

Базовые типы

integer, float, boolean, string

```
int 783 0 -192
float 9.23 0.0 -1.7e-6
bool True False
str "One\nTwo" 'I\'m'
```

↑
неизменяемая,
упорядоченная
последовательность символов

новая строка 'Экранирование
многострочный' `"""X\tY\tZ
1\t2\t3"""`
табуляция

Типы контейнеров

■ упорядоченная последовательность, быстрый доступ по индексу, повторяемые значения

```
list [1,5,9] ["x",11,8.9] ["word"] []
tuple (1,5,9) 11,"y",7.4 ("word",) ()
```

список (как массив)
кортеж
неизменяемый

выражение только с запятыми
str как упорядоченная последовательность символов

Неупорядоченные последовательности, уникальный ключ, ; keys = базовые типы или кортежи

```
dict {"key": "value"} {}
{1: "one", 3: "three", 2: "two", 3.14: "п"}
```

словарь
Ключ/значение

```
set {"key1", "key2"} {1,9,3,0} set()
```

множества

Идентификаторы

для переменных, функций, модулей, классы ... имена

a..zA..Z_ с последующими **0..9**

- диакритические знаки допускаются, но следует избегать
- ключевые слова языка запрещены
- нижний / верхний регистр различаются

⊗ **a toto x7 y_max BigOne**
⊗ **8y-and**

присваивание переменных

```
x = 1.2+8+sin(0)
```

↑
значение или вычисляемое выражение

Имя переменной (идентификатор)

```
y,z,r = 9.2,-7.6,"bad"
```

имена
переменных

контейнер с несколькими значениями (здесь кортеж)

инкремент (приращение) **x+=3**
decrement **x-=2**

x=None « неопределенное значение »

Преобразования

int ("15") Можно указать целое число base in 2nd parameter

int (15.56) усечение дробной части (**round(15.56)** округляет до int)

float ("-11.24e8")

str (78.3) представление для литерала → **repr ("Text")**
см. другую сторону для форматирования строк

bool → Использование компараторов (with ==, !=, <, >, ...), boolean результат

list ("abc") использовать каждый элемент из последовательности → ['a', 'b', 'c']

dict ((3, "three"), (1, "one")) → {1: 'one', 3: 'three'}

set (["one", "two"]) использовать каждый элемент из последовательности → {'one', 'two'}

":".join(['toto', '12', 'pswd']) → 'toto:12:pswd'
присоединение строки последовательность строк

"words with spaces".split() → ['words', 'with', 'spaces']

"1,4,8,2".split(",") → ['1', '4', '8', '2']
расщепление строки

отрицательный индекс

положительный индекс	-6	-5	-4	-3	-2	-1
	0	1	2	3	4	5
	11	67	"abc"	3.14	42	1968
положит. срез	0	1	2	3	4	5
отрицат. срез	-6	-5	-4	-3	-2	-1

```
lst = [11, 67, "abc", 3.14, 42, 1968]
```

lst[: -1] → [11, 67, "abc", 3.14, 42]
lst[1: -1] → [67, "abc", 3.14, 42]
lst[: : 2] → [11, "abc", 42]
lst[:] → [11, 67, "abc", 3.14, 42, 1968]

Для списков, кортежей, строк, ... Индексация

len (lst) → 6
последовательностей

индивидуальный доступа к пунктам с помощью [index]

lst [1] → 67 **lst [0]** → 11 первый
lst [-2] → 42 **lst [-1]** → 1968 последний

доступ к суб-последовательности [start slice : end slice : step]

lst [1:3] → [67, "abc"]
lst [-3: -1] → [3.14, 42]
lst [:3] → [11, 67, "abc"]
lst [4:] → [42, 1968]

Недостающий кусочек индикация → от начала / до конца.

На изменяемых последовательностях, используем для удаления **del lst [3:5]** а изменить с присвоением так: **lst [1:4] = ['hop', 9]**

Булева логика

Компараторы: < > <= >= == !=

a and b логическое и
both simultaneously

a or b логическое или
one or other or both

not a логическое не

True выражение истинно

False выражение ложно

Отступы

```
parent statement:
statements block 1...
:
parent statement:
statements block 2...
:
next statement after block 1
```

Условный оператор

инструкция блока выполняется, только если условие истинно

if логическое выражение:
→ блок операторов

может включать несколько elif, и только один финальный else

пример:

```
if x==42:
    # если логическое выражение x == 42 верно
    print("real truth")
elif x>0:
    # если логическое выражение x> 0 верно
    print("be positive")
elif bFinished:
    # если логическая переменная bFinished = истина
    print("how, finished")
else:
    # блок для других случаев
    print("when it's not")
```

числа с плавающей точкой ... приближенные значения!

операторы: + - * / // % **

↑
целое ÷ ÷ остаток

(1+5.3)*2 → 12.6

abs (-3.2) → 3.2

round (3.57, 1) → 3.6

математика

углы в радианах

```
from math import sin, pi...
sin(pi/4) → 0.707...
cos(2*pi/3) → -0.4999...
acos(0.5) → 1.0471...
sqrt(81) → 9.0
log(e**2) → 2.0 etc. (cf doc)
```

