Study guide by ExamNotes.net

Study Guide for 1D0-135 JavaScript Fundamentals v 4

The JavaScript Fundamentals exam is required to obtain the Enterprise Developer certificate, however does not return a professional certificate of it's own.

The exam is comprised of one section only. The exam has a total of 50 questions. You must score a 75% or higher to receive credit for passing the exam. The exam contains only single answer multiple-choice questions.

Skills Being Measured

Candidates taking the JavaScript Fundamentals exam should know how to use the features of the JavaScript language and design client-side, platform-independent solutions. Candidates should know how to write JavaScript programs, script for the JavaScript object model, control program flow, validate forms, animate images, target frames, and create cookies.

Introduction to JavaScript

The Netscape Corporation originally developed JavaScript. Originally named Live Script, Netscape changed the name to JavaScript more for name recognition than similarities to Java. Server Side JavaScript is known as Live Wire.

There are Currently 5 versions of JavaScript 1.0,1.1,1.2,1.3,and 1.4.

Key JavaScript Characteristics

- JavaScript is a scripting Language
- JavaScript is Object Based, not Object Oriented
- JavaScript is Event-Driven
- JavaScript is Platform Independent.
- JavaScript enables quick development
- JavaScript is relatively Easy to learn

Client Side JavaScript is parsed and executed on the client machine.
Server Side JavaScript is stored, and executed on the server machine.

Differences between JavaScript and Java

JavaScript	**Java**
Interpreted by the client	Compiled on server before execution on the client
Object-Based. Uses built-in, extensible objects, but no classes or inheritance	Object-Oriented. Applets consist of object classes with inheritance
Variable data types not declared	Variable data types must be declared
Dynamic Binding. Object references checked at run time	Static Binding. Object references must exist at compile time
Secure cannot write to hard disk	Secure. Cannot write to hard disk
Code embedded in HTML	

JavaScript – Netscape client side scripting language, for a large part of its functionality JS depends on objects and their attendant methods and properties.
VBScript – Microsoft relies less on traditional object classes and more on dynamic built in customizing functions.
ECMAScript – The formalization of JavaScript using input to design the language from Netscape, and Microsoft.

JavaScript code, and Comments:

```
<html>
<head>
<title>Title</title>
<script language="JavaScript">
<!--
JavaScript code goes here
// -->
</script>
</head>
```

Working with Variable and Data

Variables: Variables are named spaces of memory. They are containers that hold values, which may be accessed repeatedly in your scripts.

Variable Data Types

Data Type	Examples
Number	Any Numeric Value
String	Any String of Characters
Boolean	True or False values only
Null	A special keyword for the null value

Literals: The actual data values you provide in JavaScript

var correctanswer=20;

Keywords: predefined identifiers that form the foundation of JavaScript. They perform unique duties, such as declaring variable (var) and defining functions (function)

Reserved Words: Words reserved for explicit use by the language.

You may not use Keywords or Reserved words as variable, functions, objects, or methods.

Expressions: An expression is part of a statement that is evaluated as a value with the exception of the assignment expression, which assigns a value to a variable.

Operators: Used in expressions to store or return a value.
Operator Varieties

- Assignment
- Arithmetic
- Unary
- Logical

Inline Scripting: Embedding script within the HTML Elements.
<body onUnLoad="alert('GoodBye! Come Back Soon')">

Functions, Methods, and Events

Functions: Organized blocks of code that handle actions generated by user events. Can improve program efficiency and readability.
JavaScript allows you to build your own functions, or use built in functions such as parseInt()

Defining a function:

```
function function_name(argument1, argument2)
{
//Code Block
//Statements go here
}
```

Calling Statement: A statement that transfers program execution to a subroutine, procedure, or a function. When the function is complete execution transfers back to the call statement. These statements are used to call a function to process a functions statement.
OnUnLoad=function_name()

Methods: An Action that may be performed by an Object. Methods that return values are interchangeable with functions.

Events: Actions that are generated by a user while navigating through a page. For Instance Submitting form data, or random navigation.
JavaScript contains predetermined event handlers to deal with events. Events may be used to launch functions or methods.

Controlling Program Flow

Program Flow: The ability to branch one of multiple processes depending on the result of some condition. Used to control decisional program flow.

