**http://pythonworld.ru/osnovy/vstroennye-funkcii.html#more-166**

**Встроенные функции, выполняющие преобразование типов**

**bool(X)** - преобразование к типу bool, использующая стандартную процедуру [проверки истинности](http://pythonworld.ru/osnovy/instrukciya-if-elif-else-proverka-istinnosti-trexmestnoe-vyrazhenie-ifelse.html). Если х является ложным или опущен, возвращает значение False, в противном случае она возвращает True.

**bytearray([источник [, кодировка [ошибки]]])** - преобразование к bytearray. Bytearray - изменяемая последовательность целых чисел в диапазоне 0<=X<256. Вызванная без аргументов, возвращает пустой массив байт.

**bytes([источник [, кодировка [ошибки]]])** - возвращает объект типа [bytes](http://pythonworld.ru/tipy-dannyx-v-python/bajty-bytes-i-bytearray.html), который является неизменяемой последовательностью целых чисел в диапазоне 0<=X<256. Аргументы конструктора интерпретируются как для bytearray().

**complex([real[, imag]])** - преобразование к комплексному числу.

**dict([object])** - преобразование к [словарю](http://pythonworld.ru/tipy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html).

**float([X])** - преобразование к [числу с плавающей точкой](http://pythonworld.ru/tipy-dannyx-v-python/chisla.html). Если аргумент не указан, возвращается 0,0.

**frozenset([последовательность])** - возвращает неизменяемое множество.

**int([object], [основание системы счисления])** - преобразование к целому числу.

**list([object])** - создает [список](http://pythonworld.ru/tipy-dannyx-v-python/spiski-list-funkcii-i-metody-spiskov.html).

**memoryview([object])** - создает объект memoryview.

**object()** - возвращает безликий объект, являющийся базовым для всех объектов.

**range([start=0], stop, [step=1])** - арифметическая прогрессия от start до stop с шагом step.

**set([object])** - создает [множество](http://pythonworld.ru/tipy-dannyx-v-python/mnozhestva-set-i-frozenset.html).

**slice([start=0], stop, [step=1])** - объект среза от start до stop с шагом step.

**str([object], [кодировка], [ошибки])** - строковое представление объекта. Использует метод \_\_str\_\_.

**tuple(obj)** - преобразование к [кортежу](http://pythonworld.ru/tipy-dannyx-v-python/kortezhi-tuple.html).

**Встроенные функции, использующие методы класса (логика может отличаться в зависимости от типа данных)**

|  |  |  |
| --- | --- | --- |
| **Функция** | **Используемый метод класса** | **Основное предназначение** |
| **abs(X)** | \_\_abs\_\_ | Модуль (абсолютное значение) X. |
| **len(X)** | \_\_len\_\_ | Число элементов объекта |
| **hash(object)** | \_\_hash\_\_ | Хэш-сумма (если есть) |
| **iter(object)** | \_\_iter\_\_ | Возвращает объект-итератор |
| **next(iterator)** | \_\_next\_\_ | Следующий элемент итератора |
| **format(value[,format\_spec])** | \_\_format\_\_ | Форматирование (обычно строки) |
| **pow(x, y, [r])** | \_\_pow\_\_ | ( x в степени y ) % r |
| **reversed( object )** | \_\_reversed\_\_ | Итератор из развернутого объекта |
| **repr(obj)** | \_\_repr\_\_ | Представление объекта |

**Встроенные функции, использующиеся в качестве** [**декораторов**](http://pythonworld.ru/osnovy/dekoratory.html)

**classmethod(function)** - возвращает метод класса для функции.

**property(fget=None, fset=None, fdel=None, doc=None)**

**staticmethod(function)** - статический метод для функции.

**Встроенные функции, для перевода между системами счисления**

**bin(x)** - преобразование целого числа в двоичную строку.

**hex(х)** - преобразование целого числа в шестнадцатеричную строку.

**oct(х)** - преобразование целого числа в строку восьмеричное.

**Другие встроенные функции**

**all(последовательность)** - Возвращает True, если все элементы истинные (или, если последовательность пуста).

**any(последовательность)** - Возвращает True, если хотя бы один элемент - истина. Для пустой последовательности возвращает False.

**ascii(объект)** - Как repr(), возвращает строку, содержащую представление объекта, но заменяет не-ASCII символы на экранированные последовательности.

**chr(X)** - Возвращает односимвольную строку, код символа которой равен X.

**compile(source, filename, mode, flags=0, dont\_inherit=False)** - Компиляция в программный код, который впоследствии может выполниться функцией eval или exec. Строка не должна содержать символов возврата каретки или нулевые байты.

**delattr(объект, имя)** - Удаляет имя атрибута из объекта.

**dir([Объект])** - Список имен объекта, а если объект не указан, список имен в текущей локальной области видимости.

**divmod(a, b)** - Возвращает частное и остаток от деления a на b.

**enumerate(iterable, start=0)** - Возвращает итератор, при каждом проходе предоставляющем кортеж из номера и соответствующего члена последовательности.

**eval(expression, globals=None, locals=None)** - Выполняет строку программного кода.

**exec(object[, globals[, locals]])** - Выполняет программный код на Python (много, много программного кода :))

**filter(function, iterable)** - Возвращает итератор из тех элементов, для которых function возвращает истину.

**getattr(объект, имя ,[по умолчанию])** - извлекает атрибут объекта или по умолчанию.

**globals()** - Словарь глобальных имен.

**hasattr(object, name)** - Имеет ли объект атрибут с именем 'name'.

**help([object])** - Вызов встроенной справочной системы.

**id(объект)** - Возвращает "адрес" объекта. Это целое число, которое гарантированно будет уникальным и постоянным для данного объекта в течение срока его службы.

**input([prompt])** - Возвращает введенную пользователем строку. Prompt - подсказка пользователю.

**isinstance(объект, ClassInfo)** - Истина, если объект является экземпляром ClassInfo или его подклассом. Если объект не является объектом данного типа, функция всегда возвращает ложь.

**issubclass(класс, ClassInfo)** - Истина, если класс является подклассом ClassInfo. Класс считается подклассом себя.

**locals()** - словарь локальных имен.

**map(function, iterator)** - итератор, получившийся после применения к каждому элементу последовательности функции function.

**max(iter, [args ...] \* [, key])** - максимальный элемент последовательности.

**min(iter, [args ...] \* [, key])** - минимальный элемент последовательности.

**open(file, mode='r', buffering=None, encoding=None, errors=None, newline=None, closefd=True)** - открывает файл и возвращает соответствующий поток. Если файл не может быть открыт, IOError поднимается.

**ord(с)** - код символа.

**print([объект, ...], \*, sep = " ", end = '\n', file = sys.stdout)** - печать

**round(X [, N])** - округление до N знаков после запятой.

**setattr(объект, имя, значение)** - устанавливает атрибут объекта.

**sorted(iterable[, key][, reverse])** - отсортированный список.

**sum(iter, start=0)** - сумма членов последовательности. Для более точного значения суммы последовательности из чисел с плавающей точкой используйте math.fsum() из модуля [math](http://pythonworld.ru/moduli/modul-math.html).

**super([тип [, объект или типа]])** - доступ к родительскому классу.

**type(object)** - тип объекта.

**type(name, bases, dict)** - возвращает новый экземпляр класса name.

**vars([object])** - словарь из атрибутов объекта. По умолчанию - словарь локальных имен.

**zip(\*iters)** - итератор, возвращающий кортежи, состоящие из соответствующих элементов аргументов-последовательностей.