

# Переменные и константы

**Идентификатор**—это имя объекта. Он может состоять из больших и маленьких букв ,цифр,знаков подчёркивания, не должен начинаться с цифры и не может совпадать с зарезервированными ключевыми словами языка .Регистр имеет значение.

**Переменная** в Python—ссылка на какой-либо объект.

Согласно PEP8, имена переменных должны записываться маленькими буквами через знаки подчёркивания: `my_variable`.

*Константа* — значение, которое не предназначено для изменения.

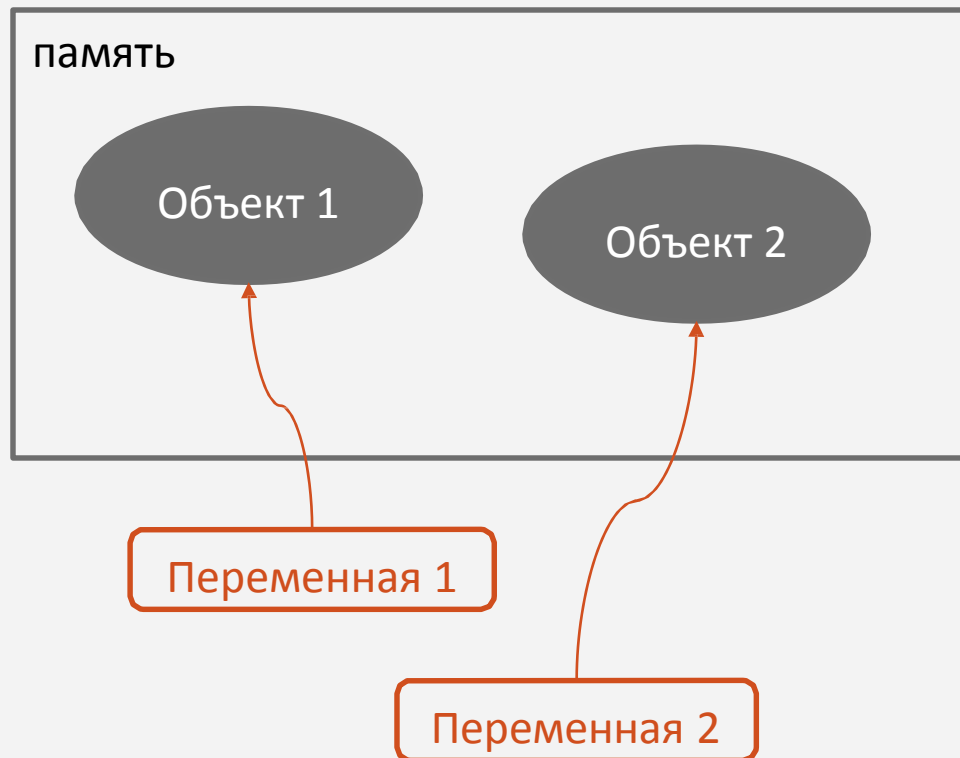
В Python нет отдельной синтаксической конструкции для объявления констант. Принято называть их идентификаторами, написанными заглавными буквами: `MY_CONSTANT`.

