

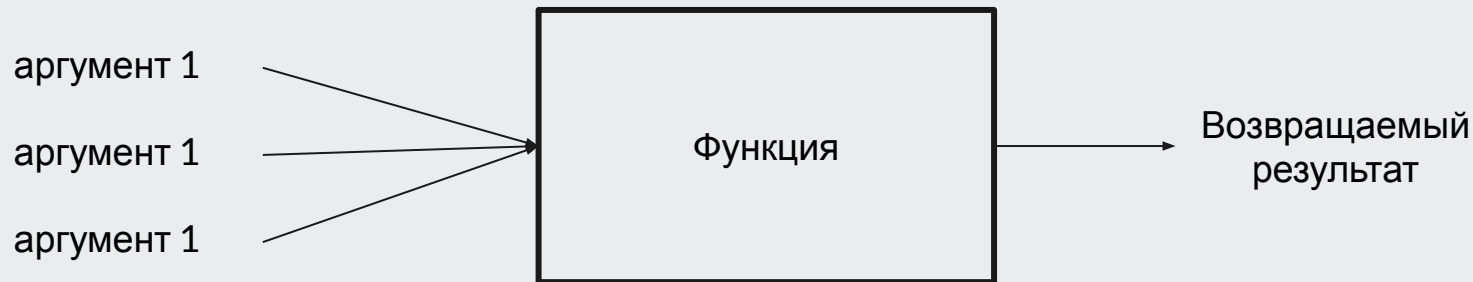


Python

Функции



Задача 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

- 1) Создать функцию, которая принимает аргумент и возвращает True, если число четное, и False если число нечетное.
- 2) Создать функцию, которая принимает два аргумента и возвращает наибольшее



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)
```



Задание

- 1) Написать функцию, которая получает функцию, как аргумент и вызывает ее с аргументами 1, 2
- 2) Написать функцию, которая получает число и возвращает список четных чисел от 2 до указанного числа



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)



Практическое задание

Написать функцию, которая получает значение и возвращает True, если это значение преобразуется в число и False в ином случае.



Задание

- 1) Написать функцию, которая получает два числа и знак, и возвращает результат вычисления
- 2) Функция должна проверять типы данных и оповещать пользователя о неверно указанном значении или о попытке произвести деление на 0



***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)
```



Задание

- 1) Написать функцию, которая получает аргументы и возвращает список только из чисел (отбросив нечисловые данные)



****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}
```




Задание

- 1) Написать функцию, которая получает именованные аргументы и возвращает имя аргумента с наибольшим значением



Практическое задание

Написать функционал по добавлению, изменению и удалению пользователей.
Данные должны храниться в списке, данные о пользователе должны быть в виде словаря (id, имя, фамилия, возраст)

Должны быть реализованы процессы добавления, изменения и удаления пользователей отдельными функциями.