If Statements: A simple condition where no alternate condition exists. If condition evaluates to True then code block executes.
If Else: If Else statements are used to evaluate for 2 conditions a true and false statement. If condition is true execute first code block. Else execute second code block.
Else if: Used to evaluate multiple conditions where any of an array could be true. If statement true execute code block else if true execute second code block, else execute last code block.

```
if (expression)
{
        statements to execute if expression is true
}
else
{
        statements to execute if expression is false
}

if (userAnswer==25)
{
        alert("You are 25");
}
else
{
        alert("You are not 25");
}
```

While Statement: The While statement is used to execute a code group for as long (while) a certain condition is true.

```
var num = 0;
while (num < 10)
{
        document.write(num);
        num++;
}-
```

For Statement: The primary purpose of the for statement is to repeat a group of statements for some particular range of values.

```
For (x,y,z)
{
code Block
}
```

The Break Statement: Typically used in an IF statement, If a certain condition is met the user is allowed to break outside a loop, if not the loop will continue.

The Continue statement: The continue statement s used to force control back to the top of the loop.

Switch Statement: The switch statement acts as multiple If statements, however the Switch statement allows you to specify a default set of statements to execute if the script does not find a match.
Do While Statement: The do while statement does not check the conditional expression until after the first time through the loop.

The JavaScript Object Model

!! Know all the properties and methods of the Different Objects.

JavaScript Object Model: JavaScript was designed explicitly for web page use. To take advantage of the different features and capabilities provided by the browser, special browser objects have been built into the JavaScript Language. The JavaScript Object Model Divides objects into three groups, browser objects, language objects, and form field objects.

The Window Object: The highest-level object in the JavaScript object hierarchy
It is the default object
Doesn't need to be referenced by name
Dot notation
window.status or status
Reference window name when opening additional windows

Properties	Methods	Event Handlers
defaultStatus	alert(MessageText)	onLoad
frames	close()	onUnLoad
length	confirm(MessageText)	
name	open(url,name,Featurelist)	
parent	prompt(Messagetext,response)	
self	setTimeOut(expression,Time)	
status	clearTimeOut(timerID)	
top		
window		

The Document Object:
Subordinate to the window object in the window hierarchy
Defined when the <BODY> tag is evaluated in an HTML page

Property	Method
alinkColor	close()
anchors	open()
bgColor	write(content)
cookie	writeln()
fgcolor	
forms	
linkColor	
lastModified	
links	
location	
referrer	
title	
vlinkColor	

The Image Object: Allows you to manipulate images in IE 4.0 , and Netscape 3.0 and Later.
Subordinate to the document object
Loads as an array

document.images[0].src = newImage.jpg;

document.[imageName].src = “newImage.gif”
To create an instance of the image object:

imageVarName = new Image();

To assign a source:

imageVarName.src = “image.gif”;

Properties	Event Handlers
src	onAbort
height&Width	onError
length	onKeyDown
lowsrc	onKeyPress
complete	onKeyup
hspace&vspace	onLoad
border	

The History Object: Subordinate to the window object in the window hierarchy. Allows you to access browser history through scripts.
Subordinate to window object
back and forward buttons use these objects

history.back();

history.forward();

Properties	Methods
Length	Back()
	Forward()
	Go(x)

The Location Object: Subordinate to the window object in the window hierarchy. Allows you to specify URLs in a script.
Subordinate to window object
Can create buttons instead of text or graphic links to send users to different targets.
Can also tie the change of location to some other portion of script.
location.href = http://www.ciwcertified.com;

Property	Description
href	Specifies the URL of a file or site
protocol	http or ftp currently in use
host	Refers to the hostname:port portion of URL
hostname	Specifies the host name of the URL
port	
pathname	Indicates the path to the desired file
search	Returns the text following the ? Character
hash	Specifies the internal link anchor name (#) in
url	

The Navigator Object: Refers to information about the browser being used.
var userBrowser = navigator.appName;

Properties	**Methods**
appVersion	JavaEnabled()
language	preference()
mimeTypes	taintEnabled()
platform	
plugins	
userAgent	

JavaScript Language Objects
JavaScript refers to HTML forms and form elements as objects that can be used to manipulate string, date, and math information in useful ways. JavaScript has special Language objects String, Date, Array, and Math that are designed to deal with HTML elements.

The String Object: The String Object has methods that allow you to write strings of characters as well as test for the presence of certain characters, extract a subset (called a substring) for a given string, retrieve the number of characters from a string, and find the position of given characters in a string.