# Понятие типов данных

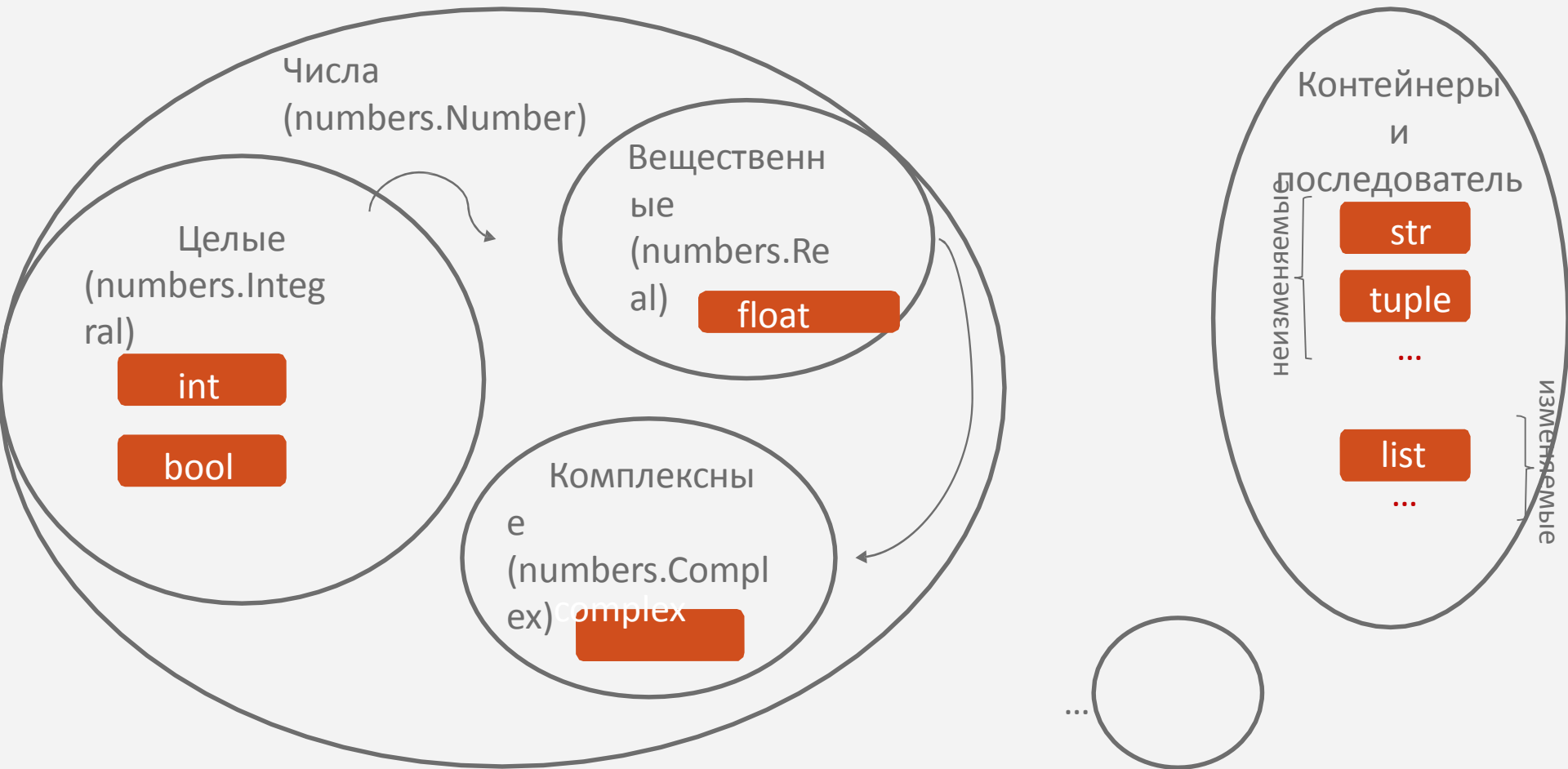
*Тип данных*(тип)—множество значений и операций на этих значениях.

Тип определяет возможные значения и их смысл, операции , а так же способы хранения значений типа. Изучается теорией типов. Неотъемлемой частью большинства языков программирования являются системы типов, использующие типы для обеспечения той или иной степени безопасности.

Операция назначения типа информационным сущностям называется *типизацией*.



# Основные стандартные типы дан



# Операции с числами

---

$x + y$	сумма
$x - y$	разница
$x * y$	произведение
$x / y$	частное
$x // y$	операция целочисленного деления
$x \% y$	остаток от деления
$-x$	число, противоположное $x$
$+x$	$x$

