1. **Функции в JavaScript** - это блоки кода, которые имеют свое имя (кроме анонимных) и могут быть вызваны по этому имени.  
     
   **Функции  в JavaScript** бывают **встроенными**, например alert() и **пользовательскими**, которые программист создает сам.

**В Функции в JavaScript** могут передаваться некоторые параметры, например необходимые для вычисления возвращаемых значений.

**Функции в JavaScript** являются объектами, и как следствие могут, например, присваиваться переменным, передаваться другим функциям, можно присваивать значения их свойствам и вызывать их методы. (Понятие объекта мы рассмотрим более подробно в следующем уроке).

**Функции в JavaScript** именуются по тем же правилам что и [переменные](https://stepik.org/lesson/41646/step/1?unit=20086).

Основное назначение функции заключается в том, чтобы избавить программу от дублирования кода. Записав определенную последовательность команд можно вызывать ее сколько угодно раз.

1. Создание, т.е. **определение функции** начинается с ключевого слова **function**, после которого указываются следующие данные:

**Имя функции**, которое будет использовано при создании переменной, которой будет присвоен объект нашей новой функции.  
**Список входных параметров** в круглых скобках (может быть пустым).  
**Тело функции** в фигурных скобках - собственно список исполняемых команд (также может отсутствовать) .

Давайте рассмотрим пример создания функции, которая будет выводить слова "Hello World!"

function printText() { //Определение функции

document.write("Hello World!"); //Тело функции - вывод текста в документ

};

Теперь для того, чтобы напечатать на экране  слова "Hello World!" нам достаточно в коде просто вызвать эту функцию:

printText();

1. А если мы хотим создать функцию, в которую будет передаваться произвольное значение, которое нужно вывести, то нам нужно создать функцию с параметром:

function printText(a) { //Определение функции

document.write(a); //Тело функции - вывод содержимого переменной "а" в документ

};

Теперь мы можем вызывать нашу функцию с параметром:

printText("Hello World!");

1. При определении функции после ключевого слова может не быть имени.  Например, мы можем объявить функцию из прошлого шага вот таким способом:

var printText = function(a) {document.write(a);};

Функция может быть сразу вызвана с необходимым входным параметром. В следующем примере функция сразу будет вызвана - выведет в документ фразу "Hello World!". Если бы функция возвращала какой-либо результат, он бы был записан в переменную printText.

var printText = function(a) {document.write(a);}("Hello World!");

Для того чтобы обозначить что функция является самовызываемой, ее заключают в скобки. Это не требование стандарта, однако  так делают для большей очевидности.

var printText = (function(a) {document.write(a);}("Hello World!"));

**Для возврата** результата работы функции используется ключевое слово **return**.

Оно помещается в теле функции в том месте, где нужно вернуть результат и выйти из функции обратно к месту в коде, откуда она была вызвана.

function mySquare(x) {

return x\*x;

}

В приведенном примере функция возвращает переданное ей значение, возведенное в квадрат.

Ключевое слово **return**может указываться в функции не один раз, а сколько необходимо. В приведенном ниже примере можно видеть что выход из функции может осуществляться в разных местах функции в зависимости, например, от значения входного параметра:

function divideOneTo(x) {

  if (x != 0) {

    return 1/x;

  } else {

  return "А на ноль делить нельзя!";

  }

}

В данном случае мы сделали функцию, возвращающую результат деления единицы на входной параметр, если он отличен от нуля, или фразы "А на ноль делить нельзя!" - если мы передаем в функцию ноль.

Однако, использование ключевого слова **return**не является обязательным. Если в теле функции его нет, то после выполнения всех команд интерпретатор закончит выполнение функции. Значение, которое возвращает эта функция будет "**undefined**".

Функция, имеющая имя, может вызывать саму себя. Такие функции называются **рекурсивными**.

Например, в приведенном ниже примере функция с помощью рекурсивного вызова вычисляет факториал:

function factorial(x) { //Объявление функции

if (x <= 1) return 1; //Проверка условия окончания расчета

return x \* factorial(x-1); //Вызов этой же функции с уменьшенным на 1 аргументом

}

**Важное замечание:** в рекурсивных функциях нужно не забывать про условие прекращения рекурсии, иначе код "зациклится".

В примере выше, таким условием будет проверка, что x <= 1