## Python 3 Шпаргалка (ru)

```
integer, float, boolean, string Базовые типы
                                                                                                    Типы контейнеров
                                                ■ упорядоченная последовательность, быстрый доступ по индексу, повторяемые значения
                          -192
    int 783
                                                                            ["x", 11, 8.9]
                                                    list [1,5,9]
                                                                                                   ["word"]
                                                         массив)
e (1,5,9)
                                               список (как мас

tuple

кортеж
float 9.23
                    0.0
                                                                             11, "y", 7.4
                                                                                                   ("word",)
                              -1.7e-6
                                                                                                                     ()
                                                                    выражение только с запятыми
                                     10^{-6}
                                                 неизменяемый
  bool True
                    False
                                                     >>str как упорядоченная последовательность символов
                               'I\'m'
    str
         "One\nTwo"
                                                Неупорядоченные последовательности, уникальный ключ, ; keys = базовые типы или кортежи
               новая строка Ужранировани
                                                    dict {"key":"value"}
           многострочный"""X\tY\tZ
                                                             {1: "one", 3: "three", 2: "two", 3.14: "π"}
                                                Ключ/значение
неизменяемая.
                            t2\t,3"""
                                                множества
<u>VПОРЯДОЧЕННАЯ</u>
                                                            {"key1", "key2"}
                                                                                          {1,9,3,0}
                              табуляция
                                                                                                               set()
послеловательность символов
для переменных, функций, Идентификаторы
                                                                                     type (выражение) Преобразования
модули, классы ... имена
                                                            Можно указать целое число base in 2<sup>nd</sup> parameter
                                          int ("15")
а.. z A.. Z _ с последующими 0..9
                                          int (15.56) усечение дробной части(round (15.56) округляет до int)
□ диакритические знаки допускаются, но
следует избегать
                                          float ("-11.24e8")
□ ключевые слова языка запрещены
                                          str (78.3)
                                                                                                   → repr("Text")
                                                          представление для литерала -
\hfill\Boxнижний / верхний регистр различаются
                                                     см. другую сторону для форматирования строк
  © a toto x7 y_max BigOne
  ⊗ 8<del>y</del>=a<del>nd</del>
                                          bool \longrightarrow Использование компараторов (with ==, !=, <, >, ...), boolean результат
                                                                   использовать каждый элемент
          присваивание переменных
                                          list("abc")_
                                                                                                 →['a','b','c']
                                                                    из пооследовательности
    = 1.2 + 8 + \sin(0)
                                          dict([(3, "three"), (1, "one")])
                                                                                             > {1:'one',3:'three'}
      значение или вычисляемое выражение
                                                                     использовать каждый элемент
                                          set (["one", "two"]) ______из пооследовательности
                                                                                                    → {'one','two'}
 Имя переменной (идентификатор)
 y,z,r = 9.2,-7.6, "bad"
                                          ":".join(['toto','12','pswd'])—
                                                                                              → 'toto:12:pswd'
               контейнер с несколькими
  имена
                                         присоединение строки
                                                                 последовательность строк
              значениями (здесь кортеж)
                                          "words with spaces".split()—→['words','with','spaces']
           инкремент(приращение)
               decrement \longrightarrow x-=2
                                          "1,4,8,2".split(<u>","</u>)-
 x=None « неопределенное значение »
                                                            расщепление строки
                                                                   Для списков, кортежей, строк, ... Индексации
отрицательный индекс
                                                                                                  последовательгостей
                                                   -2
                                                           -1
                                                                     len(lst) \longrightarrow 6
положительный индекс
                        1
                                 2
                                            3
                                                    4
                                                            5
                                                                  индивидуальный доступа к пунктам с помощью [index]
                       67,
                              "abc"
                                                   42,
                                         3.14,
                                                          1968]
                                                                     lst[1] \rightarrow 67
                                                                                               1st[0] → 11 первый
положит. срез
                                                                         lst[-2] \rightarrow 42
                                                                                               lst[-1] →1968 последний
                                      -¦3
                                                -2
                                                      -1
отрицат. срез
                  -5
                          -4
                                                                    доступ к суб-последовательности [start slice: end slice: step]
        lst[:-1] \rightarrow [11, 67, "abc", 3.14, 42]
                                                                     lst[1:3] \rightarrow [67, "abc"]
        lst[1:-1] \rightarrow [67, "abc", 3.14, 42]
                                                                     lst[-3:-1] \rightarrow [3.14,42]
        lst[::2] \rightarrow [11, "abc", 42]
                                                                     lst[:3] → [11,67, "abc"]
        lst[:]→[11,67, "abc", 3.14,42,1968]
                                                                     lst[4:] \rightarrow [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]
                                                                      инструкция блока выполняется, Условный оператор
               Булева логика
                                                      Отступы
Компараторы: < > <= >= !=
                                                                        только если условие истинно
                                   parent statement
                                      statements block 1...