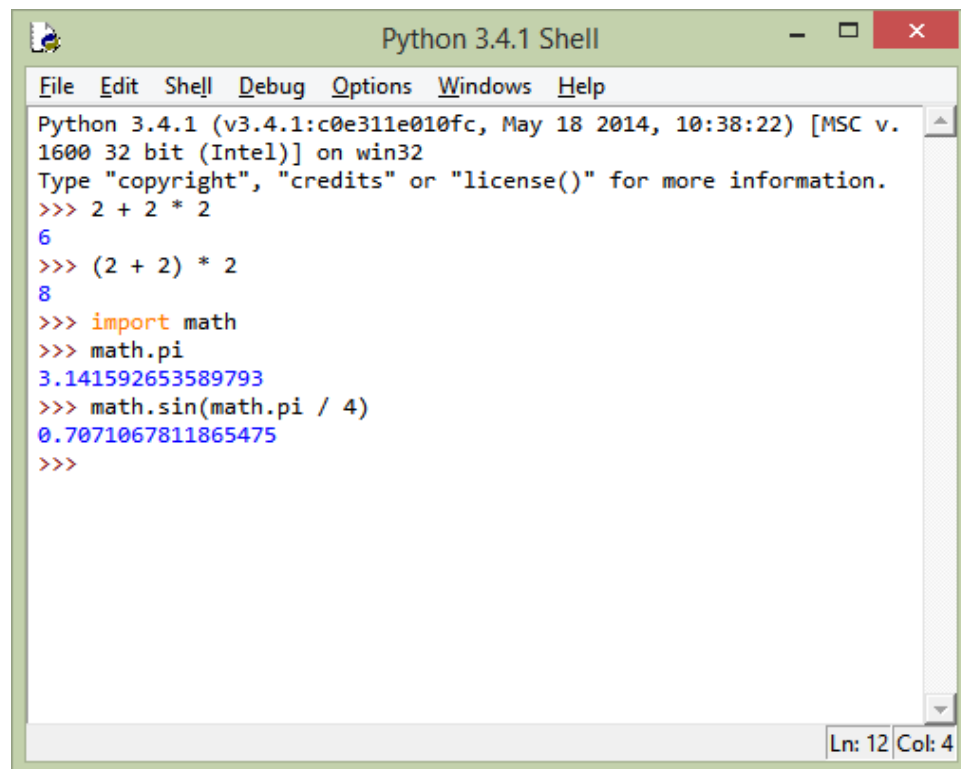
# Операции с числами

---

<code>abs(x)</code>	модуль числа $x$
<code>int(x)</code>	преобразовать $x$ в целое число
<code>float(x)</code>	преобразовать $x$ в комплексное число
<code>complex(re, im)</code>	создать комплексное число $re + im*i$
<code>c.conjugate()</code>	число, сопряжённое комплексному числу $c$
<code>x ** y</code> <code>pow(x, y)</code>	$x$ в степени $y$
<code>round(x)</code> <code>round(x, n)</code>	округлить действительное число $x$ (до $n$ цифр после запятой, если $n$ указано)

# Операции с числами

- У арифметических операций есть приоритеты (такие же, как и в математике). Чтобы изменить порядок вычислений, следует использовать круглые скобки.
- Можно импортировать модуль `math`, написав `import math`, и использовать его функции и константы.



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.
1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 2 + 2 * 2
6
>>> (2 + 2) * 2
8
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(math.pi / 4)
0.7071067811865475
>>>
```

Ln: 12 Col: 4

# Логические операции и сравнения

- Логическая операция — операция над выражениями логического (булевского) типа, соответствующая некоторой операции над высказываниями в алгебре логики. Как и высказывания, логические выражения могут принимать одно из двух истинностных значений — «истинно» или «ложно».
- Логические операции служат для получения сложных логических выражений из более простых. В свою очередь, логические выражения обычно используются как условия для управления последовательностью выполнения программы.



# Таблицы истинности

x or y	если x – ложь, то y, иначе x
x and y	если x – ложь, то x, иначе y
not x	если x – ложь, то True, иначе False

# Операции сравнения

$x < y$	x строго меньше y
$x > y$	x строго больше y
$x \leq y$	x меньше или равен y
$x \geq y$	x больше или равен y
$x == y$	x равен y
$x != y$	x не равен y
$x \text{ is } y$	x и y – это один и тот же объект
$x \text{ is not } y$	x и y не являются одним и тем же объектом в памяти

С операциями  $<$ ,  $>$ ,  $\leq$ ,  $\geq$  можно использовать двойные сравнения, например,  $-2 \leq x < 3$ .

# Работа со строками

- Строки – текстовые данные типа `str`.
- Строковые литералы (значения, которые задаются в коде) обрамляются с обеих сторон одинарными или двойными кавычками.
- Несколько строковых литералов подряд распознаются как одна строка.
- Строки, которые выделяются одной парой кавычек, должны располагаться на одной строке кода. Если необходимо создать строку из нескольких строчек текста, можно:
  - использовать специальную последовательность символов `\n` ;
  - расположить несколько строковых литералов на разных строках кода, не добавляя между ними никаких операций;
  - использовать специальный вид строковых литералов, который обрамляется тремя парами одинарных или двойных кавычек. Любая часть исходного текста программы, располагающаяся между ними, включая переводы строк и отступы, считается строкой.

## Некоторые операции со строками

<code>s1 + s2</code>	конкатенация (объединение) строк
<code>s % x</code> <code>s % (x<sub>1</sub>, x<sub>2</sub>, ..., x<sub>n</sub>)</code>	форматирование строки в стиле C s – форматная строка, x <sub>1</sub> ...x <sub>n</sub> – значения <a href="https://docs.python.org/3/library/stdtypes.html#printf-style-string-formatting">https://docs.python.org/3/library/stdtypes.html#printf-style-string-formatting</a>
<code>s.format(args)</code>	форматирование строки в стиле C# <a href="https://docs.python.org/3/library/stdtypes.html#str.format">https://docs.python.org/3/library/stdtypes.html#str.format</a>
<code>s[i]</code>	символ, который стоит в строке s на позиции i (нумерация начинается с нуля)

## Вывод

Для вывода значений на экран служит функция print:

```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

где

- objects – это объекты, которые необходимо вывести
- sep – разделитель
- end – строка, которую необходимо вывести после всех объектов
- file – файл, в который необходимо вывести данные
- flush – нужно ли сразу после вывода сбросить содержимое буфера в файл. Если вы выводите информацию на одной строке через продолжительные промежутки времени, и она не появляется на экране, пока вы не выведете символ новой строки, добавьте параметр flush=True

Ни один из этих параметров не является обязательным.

- Для ввода данных с клавиатуры можно использовать функцию `input`:
  - `input(prompt)`
  - `input()`
- Если параметр `prompt` задан, то он выводится, как поясняющий текст, приглашение ко вводу.
- Функция приостанавливает выполнение программы, пока пользователь не введёт строку
- Обратите внимание, что она возвращает именно строку, то есть значение типа `str`, поэтому если необходимо ввести число, то нужно воспользоваться одной из функций, которые были рассмотрены выше, чтобы сконструировать число из его текстового представления.