JS HOME

JS Output JS Syntax

JS Objects JS Scope

JS Events JS Strings

JS Math

JS Dates

JS Arrays

JS Switch

JS Break

JS RegExp

JS Errors

JS Hoisting

JS Mistakes

JS JSON

Forms API

DOM Intro

DOM CSS

JS Window JS Screen

JS Location JS History

JS Timing JS Cookies

JS Browser JS Quiz

JS Loop For

It just takes

₹750* +0

feed a child

For an entire

year!

DONATE

W3SCHOOLS EXAMS

HTML, CSS, JavaScript,

PHP, jQuery, Bootstrap

and XML Certifications

COLOR PICKER

LEARN MORE:

Color Converter

Google Maps

Animated Buttons

Modal Boxes

Modal Images

Tooltips

Loaders

JS Animations

Progress Bars

Dropdowns Slideshow

Side Navigation

HTML Includes

Color Palettes

SHARE THIS PAGE

```
CSS JAVASCRIPT
                                                                                                          TUTORIALS ▼
                                                                                                                           REFERENCES ▼
JS Tutorial
                              BUILD AWESOME APPS, FASTER
                              Try today and get a FREE t-shirt!
                                                                                                  MAKE IT YOURS >
JS Introduction
JS Where To
                           JavaScript Function Definitions
JS Statements
                           « Previous
                                                                                                                           Next Chapter »
JS Comments
JS Variables
JS Operators
                           JavaScript functions are defined with the function keyword.
JS Arithmetic
                           You can use a function declaration or a function expression.
JS Assignment
JS Data Types
JS Functions
                           Function Declarations
                           Earlier in this tutorial, you learned that functions are declared with the following syntax:
                             function functionName(parameters) {
JS String Methods
                               code to be executed
JS Numbers
                             }
JS Number Methods
                           Declared functions are not executed immediately. They are "saved for later use", and will be executed later, when they are
                           invoked (called upon).
JS Date Formats
JS Date Methods
                             Example
JS Array Methods
                               function myFunction(a, b) {
JS Booleans
                                    return a * b;
JS Comparisons
JS Conditions
JS Loop While
JS Type Conversion
                                  Semicolons are used to separate executable JavaScript statements.
                                  Since a function declaration is not an executable statement, it is not common to end it with a semicolon.
JS Debugging
JS Strict Mode
                           Function Expressions
JS Style Guide
JS Best Practices
                           A JavaScript function can also be defined using an expression.
                           A function expression can be stored in a variable:
JS Performance
JS Reserved Words
                             Example
JS Forms
                               var x = function (a, b) {return a * b};
Forms Validation
JS Objects
                           After a function expression has been stored in a variable, the variable can be used as a function:
Object Definitions
Object Properties
Object Methods
                             Example
Object Prototypes
                               var x = function (a, b) {return a * b};
JS Functions
                               var z = x(4, 3);
Function Parameters
Function Invocation
Function Closures
                           The function above is actually an anonymous function (a function without a name).
JS HTML DOM
                           Functions stored in variables do not need function names. They are always invoked (called) using the variable name.
DOM Methods
                                  The function above ends with a semicolon because it is a part of an executable statement.
DOM Document
DOM Elements
DOM HTML
                           The Function() Constructor
DOM Animations
DOM Events
                           As you have seen in the previous examples, JavaScript functions are defined with the function keyword.
DOM EventListener
                           Functions can also be defined with a built-in JavaScript function constructor called Function().
DOM Navigation
DOM Nodes
DOM Nodelist
                             Example
JS Browser BOM
                               var myFunction = new Function("a", "b", "return a * b");
                               var x = myFunction(4, 3);
JS Navigator
JS Popup Alert
                           You actually don't have to use the function constructor. The example above is the same as writing:
                             Example
JS Examples
JS Examples
                               var myFunction = function (a, b) {return a * b};
JS HTML DOM
                               var x = myFunction(4, 3);
JS HTML Input
JS HTML Objects
JS HTML Events
JS Certificate
JS Summary
                                  Most of the time, you can avoid using the new keyword in JavaScript.
JS References
JavaScript Objects
HTML DOM Objects
                           Function Hoisting
                           Earlier in this tutorial, you learned about "hoisting".
                           Hoisting is JavaScript's default behavior of moving declarations to the top of the current scope.
                           Hoisting applies to variable declarations and to function declarations.
                           Because of this, JavaScript functions can be called before they are declared:
                             myFunction(5);
                             function myFunction(y) {
                                  return y * y;
                           Functions defined using an expression are not hoisted.
                           Self-Invoking Functions
                           Function expressions can be made "self-invoking".
                           A self-invoking expression is invoked (started) automatically, without being called.
                           Function expressions will execute automatically if the expression is followed by ().
                           You cannot self-invoke a function declaration.
                           You have to add parentheses around the function to indicate that it is a function expression:
                             Example
                               (function () {
                                    var x = "Hello!!";
                                                              // I will invoke myself
                               })();
```

```
The function above is actually an anonymous self-invoking function (function without name).
Functions Can Be Used as Values
JavaScript functions can be used as values:
```

function myFunction(a, b) { return a * b;

```
JavaScript functions can be used in expressions:
 Example
   function myFunction(a, b) {
        return a * b;
```

```
But, JavaScript functions can best be described as objects.
JavaScript functions have both properties and methods.
```

Example

Functions are Objects

var x = myFunction(4, 3) * 2;

Example

var x = myFunction(4, 3);

function myFunction(a, b) { return arguments.length;

The **typeof** operator in JavaScript returns "function" for functions.

```
The toString() method returns the function as a string:
 Example
   function myFunction(a, b) {
       return a * b;
   var txt = myFunction.toString();
```

The arguments.length property returns the number of arguments received when the function was invoked:

REPORT ERROR

Top 10 Tutorials HTML Tutorial

CSS Tutorial

JavaScript Tutorial

W3.CSS Tutorial

Bootstrap Tutorial

SQL Tutorial

PHP Tutorial

jQuery Tutorial

Angular Tutorial XML Tutorial

```
A function defined as the property of an object, is called a method to the object.
       A function designed to create new objects, is called an object constructor.
                                                                                                    Next Chapter »
« Previous
```

PRINT PAGE

Top 10 References

HTML Reference

CSS Reference

JavaScript Reference

W3.CSS Reference

Browser Statistics

PHP Reference

HTML Colors

HTML Character Sets

jQuery Reference

AngularJS Reference

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2016 by Refsnes Data. All Rights Reserved. Powered by W3.CSS

FORUM

Top 10 Examples

HTML Examples

CSS Examples

JavaScript Examples

W3.CSS Examples

HTML DOM Examples

PHP Examples

jQuery Examples

ASP Examples

SVG Examples

ABOUT

Web Certificates

HTML Certificate

HTML5 Certificate

CSS Certificate

JavaScript Certificate jQuery Certificate

PHP Certificate

Bootstrap Certificate

XML Certificate