

# Язык JavaScript (введение)

# Математическая функция

Функция одной переменной

$$y = f(x)$$

Функция нескольких переменных

$$y = f(x, y)$$

# Функция

```
// Реализация функции  
function add(x, y) {  
    var total = x + y;  
    return total;  
}
```

```
> add(2, 3, 4);
```

В чем несоответствие ?

# Присвоение значения из функции

// Реализация функции

```
function add(x, y) {  
    var total = x + y;  
    return total;  
}
```

```
var result = add(2, 3, 4);
```

// Какие типы можно

возвращать из функции ?

## Кол-во аргументов `arguments[ ]`

```
function add( ) {  
    var sum = 0;  
    var j = arguments.length;  
    for (var i = 0, i < j; i++)  
    {  
        sum += arguments[i];  
    }  
    return sum;  
}
```

```
> add(2, 3, 4, 5)
```

# Область видимости переменной

```
var sum = 0;
function add() {
    var sum = 0;
    var j = arguments.length;
    for (var i = 0; i < j; i++)
    {
        sum += arguments[i];
    }
    return sum;
}
add(2, 3);
>sum // Чему равна переменная ?
```

# Область видимости переменной

```
var sum = 0; ← глобальная  
                переменная
```

```
function add( ) {  
    var sum = 0; ← локальная  
                    переменная  
    var j = arguments.length;  
    for (var i = 0; i < j; i++) {  
        sum += arguments[i];  
    }  
    return sum;  
}
```

# Перегрузка функции

```
function addSomeNumber(num) {  
    return num + 100;  
}
```

```
function addSomeNumber(num) {  
    return num + 200;  
}
```

```
var result = addSomeNumber(100);  
// 300
```



В результате перегрузки иногда возникают конфликты старой и новой библиотеки. Т.к js подключается последовательно.

Что будет если вызов функции  
возникнет раньше ее реализации?

```
alert( sum( 10 , 10 ) );  
function sum( num1, num2 ) {  
    return num1 + num2;  
}
```

## Задачи.

1. Написать функцию подсчета суммы от  $n$  до  $m$
2. Написать функцию которой передается объект  $a$  она выводит его содержимое.