String Object Formatting Methods

Method	**Example**	**HTML Equivalent**
anchor("anchorName")	"part 2".anchor("p2")	part 2
big()	"Welcome".big()	<big>Welcome</big>
blink()	"New".blink()	<blink>New</blink>
bold()	"Hot".bold()	Hot
fixed()	"Name Phone".fixed()	<tt>Name Phone</tt>
fontcolor("color")	"savage".fontcolor("blue")	savage
fontsize(size)	"Dude".fontsize(6)	Dude
italics()	"Other".italics()	Other
link("url")	"NASA".link("http://nasa.gov")	NASA
small()		
strike()		
sub()		
sup()		
toLowerCase()		
toUpperCase()		

String Object Special Characters

Character	Description
\b	Backspace
\f	Form Feed
\n	New Line
\r	Carriage Return
\t	Tab
\”	Double Quote
\’	Single Quote
\\	Backslash

Evaluation Strings

The length property
The indexOf() and lastindexOf() methods
The substring() method
The charAt() method

The Array Object:

Arrays are lists of variables with multiple memory slots to hold the variables.
Slots hold different values.
Each slot can be referenced by its index number
Zero-based - first slot [0]
Some objects have built-in arrays

- Form elements
- Images

The Date Object

Based on arrays
The only way to use date and time information in JavaScript
var myDateObject = new Date();

Date Methods

Method	Description
getDate()	Retrieves the day number(1-31)
getDay()	Retrieves the day-of-week value (0-6; 0 is Sunday)
getMonth()	Retrieves the month number (0-11; 0 is January)
getYear()	Retrieves the number of years since 1900
getHours()	Retrieves the hour number (0-23; 0 is midnight)
getMinutes()	Retrieves the number of minutes (0-59)
getSeconds()	Retrieves the number of seconds (0-59)
getTime()	Retrieves the number of milliseconds that have elapsed since midnight on Jan 1, 1970
setDate(value)	Assigns the date within the month (1-31)

The Math Object:

Contains properties and methods that aid in the creation of advanced mathematical calculations
Object is not a value but a static object that contains some values as properties
Such as PI
Don't need to create a new instance
Cannot reassign values to math objects
Syntax: variableName = Math.method(value);

Interactive Forms

OverView of Form Controls:

Button	Reset	
Checkbox	Select	
Hidden		Submit
Password	Text	
Radio	Textarea	

Referring to Form Objects:
Two ways:
By name

```
<form name="bob">a
<input type="text" name="bubba">
</form>
document.bob.bubba.value;
```

By index number in the form elements array
document.forms[0].elements[0].

Features of Form Objects:

Properties	**Methods()**	**Event Handlers**
action	reset()	onReset
elements	submit()	onSubmit
encoding		
length		
method		
name		
target		

The Button Object

The simplest of all objects

Properties	Methods()	Event Handlers
name	click()	onClick
value		

<input type="button" value="do action" onClick="function();">
document.formName.buttonName.value="New Value";
document.formName.buttonName.value;

The CheckBox Object

A checkbox is a toggle object in the shape of a small square that can be checked on or off

boolean input - on or off

Properties	Methods()	Event Handlers
checked	click()	onClick
defaultChecked		
name		
value		

The Text and Text Area Objects:

Text objects can only display a single line of text

Textarea objects can display multiple, scrolling lines of text

Properties	Methods()	Event Handlers
defaultValue	blur()	onBlur
name	focus()	onFocus
value	select()	onSelect

The Radio Button Object

Used to select among mutually exclusive options

Properties	Methods()	Event Handlers
checked	click()	onClick
defaultChecked		
length		
name		
value		

The Select Object

A drop-down list or a list box of items used in an HTML form
No methods are defined for the select object

Properties	Subproperty	Description
length	formRef.length	
name	formRef.name	
options	formRef.options	
	length	formRef.options.length integer value indicating the number of options in the select object
	selectedIndex	formRef.options.selectedIndex Integer value of array number of the selected option
options[i]	Where i is the index number of the currently selected option	
	defaultSelected	formRef.options[i].defaultSelected boolean value determined by whether the option was selected by default in the original option tag

Cookies and JavaScript Security

Cookies: Small memory-resident pieces of information sent from a server to your computer
Often referred to as “persistent cookies” and “persistent HTML” A cookie is originally stored in the memory of your computer, but a web server can also store this information more permanently on a hard drive.

