


# Приоритет выполнения операций



mango-office.ru

РЕКЛАМА

Виртуальная АТС Расширенная

2 000 ₽

Узнать больше

[Справочник по языку Python3.](#) / Приоритет выполнения операций

Выражение - это код, который интерпретатор Python вычисляет для получения значения. Простейшими выражениями являются литералы и идентификаторы. Другие выражения строятся посредством объединения подвыражений с помощью операторов, выполняющих соответствующие операции и/или разделителей.

Нижe, операции/выражения перечислены [в порядке уменьшения приоритета](#). Операции с более высоким приоритетом выполняются до выполнения операций с более низким приоритетом. Операции, указанные в одной строке, имеют одинаковый приоритет.

1. Создание словаря, генератор словарей:

{key: expr, ... }  
{key: expr for item in Iterator if conditional}
2. Создание множества, генератор множеств:

{expr, ... }  
{expr for item in Iterator if conditional}
3. Создание списка, генератор списков:

[expr, ... ]  
[ expr for item in Iterator if conditional ]
4. Создание кортежа, подвыражение (просто скобки), выражение-генератор:

(expr, ... )  
(expr for item in iterator)
5. Обращение к атрибуту:

x.attr
6. Вызов функции:

f(expr, ... )
7. Выделение среза:

x[start: stop: step]
8. Взятие элемента по индексу:

x[index]
9. Возведение в степень (x в степени y):

x\*\*y
10. Побитовое НЕ (инверсия):

~x
11. Унарные "плюс" и "минус", тождественность:

+x, -x
12. Деление: истинное и с округлением вниз:

x / y, x // y
13. Остаток от деления, формат:

x % y
14. Умножение, повторение:

x \* y
15. Вычитание, разность множеств:

x - y
16. Сложение и конкатенация:

x + y

17. Сдвиг значения  $x$  влево или вправо на  $y$  битов:

$x \ll y$ ,  $x \gg y$

18. Побитовое И, пересечение множеств:

$x \& y$

19. Побитовое исключающее ИЛИ, симметрическая разность множеств:

$x \wedge y$

20. Побитовое ИЛИ, объединение множеств:

$x | y$

21. Операторы проверки на равенство:

$x \neq y$ ,  $x == y$

22. Операторы сравнения, проверка на подмножество и надмножество:

$x < y$ ,  $x \leq y$ ,

$x > y$ ,  $x \geq y$ ,

23. Проверка идентичности (тождественности) объектов:

$x \text{ is } y$ ,  $x \text{ is not } y$

24. Проверки на входжение/принадлежности (для итерируемых объектов и множеств):

$x \text{ in } y$ ,  $x \text{ not in } y$

25. Логическое НЕ (отрицание):

$\text{not } x$

26. Логическое И, значение  $y$  вычисляется, только если значение  $x$  истинно:

$x \text{ and } y$

27. Логическое ИЛИ, значение  $y$  вычисляется, только если значение  $x$  ложно:

$x \text{ or } y$

28. Трехместный оператор выбора, значение  $x$  вычисляется, только если значение  $y$  истинно:

$x \text{ if expr else } y$

29. Создание анонимной функции:

$\text{lambda } x, \dots: \text{expr}$

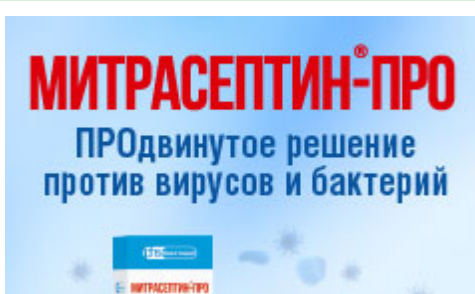
30. Поддержка протокола `send` в функциях-генераторах:

`yield x`

#### Примечание:

- Символы `...` означают запятые, присоединяющие нуль или более повторений предыдущего компонента. Во всех подобных случаях замыкающая запятая является необязательной, и ее наличие ни на что не влияет.
- Синтаксическая конструкция `[...]` используется для определения литералов [списков](#) и [выражений-генераторов](#) списков. В последнем случае предполагается выполнение цикла и накопление результатов в новом списке.
- Синтаксическая конструкция `(...)` используется для определения [кортежей](#), подвыражений и выражений-генераторов - разновидности генераторов списков, которая воспроизводит значения по требованию.
- Синтаксическая конструкция `{...}` используется для определения литералов [словарей](#), для определения литералов [множеств](#), генераторов словарей и множеств.
- Инструкция [yield](#) в функциях-генераторах возвращает аргументы функции. Если инструкция `yield` - не единственное, что находится справа от оператора присваивания, она должна заключаться в круглые скобки.
- Трехместный оператор выбора `x if expr else y` является краткой формой записи многострочной инструкции [if ... else](#).
- [Операторы сравнений](#) могут объединяться в цепочки: например `x < y < z` воспроизводит тот же результат, что и конструкция `x < y and y < z`.
- выражение извлечения [среза](#) `x[i:j:k]` является эквивалентом операции индексирования с применением объекта среза: `x[slice(i, j, k)]`.

ХОЧУ ПОМОЧЬ  
ПРОЕКТУ





[DOCS-Python.ru](https://docs-python.ru)<sup>™</sup>, 2023 г. (Внимание! При копировании материала ссылка на источник обязательна) [@docs\\_python\\_ru](https://t.me/docs_python_ru)