



Занятие №4

# Функции



# Функции

Функции представляют собой набор инструкций, которые выполняют определенное действие или вычисляют определенное значение.

Определение функции начинается с ключевого слова **function**, после которого следует имя функции.

Наименование функции подчиняется тем же правилам, что и наименование переменной. После имени функции в скобках идет перечисление параметров.

Затем в фигурных скобках идет тело функции, содержащее набор инструкций.

```
function nameOfFunction(argument1, argument2) {  
    body  
}
```

Вызвать функцию надо по имени

```
<script>  
    function sayHello () {  
        console.log("hello")  
    }  
  
    sayHello()  
</script>
```

# Параметры функции

Функция в JavaScript может принимать параметры.

Параметры представляют способ передачи в функцию данных. Параметры указываются в скобках после названия функции.

Параметры функции передаются при создании.  
Аргументы передаются при вызове функции.

```
<script>  
  
    function sayHelloByName (name) {  
        console.log("Hello", name)  
    }  
  
    sayHelloByName("first name")  
  
</script>
```

# Возвращаемое значение функции.

В объявлении функции для возврата значения используется ключевое слово «**return**», после которого указываются передаваемые из функции данные.

Выполнение команды «**return**» прекращает работу функции, даже если в ее теле есть дальнейшие инструкции.

## Функция может возвращать значения

```
<script>  
    function returnHello (name) {  
        return "Hello" + name  
    }  
  
    console.log(returnHello("second name"))  
</script>
```



# Неопределенное количество параметров

С помощью оператора ...  
(многоточие) мы можем  
указать, что с помощью  
параметра можно  
передать переменное  
количество значений.

```
function func (...params) {  
    // цикл for-in бегаёт по индексам  
    for (let i in params) {  
        console.log(i)  
    }  
    // цикл for-of бегаёт по значениям  
    for (let i of params) {  
        console.log(i)  
    }  
}  
  
func(1, 8, "e", true, "-")  
func(1, 87, 1, 0, 25)  
func()  
func("r", "e", "q")
```

# Задание

1. Написать функцию, которая принимает 2 числа и возвращает меньшее из них.
2. Написать функцию, которая возводит переданное число в указанную степень.
3. Написать функцию, которая принимает 2 числа и знак (+ - \* /), считает пример и возвращает результат.
4. Написать функцию, которая проверяет, является ли переданное ей число простым.

# Задание

5. Написать функцию, которая принимает число и выводит таблицу умножения для этого числа. Вызовите функцию для всех чисел от 2 до 9.
6. Написать функцию, которая принимает от 1 до 5 чисел и возвращает большее из них.
7. Написать функцию, которая выводит все четные или нечетные числа, в указанном пользователем диапазоне. Какие числа выводить, определяется третьим параметром типа bool (true – четные, false – нечетные).

# Рекурсия

Рекурсией называют ситуацию, в которой функция вызывает сама себя.

Функция

## getFactorial

возвращает

значение 1, если параметр n

равен 1, либо возвращает

Результат опять же

функции

getFactorial, то в

нее передается

значение  $n-1$ .

```
function factorial (n) {  
  if (n === 1) {  
    return 1  
  }  
  else {  
    return n * factorial(n - 1)  
  }  
}
```

```
console.log(factorial(7))  
console.log(factorial(4))  
console.log(factorial(12))
```

# Определение чисел ФИБОНАЧЧИ

Каждое последующее число равно сумме двух предыдущих чисел.

Каждое последующее  
число равно  
сумме двух предыдущих  
чисел.

```
function fib (n) {  
  if (n === 0) {  
    return 0  
  }  
  if (n === 1) {  
    return 1  
  }  
  else {  
    return fib(n-1) + fib(n-2)  
  }  
}
```

```
console.log(fib(4))  
console.log(fib(7))  
console.log(fib(12))
```

# Задание

1. Написать функцию, которая выводит все числа из заданного пользователем диапазона в прямом порядке и в обратном порядке.



ВОПРОСЫ

ПРАКТИКА



Давайте подведем итоги.