

Функции в языке JavaScript

Функции в JavaScript выполняют важную роль. Они обеспечивают сокращение кода за счет вынесения его в отдельные места. За урок мы научимся создавать функции и передавать в них значения.

Функции можно назвать небольшими подпрограммами, куда можно вынести повторяющийся код и обращаться к нему, когда это будет нужно. Функции значительно облегчают построение программ, так как нам не надо копировать однотипный код множество раз, а можно просто воспользоваться одной общей функцией.

Многие путают функции и методы и не понимают отличий между ними. На самом деле отличий нет, так как что методы, что функции являются одним и тем же. Функции что записаны вне классов называют функциями, а функции что записаны внутри классов называются методами.

Точно такая же ситуация обстоит с переменным. В классах переменные называются полями, а вне классов - переменными.

В JS функции создаются при помощи ключевого слова `function`. Каждая функция может иметь какие-либо параметры или же не иметь их вовсе. Функции способны что-либо возвращать в ходе выполнения кода, если это требуется.

Создание функции

На основе всех данных наша функция будет выглядеть следующим образом:

```
function test() {  
    console.log("Вывод чего-либо в консоль");  
}
```

Функция выше не принимает никаких параметров и ничего не возвращает. Она просто пишет слово в консоль. Давайте разнообразим функцию и добавим параметр:

```
function test(word) {  
    console.log(word);  
}
```

Теперь функция принимает параметр, который будет отображен в консоли.

Если функция должна что-либо вернуть, то прописываем тип данных который будет возвращен. Для возвращения данных используем ключевое слово `return`:

```
function test(some_number) {  
    some_number *= 2;  
    return some_number;  
}
```

Локальные и глобальные переменные

В JavaScript есть несколько полей видимости: локальная и глобальная. Если записать глобальную переменную, то такая переменная будет видна повсюду и с ней можно работать отовсюду в документе. Если записать локальную переменную, то такая переменная будет видна лишь в той области, где она записана.

Для создания глобальной переменной её необходимо прописать вне функции, класса или же объекта. Для создания локальных переменных вам достаточно поместить переменную в функцию. Переменная в функции будет видна лишь внутри блока с функцией и нигде более.

Задание 1:

Интересная функция

Создайте функцию с одним параметром.

Если параметр будет больше за число 18, то возвращайте значение true. Иначе возвращает окно с вопросом: «Родители разрешили?».

Код функции:

```
function checkAge(age) {  
    if (age > 18) {  
        return true;  
    } else {  
        return confirm("Родители разрешили?");  
    }  
}
```

Количество вызовов функции

Создайте функцию, что будет выводить на экран число. Это число будет суммарным количеством вызовов функции.

Решение задания:

```
var count = 0;

function test() {
    count++;
    console.log("Количество вызовов: " + count);
}

test(); // Первый вызов
test(); // Второй вызов
test(); // Третий вызов
```

Подсчет километров

Создайте функцию, что будет принимать 2 параметра: время и скорость.

Функция должна возвращать расстояние, которое проедет пользователь за время что было передано в функцию.

Полученный результат выведите в консоль.

Решение задания:

```
function km_count(km, hour) {
    var res = km * hour;
    printRes(res);
    return res;
}
```

```
function printRes(res) {  
    var word = res == 1 ? "Вы проедете: 1 километр" : "Вы проедете: " + res + "  
километров";  
    console.log(word);  
}
```

```
var res = km_count(21, 2.4);
```