

# Повторение Python

## Установка

1. Установить Python с официального сайта <https://www.python.org/>

1.1) Не забудьте указать галочку (add to PATH)

1. Установить редактор кода <https://www.sublimetext.com/>

2. Открыть командную строку и проверить работу python командой python

Ссылка на видео туториал: <https://www.youtube.com/watch?v=eFANOmPjjWE>

## Переменные

```
# объявление переменных
a = 12
b = "привет"
c = True

# вывод данных
print(b)

# чтение значения из консоли
d = input("Введите значение: ")
```

## Формирование строк

```
a = "привет"
b = "друг"

# конкатенация
print(a+" "+b)

# интерполяция
print(f"{a} {b}")

# метод format
print("{} {}".format(a, b))
```

## Преобразование в int

```
# Умножение строки на число
val1 = input('Введите число: ')
print(val1*12)

# Умножение числа на число
val1 = input('Введите число: ')
val1_int = int(val1)
print(val1*12)
```

## Преобразование в str

```
# Конкатенация строки со строкой
val1 = input('Введите число: ')
val1_int = int(val1)

result = str(val1*12) # перевод результата в строку
print('Результат: '+result)
```

## Преобразование во float

```
a = '3.14'
# ошибка!
a_01 = int(a)

# перевод в дробное число
a_02 = float(a)

# отсечение дробной части
a_03 = int(a_02)
```

## Задание

1) Напишите программу, которая считывает длины двух катетов в прямоугольном треугольнике и выводит его площадь. Каждое число записано в отдельной строке.

## Условный оператор

```
a = 5
b = 6

if a > b:
    print('a>b')
elif a < b:
    print('a<b')
else:
    print('a=b')
```

## Задание

В математике функция **sign(x)** (знак числа) определена так:

```
sign(x) = 1, если x > 0,
sign(x) = -1, если x < 0,
sign(x) = 0, если x = 0.
```

Для данного числа  $x$  выведите значение `sign(x)`. Эту задачу желательно решить с использованием каскадных инструкций `if... elif... else`.

## Списки

```
lst = [12, 345, 23, 54, 457, 7]

print(lst[2]) # 23 (вывод по индексу)
lst[2] = -1 # изменение значения
lst.append(15) # добавление значения
len(lst) # получить длину списка
```

Необходимо различать список и итерируемые объекты. Список это определенный тип данных, итерируемый объект, это множество типов данных, по которым можно пройти в цикле

## Цикл for

```
# цикл по элементам
for elem in lst:
    result = elem**2
    print(result)

# цикл по индексам
for i in range(len(lst)):
    result = lst[i]**2
    print(result)
```

Функция range возвращает итерируемый объект с индексами.

## Сдвиги по индексу

```
for i in range(1, len(lst)-1):
    result = f'{lst[i-1]} {lst[i]} {lst[i+1]}'
    print(result)
```

При сдвиге по индексу необходимо исправлять диапазон в range, это позволит не выйти за пределы списка.

## split и join

```
string = '12 43 2 74 23 45'

# преобразование строки в список
lst = string.split(' ') # ['12', '43', '2', '74', '23', '45']

# преобразование списка в строку
string = ','.join(lst) # '12,43,2,74,23,45'
```

Для преобразования списка в строку все элементы должны быть строчными

## Задание

- 1) Выведите все четные элементы списка. При этом используйте цикл `for`, перебирающий элементы списка, а не их индексы!
- 2) Дан список чисел. Если в нем есть два соседних элемента одного знака, выведите эти числа. Если соседних элементов одного знака нет — не выводите ничего. Если таких пар соседей несколько — выведите первую пару.

## Генератор списков

```
string = '12 43 2 74 23 45'

lst = string.split(' ') # ['12', '43', '2', '74', '23', '45']

# преобразование значений через цикл
for i in range(len(lst)):
    lst[i] = int(lst[i])

# преобразование значений через генератор списков
lst = [int(elem) for elem in lst]
```

## Цикл while

```
# вывод в консоль чисел от 0 до 9
i = 0
while i < 10:
    print(i)
    i += 1

# обработка значений до ввода слова end
while True:
    value = input('Введите значение: ')
    if value == 'end':
        break
    print(value)
```

## Словари

```
user = {
    "name": 'Гайк',
    "lastname": 'Инанц',
    "age": 27
}

print(user['name']) # обращение к значению через ключ
user['age'] = 28 # переопределение значения
user['gender'] = 'М' # добавление нового ключа и значения
```

## Задание

В строке записан текст. Для каждого слова из текста подсчитайте, сколько раз оно встречалось в этом тексте ранее.

## Функции

Возвращение результата

```
# функция ничего не возвращает
a = print('привет')

# функция возвращает число 12
a = int('12')
```

## Создание функции

```
# объявление функции
def power(v):
    result = v**2
    return result

# вызов функции
a = power(12) # a = 144
b = power(15) # b = 225
```

## Особенности return

```
# после слова return процесс выполнения функции
# прекращается
def power(v):
    result = v**2
    return result
print('ты не увидишь эту строку')
```

## Задание

Напишите функцию, которая получает список и возвращает словарь со значениями max, min, avg списка.

## Функция как аргумент

```
lst = [{'v':5}, {'v':1}, {'v':3}, {'v':12}, {'v':15}]

def forSort(x):
    return x['v']

# функция forSort передается как аргумент в функцию sorted
result = sorted(lst, key=forSort)
print(result)
```

## lambda

```
power = lambda v, p: v**p
result = power(12, 2)
print(result)
```

## Передача lambda

```
lst = [{'v':5}, {'v':1}, {'v':3}, {'v':12}, {'v':15}]

result = sorted(lst, key=lambda x: x["v"])
```

```
print(result)
```

## try...except

```
value = input('Введите число: ')

try:
    # попытка выполнить код
    value = int(value)
except Exception as e:
    # процесс, который выполнится в ином случае
    print(str(e))
```

Exception - тип ошибки, которая будет обработана  
(тип ошибки можно посмотреть в консоле)

### Задание

Напишите функцию, которая получает строку и в случае, если эта строка преобразуема в число, то возвращает число, в ином случае возвращает None

## \*args

```
# *args позволяет получить все аргументы функции в виде кортежа
def func(*args):
    return args

result = func(23,4,35,5)
print(result) # (23, 4, 35, 5)
```

## \*\*kwargs

```
# kwargs содержит полученные именованные аргументы в виде словаря
def func(**kwargs):
```



```
return kwargs
```

```
result = func(a=12, b=43, c=2)  
print(result) # {'a': 12, 'b': 43, 'c': 2}
```