

Повторение Python

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}
```