                                                                                 if логическое выражение:
 a and b логическое и
                                                                                     → блок операторов
           both simultaneously
             логическое ил
                                      parent statement:
                                                                     может включать несколько elif, и только один финальный else
            one or other or bo
                                         statements block 2...
                                                                     пример:
 not
             логическое не
 True
           выраженеие истинно
                                                                     if x==42:
                                                                          # если логическое выражение x == 42 верно
 False выражение ложно
                                   next statement after block 1
                                                                          print("real truth")
                                                                     elif x>0:
🙎 числа с плавающей точкой ... приближенные значения!
                                                       математика
                                                                          # если логическое выражение x > 0 верно
                                           углы в радианах
                                                                          print("be positive")
 операторы: + - * / // % **
                                 from math import sin, pi...
                                                                     elif bFinished:
                         a
                                 \sin(pi/4) \to 0.707...
                        ÷ остаток
                                                                          # если логическая переменная bFinished =истина
                                 \cos(2*pi/3) \rightarrow -0.4999...
                                                                          print("how, finished")
 (1+5.3)*2\rightarrow12.6
                                 acos (0.5) →1.0471...
                                                                     else:
 abs (-3.2) \rightarrow 3.2
                                                                     # блок для других случаев
print("when it's not")
                                 sqrt(81) \rightarrow 9.0
                                 log(e**2) \rightarrow 2.0 etc. (cf doc)
 round (3.57, 1) \rightarrow 3.6
```

```
Условный цикл while
                                                             🐧 🕻 инструкции блока выполняются
 инструкции блока выполняются
                                                                                                   Итерационный цикла
                                                               для каждого элемента контейнера или итератора
 пока условие истинно
            while логическое выражение:
                                                                         for переменная in последовательность:
                    блок инструкций
                                                  Инструкции выхода
                                                                                     📕 блок инструкций
                                                      найти значение в последовательности немедленный выход
      1 } инициализации перед циклом
                                                  break
                                                                         s = "Some text"
                                                                                                -инициализации перед циклом
 состояние, по крайней мере, одного значения переменной
                                                                         cnt = 0
                                                    следующая итерация
 while i <= 100:
                                                                            Переменная иикла, значение управляется инструкцией for
      # инструкция выполняться до тех пор, пока i \le 100
                                                                          for cin s:
                                                     i = 100
                                                                                                         Подсчитать
      s = s + i**2
                                                                                                         количество е в строке
      і = і + 1 } 🖁 измененить состояния переменной
                                                                                     cnt = cnt + 1
                                                                          print("found", cnt, "'e'")
 print ("sum:", s) } вычисленный результат после цикла
         🖠 будьте осторожны, избегайте бесконечных циклов!
                                                                 в деталях index последовательностей
                                            Ввод /Вывод
                                                                 □ изменить пункт по индексу
                                                                 lst = [11, 18, 9, 12, 23, 4, 17]
                                                                 lost = []
                                                                                                       если предельные значения
 элементы отображения: литералы, переменные, выражения
                                                                 for idx in range(len(lst)):
                                                                                                      больше, чем 15,
 print опции:
                                                                      val = lst[idx]
                                                                                                       запоминим утраченные
   □ sep=" " (items separator, default space)
                                                                       if val > 15:
                                                                                                       значений.
    □ end="\n" (end of print, default new line)
                                                                            lost.append(val)
    □ file=f (print to file, default standard output)
                                                                            lst[idx] = 15
 s = input("Instructions:")
                                                                 print("modif:",lst,"-lost:",lost)
    🖞 input всегда возвращает string
                                                                 одновременнно index and values:
                                                                 for idx, val in enumerate(lst):
                                                                    часто используется Генератор int последовательностей
len (c) → подсчет элементов
                                    Операции на контейнерах
                                    заметка: Для словарей и множеств,
                                                                                    по умолчанию 0
min(c)
          max(c)
                                    эти операции используют keys.