Cookies are sent When a user generates an HTTP request.
Two actions can occur:

- A server can deposit cookies on the user’s hard drive
- Any cookies already on the user’s system that match the server’s domain can be passed

along in the request header
A server can set, or deposit, a cookie only if a user visits that particular site

- Cross-domain posting is impossible
- Some domains “share” cookies

Storing Cookies:

- Cookies store name=value pairs as text strings
- Domains can store no more than 20 cookies on a user’s computer

- Users can delete cookie files

Assigning:
- document.cookie = “name=value”;
- document.cookie = “name=value;expires=date;secure”;

Testing for cookie presence:
- alert(document.cookie);

Why use Cookies?
Authentication
Storing user information
- Operating system and browser type
- Service provider
- IP address
- History of sites visited

State maintenance with cookies

Test for cookie Presence:
- You can easily test for the presence of any cookie by using “document.cookie” in your script

Clearing a Cookie:
- To clear a cookie, reassign it, adding an expiration date that has already passed

Controlling Cookies in the Browser:
- Netscape Navigator 4.0 options
    1. Accept all cookies
    2. Accept only cookies that get sent back to the originating server
    3. Disable cookies
    4. Warn before accepting a cookie

JavaScript Security Issues:
- JavaScript and Helper Applications
- Malicious and Accidental programs
- Previous browser versions and security

Controlling Frames in JavaScript

Frames help to maximize the sophistication of your web site by allowing you to leave some information with the user at all times. This technique provides both ease of navigation, and a means to control the users paths and destinations.

Understanding Frames and Targets:
Frames exist within framesets
Each frame in a frameset will usually have a default source file

```
<html>
<head>
<title>Frames demo</title>
</head>
<frameset cols="25,*">
        <frame src="a.htm" name="a">
        <frame src="b.htm" name="b">
</frameset>
</html>
```

You will need to understand how the frame nesting process works in HTML

Targeting Frames in JavaScript:

To target frames, use either of the following techniques:

- By target name
- By number in the frames array

Changing two or more frames:

- To change two (or more) frames at once in a script, write a simple function that includes two (or more) location.href lines
    - All frames are contained in an array of the window object
        - frames[0]
    - To reference:
      relationship.frameName
    - Ex. top.twofer.location.href = home.htm
      Would change the loaded page to home.htm

Frames, Function and Variables:

Variables can be stored in any of the files involved in making a frameset

- relationship.frameName.variableName
  document.bgColor = parent.userBGColorChoice;-
  relationship.frameName.functionName()
  parent.testFunction();

Targeting windows

- floatingWindow = open("","NewWin","width=200,height=375");
- floatingWindow.location.href = "new.htm";
- Can call:
    - windowName.functionName();
    - windowName.variableName;

Targeting Windows:

Targeting the “opener” window
The window that launches the floating window is known as the “opener” object
Can change the location of the original window from the floating window with:
window.opener.location.href = “joe.htm”;
Can also call:
window.opener.functionName();
window.opener.variableName;

Client Side Java-Script

Have a good understanding of how to create Image Maps in HTML.
Understand how to add scripts to Image Maps
(Add inline Scripts with event handlers to the <AREA> Tag)
Understand how to call functions from an Image Map
(Add inline Scripts with event handlers to the <AREA> Tag)
Understand how to use the Navigator Object for Browser Detection.

```
<HTML>
<HEAD>
<TITLE>Browser Detector</TITLE>
<SCRIPT>
<!--
 function showInfo() {
   var info=""
   info += "\nWelcome, " + navigator.appCodeName
   info += " user!\nYou are using the "
   info += navigator.appName + " browser,\nversion "
   info += navigator.appVersion + ".\nYour user agent "
   info += "information is " + navigator.userAgent
   alert(info)
 }

//-->
</SCRIPT>
<link rel="STYLESHEET" type="text/css" href="jsclass.css">
</HEAD>
<BODY BGCOLOR="#CC9966" text="#330000">
<H3 align=center>Click on the image for browser information.<BR><BR>
<A HREF="javascript:void(showInfo());"><IMG SRC="images/logo17-6.gif"
border=0></A>
</H3>
</BODY>
</HTML>
```

Custom JavaScript Objects

!! Very Important. There were many questions on the test about custom JavaScript Objects.

Advantages to Custom Objects

- Creating a user-defined object offers two major advantages:
    - You can create sophisticated solutions with a minimum of coding
    - You can represent programming constructs as objects

Creating a Custom JavaScript Object: The Constructor:

- You define, or create, a custom JavaScript object with a special function called a constructor
    - The constructor defines the properties and methods of your object

Creating an Instance of a custom Object:

- To instantiate then populate the properties of each new instance with actual data, you must declare variables

Creating Object Methods:

- You can create as many methods for your object as you want
- Methods can be as simple or as sophisticated as you want

Creating Functions for your Objects:

- When you need to evaluate multiple objects in an array, you need a function as opposed to a method