Solutions to Quick Checks

Carey, JavaScript Web Warriors 7e, 9780357638002, Chapter 2: Working with Functions, Data Types, and Operators

Table of Contents

[Quick Check Answers 1](#_Toc47602313)

[Quick Check 1 1](#_Toc47602314)

[Quick Check 2 2](#_Toc47602315)

[Quick Check 3 3](#_Toc47602316)

[Quick Check 4 3](#_Toc47602317)

# Quick Check Answers

Quick Check 1

1. What is the difference between a named function and an anonymous function?

A named function is a set of related statements that is assigned a name. You can use this name to reference, or call, this set of statements in other parts of your code. An anonymous function, on the other hand, is a set of related statements with no name assigned to it. The statements in an anonymous function work only in a single context—the place in the code where they are located.

**Feedback**: Functions are named so that they can be referenced and used elsewhere in the script in the same way that a variable is named so that it can referenced and used. If the function does not need to be referenced, it can be entered as an anonymous function without a name. Generally, named functions are used for functions that are accessed repeatedly in the program and anonymous functions are used for functions accessed only once. Anonymous functions are also important for use with event handlers.

1. What is a command block?

A collected group of JavaScript statements enclosed within a set of opening and closing curly braces.

**Feedback**: Command blocks are used in many JavaScript statements to encapsulate multiple JavaScript statements.

1. Provide an expression to call the findSqr() function using 10 as the parameter value.

findSqr(10)

**Feedback**: Any named function can be accessed or called by including the name of the function within the following JavaScript expression:

*functionName*(*paramValues*);

where *functionName* is the name of the function and *paramValues* are the values assigned to the parameters (if any) of the function.

1. What is the difference between the capture phase and the bubbling phase for an event occurring within a website?

Under JavaScript’s event model, an event like click is first tracked in the capture phase, moving down the object hierarchy from the most general object (the browser window) down to the specific (the figure itself). The capture phase is followed by the bubbling phase as the event moves back up the object hierarchy ending with the browser window.

**Feedback**: A JavaScript event listener is always listening for the event as it goes down the object hierarchy (being captured) or goes up (being bubbled).

1. An HTML file contains the following tag:

<input type="button" value="Submit" id="SubmitButton" />

Provide the code to run the submitOrder() function in response to the user clicking Submit button using the following approaches: as a HTML attribute, as an object property, and as an event listener (during the bubbling phase).

HTML attribute  
<input type="button" value="Submit" id="SubmitButton"  
 onclick="submitOrder()" />  
  
Object Property  
document.getElementById("SubmitButton").onclick = submitOrder;  
  
Event Listener  
document.getElementById("SubmitButton").addEventListener("click", submitOrder)

**Feedback:** There are three ways of associating a function with an event: by adding an attribute to an HTML tag, by adding a property to a page object, or by attaching an event listener to a page object. Each approach has its advantages and disadvantages.

Quick Check 2

1. What is the difference between block scope and function scope?

Variables declared with let are block scoped in that their scope is limited to the command block in which they are defined, or any commands blocks nested therein. Variables declared with the var keyword have function scope in that their scope is limited to the function in which they are defined, or any functions nested therein.

**Feedback:** Knowing the scope of a variable determines where in the program the variable can be referenced.

1. What is the scope of variables declared with the let keyword?

block scope

**Feedback:** Variables declared with let are block scoped in that their scope is limited to the command block in which they are defined, or any code nested within that block.

1. What are the possible values for a Boolean variable?

true and false

**Feedback:** Boolean values are used in with comparison operators and conditional operators for evaluating expressions that are either true or false.

1. What is the difference between a strongly typed and a loosely typed language? Which is JavaScript?

Variables in a strongly typed language do not change after they have been declared. Programming languages that do not require you to declare the data types of variables are called loosely typed programming languages. A JavaScript is a loosely typed language.

**Feedback:** Many programming languages require that you declare the type of data that a variable contains. Programming languages that require you to declare the data types of variables are called strongly typed programming languages. A strongly typed language is also known as statically typed, because data types do not change after they have been declared. Programming languages that do not require you to declare the data types of variables are called loosely typed or duck typed programming languages. A loosely typed language is also known as dynamically typed, because data types can change after they have been declared. JavaScript is a loosely typed programming language. In JavaScript, you are not required to declare the data type of variables and, in fact, are not allowed to do so. Instead, a JavaScript interpreter automatically determines what type of data is stored in a variable and assigns the variable’s data type accordingly.

1. What is the escape sequence for the newline character?

\n

Feedback: An escape character tells compilers and interpreters that the character that follows it has a special purpose. In JavaScript, the escape character is the backslash ( \ ).

Quick Check 3

1. What is the difference between a binary operator and a unary operator?

A binary operator requires an operand before and after the operator. A unary operator requires just a single operand either before or after the operator.

Feedback: JavaScript operators are binary or unary. A binary operator requires an operand before and after the operator. The equal sign in the statement myNumber = 100; is an example of a binary operator. A unary operator requires just a single operand either before or after the operator. For example, the increment operator (++), an arithmetic operator, is used to increase an operand by a value of one.

1. How does JavaScript deal with code that performs arithmetic operations on string values?

When performing arithmetic operations on string values, a JavaScript interpreter attempts to convert the string values to numbers. However, JavaScript interpreters do not convert strings to numbers when you use the addition operator. When you use the addition operator with strings, the strings are combined instead of being added together.

Feedback: When performing arithmetic operations on string values, a JavaScript interpreter attempts to convert the string values to numbers. However, JavaScript interpreters do not convert strings to numbers when you use the addition operator. When you use the addition operator with strings, the strings are combined instead of being added together.

1. What is a comparison operator? What kind of value does it return?

A comparison operator, or relational operator, is used to compare two operands and determine if one value is greater than another. A Boolean value of true or false is returned after two operands are compared.

Feedback: The comparison operators (== and ===) consist of two and three equal signs, respectively, and perform a different function than the one performed by the assignment operator that consists of a single equal sign (=). The comparison operators compare values, whereas the assignment operator assigns values. Confusion between these two types of operators is a common programming error.

1. What is a falsy value? What are the six falsy values in JavaScript?

A falsy value is a value that is treated in comparison operations as the Boolean value false. The six falsy values in JavaScript are "", -0, 0, NaN, null, and undefined.

Feedback: Developers commonly take advantage of falsy and truthy values to make comparison operations more compact.

Quick Check 4

1. When performing operations with operators in the same precedence group, how is the order of precedence determined

The order of precedence is determined by the operator’s associativity, which is the order in which operators of equal precedence execute. Associativity is evaluated from left-to-right or right-to-left, depending on the operators involved.

Feedback: When using operators to create expressions in JavaScript, you need to be aware of the precedence of an operator. Operator precedence is the system that determines the order in which operations in an expression are evaluated.

1. Provide the expression to retrieve the value entered in the input control with the id “memberNumber”.

document.getElementById("memberNumber").value

Feedback: The value inserted into an input control can be referenced using the following value property:

*object*.value

where *object* is a reference to the input control.

1. Provide the expression to determine whether the checkbox control with the id “primeMember” has been checked”.

document.getElementById("primeMember").checked

1. Provide a statement that attaches an event listener to the input control with the id “memberNumber”, running the function updateRegistration() when that control’s value is changed

document.getElementById("memberNumber").addEventListener("change", updateRegistration);

Feedback: To apply the onchange event handler to the form control, apply the statement:

*object*.onchange = *function*;

With the event listeners, the statement appears as

*object*.addEventListener("change", *function*);