sorted(c) \rightarrow coртировка
                                                                                     range ([start,]stop [,step])
val in c → boolean, membersihp operator in (absence not in)
                                                                                                          0 1 2 3 4
                                                                    range (5)
enumerate (c) → итератор (index, value)
                                                                                                                  5
                                                                                                                     6 7
                                                                    range (3,8)
Специально для последовательностуй (lists, tuples, strings):
                                                                    range (2, 12, 3)
reversed (c) \rightarrow обратн. umepam. c*5 \rightarrow дубликатc+c2 \rightarrow конкатенация
c.index (val) → позиция
                             c.count (val) → подсчет событий
                                                                        range возвращает «генератор», преобразует его в
                                                                        список, чтобы увидеть значения, например:
                                     Операции со списками
🙎 изменить первоначальный список
                                                                        print(list(range(4)))
lst.append(item)
                           добавить пункт в конце
lst.extend(seq)
                           добавить последовательность элемент. в конце
                                                                                                  определение функций
                                                                    Имя функции (идентификатор)
!lst.insert(idx, val) вставить элемент с индексом
                                                                                         именованные параметры
                              удалить первый элемент со значением
lst.remove(val)
                    Удалить элемент по индексу, возвращает его значение
lst.pop(idx)
                                                                    def fctname(p_x,p_y,p_z):
                lst.reverse() сортировка/ реверс список не меняет
                                                                              """документация"""
                                                                           # блок инструкций,расчеты, и т.п
  Операции со словарями
                                      Операции с множествами
                                                                           return res — Результат вызова.
                                   Операторы:
d[key] = value
                   d.clear()
                                  I → объединение
d[key] \rightarrow value
                                                                                                 если нет вычисленного
                   del d[clé]
                                                                    🛮 параметры и всё, что в блоке,
                                  & → пересечение
                                                                    существуют только в блоке и во
d. update (d2) {обновить /добав.
                                  – ^ разность/ симметричн. дифф.
                                                                                                 результата, вернет:
                                                                    время вызова функции
                 ассоциации
                                   <<=>>=→ отношения включения
d.keys()
                                                                                                 return None
                                                                    ("черный ящик")
d.values() ключи, значения
                                  s.update(s2)
                                                                                                            Function call
                                                                       = fctname(3,i+2,2*i)
d.items() | accoquaquu
                                 s.add(key) s.remove(key)
                                                                                 один аргумент за параметр
d.pop(clé)
                                  s.discard(key)
                                                                     получить возвращаемый результат (в случаенеобходимости)
                                                        Файлы
 хранение данных на диске, и чтение обратно
                                                                                                 Форматирование строк
f = open("fil.txt", "w", encoding="utf8")
                                                                    директивы форматирования Значения для форматирования
 файловая
                                                                     "model {} {} {}".format(x,y,r) =
             Название файла
                            режим открытия
                                                 кодировка
 переменная
                                                                    " { выбор : форматирование ! преобразование } "
                                                 пля текстового
             на диске
                            □ 'r' читать
 для операций
                                                                                         "{:+2.3f}".format(45.7273)
                            □ 'w' запись
                                                 файла:
                                                                     выбор:
             (+путь...)
                                 ' добавлять...
                                                                                          →'+45.727'
                                                 11t f 8
                                                        ascii
 cf функции в модулях os и os.path
                                                 latin1
                                                                                         "{1:>10s}".format(8, "toto")
                                                                      0.nom
                                                                                                   toto'
                              пуст. строка, если конец файла
                                                          чтение
                                                                      4[key]
0[2]
                                                                                         "{!r}".format("I'm")
 f.write("hello")
                              \dot{s} = f.read(4) Если символ
                                                                                         →'"I\'m"'
                                                                    форматирование :
                                   читает построчно
                                                  не указан,
 💆 текст. файдол → чтения / записи
                                                                    fillchar alignment sign minwidth.precision~maxwidth type
 только строки, преобразовывать
                                                   читать весь файл
 из / в требуемого типа.
                              s = f.readline()
                                                                            + - space
                                                                                        0 at start for filling with 0
 f.close() 🛮 не забудьте закрыть файл после использования
                                                                    integer: b binary, c char, d decimal (default), o octal, x or X hexa..
        Python автоматическое закрытие: with open (...) as f:
                                                                    float: e or E exponential, f or F fixed point, g or G appropriate (default),
 Очень часто: итерационное чтение в цикле текстового файла
 for line in f :
                                                                    string: s ...
     † # блок обработки строк
                                                                    □Преобразования: s (читаемый текст) или r (буквен. представление).
```