0%B1%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F)).

**В Python имеются встроенные типы:**

* [булевый](https://ru.wikipedia.org/wiki/%D0%91%D1%83%D0%BB%D0%B5%D0%B2%D1%81%D0%BA%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF),
* [строка](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D0%BE%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D1%82%D0%B8%D0%BF),
* [Unicode](https://ru.wikipedia.org/wiki/Unicode)-строка,
* целое число произвольной точности,
* число [с плавающей запятой](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B2%D0%B0%D1%8E%D1%89%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BF%D1%8F%D1%82%D0%B0%D1%8F),
* [комплексное число](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BB%D0%B5%D0%BA%D1%81%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE)

и некоторые другие.

**Из**[**коллекций**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BB%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D1%8F_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29)**в Python встроены:**

* [список](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%28%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29),
* [кортеж](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D1%82%D0%B5%D0%B6_%28%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29) (*неизменяемый список*),
* [словарь](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2),
* [множество](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%28%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%29)

и другие.

# Некоторые распространённые типы данных

**Логический тип**

Логические, или булевы значения (по фамилии их изобретателя — Буля), могут иметь лишь одно из двух состояний — «истина» или «ложь». В разных языках обозначаются bool, BOOL, или boolean.

**Целое**, **целочисленный тип данных** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *integer*) — один из простейших и самых распространённых [типов данных](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) в [языках программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Служит для представления [целых чисел](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE).

*Множество чисел этого типа представляет собой* ***конечное***[*подмножество*](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE) *бесконечного множества целых чисел,* ***ограниченное*** [***максимальным***](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%BA%D1%81%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82)***и*** [***минимальным***](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BD%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82) ***значениями.***

В программировании различают беззнаковые целые числа и целые числа со [знаком](https://ru.wikipedia.org/wiki/%D0%97%D0%BD%D0%B0%D0%BA_%28%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29).

**Число с плавающей запятой** (или **число с плавающей точкой**) —

[экспоненциальная форма](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C) представления [вещественных (действительных) чисел](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D1%89%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE), в которой число хранится в виде [мантиссы](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C) и [порядка](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C) ([показателя степени](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BA%D0%B0%D0%B7%D0%B0%D1%82%D0%B5%D0%BB%D1%8C_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8)). При этом число с плавающей запятой имеет фиксированную относительную [точность](https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%8C) и изменяющуюся абсолютную.

Проще говоря, число с плавающей точкой – дробное значение.

**Как писать число с плавающей точкой? Правильно – через точку**:

PI = 3.1415926535897

**Неправильно:**

PI = 3,1415926535897

Так как в некоторых, преимущественно [англоязычных](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) и [англофицированных](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F%22%20%5Co%20%22%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) странах при записи чисел целая часть отделяется от дробной точкой, то в терминологии этих стран фигурирует **название «плавающая точка»** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***floating point***). Так как в [России](https://ru.wikipedia.org/wiki/%D0%A0%D0%BE%D1%81%D1%81%D0%B8%D1%8F) целая часть числа от дробной традиционно отделяется запятой, то для обозначения того же понятия исторически используется термин «плавающая запятая».

Название «плавающая запятая» происходит от того, что запятая в позиционном представлении числа (десятичная запятая, или, для компьютеров, двоичная запятая) может быть помещена где угодно относительно цифр в строке.

*На заметку.* **Скорость выполнения компьютером операций с числами**, представленными в форме с плавающей запятой, измеряется во [FLOPS](https://ru.wikipedia.org/wiki/FLOPS) (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *floating-point operations per second* и является одной из основных единиц измерения быстродействия вычислительных систем.

# Структурное программирование

**Структурное программирование** — [парадигма программирования](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), в основе которой лежит представление программы в виде иерархической структуры [блоков](https://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D0%BA_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29).

**Структурное программирование** направленно на повышение четкости, качества и времени разработки компьютерной программы путем широкого использования

1. Условных конструкций (**условные выражения if… elif…else**)
2. повторения (**циклы while и for**)
3. **блочных структур и подпрограмм** (обычно – именованные функции).

Проще говоря, в структурном программировании особое внимание уделяется организации кода в виде блоков. Переход к структурному программированию – необходимый шаг при разработке достаточно сложных программ.

В программировании важно научиться с самого начала разбивать свою программу на блоки, каждый из которых будет выполнять одну задачу.

*На заметку.* Термин родился в конце 1960-х — начале 1970-х годов на фундаменте [теоремы Бёма — Якопини](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%91%D1%91%D0%BC%D0%B0_%E2%80%94_%D0%AF%D0%BA%D0%BE%D0%BF%D0%B8%D0%BD%D0%B8), математически обосновывающей возможность структурной организации программ, и **работы Эдсгера Дейкстры«О вреде оператора goto»** (англ. Goto considered harmful).

# Документирование исходного кода

Кроме структурного программирования, важно документировать свой программный код. В языке программирования Python для этого есть правила:

* В начале программы (текстового файла с расширением .py) следует писать комментарий, заключенный в тройные кавычки:

""" Программа для вычисления чисел Фибоначчи

"""

* После определения функции следует писать комментарий, заключенный в тройные кавычки:

def fib(n):

    """Вычисляем числа Фибоначчи, меньшие числа n.

    n - переменная, получаемая как аргумент функции.

    """

Как писать комментарии?

* **Пишите комментарии так, чтобы они были понятны любому программисту, в первый раз увидевшему ваш код!**
* **Пишите комментарии так, чтобы они напомнили вам смысл и особенности вашей программы даже через несколько лет после написания!**

Полный код программы для вычисления чисел Фибоначчи:

#

# Самостоятельная работа

Посмотрите на программу **FizzBuzz**, разобранную в предыдущий раз.

Какие блоки можно выделить в этой программе?

Документируйте файл с этой программой.

По желанию, для тех, кто знаком с функциями: перепишите программу так, чтобы программа FizzBuzz использовала функцию с обязательными аргументами:

1. число остановки
2. на какие числа следует проверять делимость

Можно взять в качестве примера код программы вычисления чисел Фибоначчи