CHAPTER 2 – Functions, Data Type and Operators

Contents

[Working with Functions 1](#_Toc13695066)

[Defining Functions 1](#_Toc13695067)

[Handling Events with Functions 2](#_Toc13695068)

[Using Return statement 2](#_Toc13695069)

[Understanding Variable Scope 3](#_Toc13695070)

[Using Built-in JavaScript Functions 3](#_Toc13695071)

[Working with Data Types 4](#_Toc13695072)

[Escape characters and sequences 5](#_Toc13695073)

[Comparison and Conditional Operators 6](#_Toc13695074)

## Working with Functions

Methods - Procedures associated with an object are called methods.

Functions – In JavaScript programming, you can write your own procedure called functions. A function is a related group of JavaScript statements that are executed as a single unit.

* + Virtually identical to methods
    - Not associated with an object
  + Must be contained within a script element

### Defining Functions

* Use ***named function*** when you want to reuse code

**function** *name\_of\_function*(*parameters*) {

*statements*;

}

* Use ***anonymous function*** for code that runs only once

**function** (*parameters*) {

*statements*;

}

* Function statements
  + Do the actual work
  + Contained within function braces
* Put functions in an external .js file - Reference at bottom of body section
* Function call
  + Consists of function name followed by parentheses

Contains any variables or values assigned to the function parameters

### Handling Events with Functions

#### Specify function as value for HTML attribute

<input type="submit" onclick="showMessage()" />

#### Specify function as property value for object

document.getElementById("submitButton").onclick = showMessage;

#### Use addEventListener() method

var submit = document.getElementById("submitButton");

submit.addEventListener("click", showMessage, false);

* Adding an event listener is most flexible
  + Separates HTML and JavaScript code
  + Can specify several event handlers for a single event

Example:

var el = document.getElementById("submitButton");

el.addEventListener("click", function(){

window.alert("Thanks for your order!");

},false);

### Using Return statement

* + Returns a value to the statement calling the function
  + Use the return keyword with the variable or value to send to the calling statement

Example:

function averageNumbers(a, b, c) {

var sum\_of\_numbers = a + b + c;

var result = sum\_of\_numbers / 3;

return result;

}

### Understanding Variable Scope

* Variable scope
  + Where in code a declared variable can be used
* Global variable
  + Declared outside a function
    - Available to all parts of code
* Local variable
  + Declared inside a function
    - Only available within the function in which it is declared
  + Cease to exist when the function ends
  + Keyword **var** required

Example:

var color = "green";

function duplicateVariableNames() {

var color = "purple";

document.write(color);

// value printed is purple

}

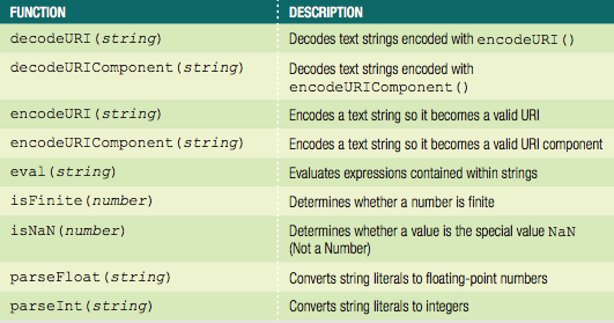
duplicateVariableNames();

document.write(color);

// value printed is green

### Using Built-in JavaScript Functions

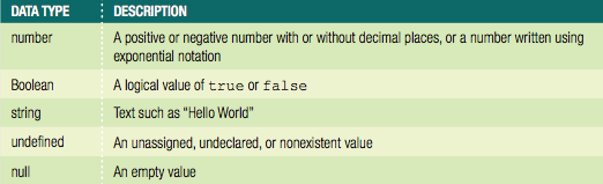
Called the same way a custom function is called



## Working with Data Types

#### Data type

* + Specific information category a variable contains
* Primitive types
  + Data types assigned a single value



* The null value: data type and a value
  + Can be assigned to a variable
  + Indicates no usable value
  + Use: ensure a variable does not contain any data
* Undefined variable
  + Never had a value assigned to it, has not been declared, or does not exist
  + Indicates variable never assigned a value: not even null
  + Use: determine if a value being used by another part of a script

Example:

var stateTax;

document.write(stateTax);

stateTax = 40;

document.write(stateTax);

stateTax = null;

document.write(stateTax);



* Strongly typed programming languages
  + Require declaration of the data types of variables
  + Strong typing also known as static typing
    - Data types do not change after declared
* Loosely typed programming languages
  + Do not require declaration of the data types of variables
  + Loose typing also known as dynamic typing
    - Data types can change after declared
* JavaScript interpreter automatically determines data type stored in a variable

