



Python

Приложение. Встроенные функции Python

Математические функции

abs(*x*)

Возвращает абсолютное значение (модуль) числа *x*. Аргумент может быть `int` или `float` числом.

pow(*x*, *y*, [*z*])

Возводит *x* в степень *y*. Если задан *z* — возвращается остаток от деления на *z*. Т.о. конструкция будет эквивалента `x**y%z`, но вычисление будет более эффективным. Все числа должны быть целыми.

round(*number*, [*ndigits*])

Возвращает число с плавающей точкой, округленное до *ndigits* знаков после запятой. По умолчанию *ndigits* = 0.

Для встроенных типов, поддерживающих `round()`, значения округляются до ближайшего числа, кратного 10 в степени минус *ndigits*; если два кратных одинаково близки, округление производится в сторону ближайшего четного (так, например, как `round(0,5)` и `round(-0.5)` равны 0, а `round(1.5) == 2`). Возвращаемое значение представляет собой целое число, если вызывается с единственным аргументом, в противном случае из того же типа, что и аргумент.

Обратите внимание, поведение `round()` для `float` может быть не очевидным: например, `round(2,675, 2)` дает 2,67 вместо ожидаемого 2,68. Это не ошибка: это результат того, что большинство десятичных дробей не может быть представлен так, как `float`.

Работа и итераторами (списки, словари и т.п.)

all(*iterable*)

Возвращает True, если все элементы аргумента-[итератора](#) (списка, кортежа, строки) соответствуют True, или если сам аргумент-итератор пуст.

any(*iterable*)

Возвращает True, если хотя бы один элемент аргумента-итератора соответствуют True. Если аргумент пуст, то возвращается False:

divmod(*a, b*)

Берет два числа и возвращает пару чисел: частное и остаток.

enumerate(*sequence*[, *start*=0])

Возвращает enumerate (нумерованный) объект, в качестве *sequence* выступает любой итерируемый объект. Пример использования функции:

filter(*function, iterable*)

Создает список из тех элементов итератора, функция которых вернет True.

len(*s*)

Возвращает длину (количество элементов) объекта. Аргумент может быть последовательностью (строка, кортеж или список) или отображением (словарь).

max(*iterable*[, *args...*][, *key*])

Возвращает максимальный элемент данного итератора.

min(*iterable*[, *args...*][, *key*])

Возвращает минимальный элемент данного итератора.

range(*[start]*, *stop*[, *step*])

Создает список с арифметической прогрессией. Чаще всего используется в цикле for.

reversed(*seq*)

Возвращает перевернутый итератор.

Преобразования (конструкторы) типов

bin(*x*)

Преобразование целого числа в двоичную строку.

bool(*[x]*)

Преобразование значения в булевы, используя стандартную процедуру проверки. Если *x* соответствует False или опущен, возвращает False, в противном случае функция возвращает True.

complex([*real*[, *imag*]])

Создает комплексное число из аргументов `real + imag*j` или преобразует строку или число к комплексному числу.

dict([*arg*])

Создание нового словаря, данные берутся из `arg`.

float([*x*])

Приводит строку или число к типу `float`.

hex(*x*)

Конвертирует целое число в шестнадцатиричную строку.

int([*x*[, *base*]])

Конвертирует строку или число в простое целое. В случае, если строка представляет собой восьмеричный или шестнадцатеричный литерал, то в качестве `base` указывается основание - 8 или 16.

list([*iterable*])

Возвращает список, элементы которого такие же и находятся в том, что и в `iterable`. `iterable` может быть последовательностью, контейнером, поддерживающим итерации и итератором. Если `iterable` уже список, будет сделана и возвращена копия, подобно `iterable[:]`. `list('ABC')` возвращает `['A', 'B', 'C']`, `list((1, 2, 3))` возвращает `[1, 2, 3]`. Если аргумент не задан, возвращается пустой список `[]`.

oct(*x*)

Выводит целое число в восьмеричном представлении.

open(*filename*[, *mode*[, *bufsize*]])

Открывает файл, возвращает объект типа `file`. Если файл не может быть открыт — выбрасывается исключение `IOError`. Для открытия файлов лучше пользоваться функцией `open()`, вместо использования конструктора `file` напрямую.

set([*iterable*])

Возвращает новый объект-множество, с элементами взятыми из `iterable`.

Работа со строками и символами

chr(*i*)

Получает в качестве аргумента целое число, соответствующего численному аргументу по Unicode-таблице. Возвращает строку из одного символа, соответствующего этому аргументу.

ord(c)

Получает в качестве аргумента строку единичной длины, возвращает целое число, соответствующего численному аргументу по Unicode-таблице.

Работа с объектами

getattr(object, name[, default])

Для object, name возвращается object.name, при этом name должно быть строкой. Например getattr(x, 'foobar') эквивалентно x.foobar.

hasattr(object, name)

Аргументы — объект и строка. Результат True, если строка это имя одного из объектов атрибута или False, если нет.

Переменные

globals()

Возвращает словарь, представляющий таблицу глобальных переменных. Это всегда словарь текущего модуля (то есть модуль, в котором написана функция или метод, а не модуль, из которого они вызываются).

id(object)

"Устанавливает личность" объекта, соответствующую адресу объекта и его представлению в виде числа.

Ввод/вывод

input([prompt])

Запрашивает данные у пользователя.

print([object, ...][, sep=' '][, end='\n'][, file=sys.stdout])

Выводит данные.

Вспомогательные функции

dir([object])

Без передачи аргумента возвращает список имен в текущей локальной области. При передаче аргумента возвращает список допустимых атрибутов для данного объекта.

help([object])

Вызывает встроенную систему помощи, очень полезную в интерактивном режиме. Если в качестве аргумента передается строка, то она ищется в именах модулей, функций, классов, методов, заголовков документации и

страницы помощи выводятся на экран. Если аргументом служит любой другой объект, страница помощи генерируется в соответствии с объектом.