Examples:

diffTypes = "Hello World"; // String

diffTypes = 8; // Integer number

diffTypes = 5.367; // Floating-point number

diffTypes = true; // Boolean

diffTypes = null; // Null

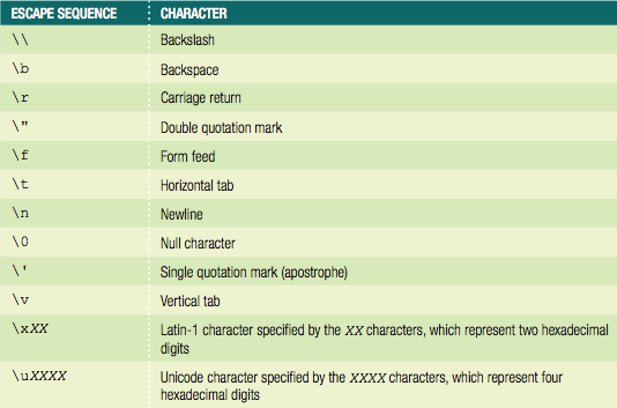
### Escape characters and sequences

Escape character

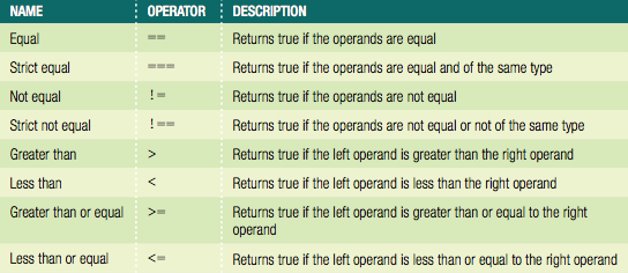
* + - Tells the compiler or interpreter that the character that follows has a special purpose
    - In JavaScript, escape character is backslash (\)

Escape sequence

* + - Escape character combined with other characters
    - Most escape sequences carry out special functions



### Comparison and Conditional Operators



#### Conditional operator

* + Executes one of two expressions based on conditional expression results

Syntax

*conditional expression* ? *expression1* : *expression2*;

* + If conditional expression evaluates to true:

Then expression1 executes

* + If the conditional expression evaluates to false:

Then expression2 executes

#### Falsy and Truthy Values

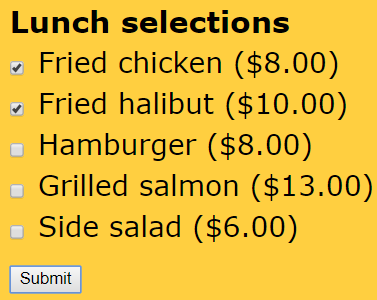
Six falsy values treated like Boolean false:

* + ""
  + -0
  + 0
  + NaN
  + null
  + undefined

All other values are truthy, treated like Boolean true

***Hands-On Project 2-4***

Create a script that totals the price of all the elements a user selects; adds sales tax; display the order total in an alert box:





<article>

<h2>Lunch selections</h2>

<form>

<input type="checkbox" id="item1" value="8" />

<label for="item1">Fried chicken ($8.00)</label>

<input type="checkbox" id="item2" value="10" />

<label for="item2">Fried halibut ($10.00)</label>

<input type="checkbox" id="item3" value="8" />

<label for="item3">Hamburger ($8.00)</label>

<input type="checkbox" id="item4" value="13" />

<label for="item4">Grilled salmon ($13.00)</label>

<input type="checkbox" id="item5" value="6" />

<label for="item5">Side salad ($6.00)</label>

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

</form>

</article>

<script>

function calcTotal(){

var itemTotal = 0;

var item1 = document.getElementById("item1");

var item2 = document.getElementById("item2");

var item3 = document.getElementById("item3");

var item4 = document.getElementById("item4");

var item5 = document.getElementById("item5");

(item1.checked) ? (itemTotal += 8) : (itemTotal +=0 );

(item2.checked) ? (itemTotal += 9) : (itemTotal += 0);

(item3.checked) ? (itemTotal += 8) : (itemTotal += 0);

(item4.checked) ? (itemTotal += 13) : (itemTotal += 0);

(item5.checked) ? (itemTotal += 6) : (itemTotal += 0);

var salesTaxRate = 0.07;

var orderTotal = itemTotal + (itemTotal \* salesTaxRate);

window.alert("Your order total is $"+orderTotal);

}

//Add the event listener:

document.getElementById("submit").addEventListener("click",calcTotal,false);

</script>