



OESTERREICHISCHE
COMPUTER GESELLSCHAFT[®]
AUSTRIAN
COMPUTER SOCIETY

OCG Client-Side Web Scripting (JavaScript)



OCG Client-Side Web Scripting (JavaScript) Syllabus Version 2.0

Professionelle
Gestaltung von Websites

Österreichische Computer Gesellschaft (OCG)

Wollzeile 1-3

A-1010 Wien

Tel: +43 1 512 02 35-0

Fax: +43 1 512 02 35-9

E-Mail: info@ocg.at

Web: www.ocg.at

Hinweis

Die offizielle Version des OCG Client-Side Web Scripting (JavaScript) Syllabus ist die auf der Website <http://www.ocg.at> veröffentlichte Version.

Haftung

Die OCG hat dieses Dokument mit Sorgfalt erstellt, kann aber weder Richtigkeit und Vollständigkeit der enthaltenen Informationen zusichern noch Haftung für durch diese Informationen verursachte Schäden übernehmen.

Urheberrecht

OCG 2012

OCG Client-Side Web Scripting (JavaScript)

Der nachfolgende Lernzielkatalog stellt die Grundlage für den praktischen Test für dieses Modul dar. Das Modul wird mit einem eigenen OCG Zertifikat „OCG Client-Side Web Scripting (JavaScript)“ abgeschlossen.

MODULZIELE

OCG Client-Side Web Scripting (JavaScript) erfordert von den Kandidatinnen und Kandidaten die Beherrschung grundlegender Programmierkenntnisse und andererseits den praktischen Umgang mit einer Scriptsprache für die Erstellung interaktiver Webseiten.

Die Kandidatin/der Kandidat soll

- ausgewählte Elemente des Sprachkerns (z.B. Datentypen und Werte, Variable, Operatoren, Kontrollstrukturen und Funktionen) erfolgreich anwenden können.
- objektorientierte Modelle für den Aufbau eines Dokuments in einem Browserfenster kennen. Durch Verwenden von Objektmethoden und -eigenschaften können Browserfenster und Browserfensterinhalte abhängig vom Benutzerverhalten beeinflusst werden.
- die Grundkonzepte von Ajax kennen.

VORKENNTNISSE

OCG Image Editing
OCG Web Editing
OCG Webdesign & Usability

Kategorie	Wissensgebiet	Ref.	Fertigkeit
1.1 Grundlagen	1.1.1 Geschichte und Bausteine	1.1.1.1	Die Entwicklungsgeschichte von JavaScript kennen.
		1.1.1.2	Zwischen dem JavaScript-Sprachkern und der Verwendung dieses Sprachkerns in einem Webbrowser (clientseitiges JavaScript) unterscheiden können.
		1.1.1.3	Den Begriffsinhalt von DOM (Document Object Model) verstehen.
		1.1.1.4	Die Bedeutung der Entwicklung von DOM Level 0 (Alt-DOM) über DOM Level 1 zum DOM Level 2 verstehen.

Kategorie	Wissensgebiet	Ref.	Fertigkeit
		1.1.1.5	Verstehen, wofür man JavaScript verwenden sollte und wofür nicht.
		1.1.1.6	Verstehen, was mit barrierefreiem JavaScript (unobstrusive JavaScript) gemeint ist.
		1.1.1.7	JavaScript von der Programmiersprache Java unterscheiden können.
	1.1.2 Editoren	1.1.2.1	Den Nutzen von Editoren mit Programmierfunktionen bei der Erstellung von JavaScript-Inhalten erkennen.
		1.1.2.2	Werkzeuge zum Erstellen von JavaScript-Inhalten kennen.
		1.1.2.3	Den Nutzen einer Sprach- und Objektreferenz einer Programmiersprache kennen.
1.2 Sprachkern	1.2.1 Lexikalische Struktur	1.2.1.1	Groß-/Kleinschreibung richtig anwenden können.
		1.2.1.2	Anweisungen richtig formulieren können, besonders das Beenden von Anweisungen mit Strichpunkt.
		1.2.1.3	Whitespace und Zeilenschaltungen anwenden können.
		1.2.1.4	Literale verwenden können: Integerwerte, Fließkommazahlen, Zeichenketten (Strings), Wahrheitswerte (Boolsche Werte)
		1.2.1.5	Identifizier als Variablenamen verwenden können.
		1.2.1.6	Kommentare verwenden können.
		1.2.1.7	Die Bedeutung von Keywords (Schlüsselwörtern) verstehen.
		1.2.1.8	Den Unterschied zwischen explizit definierten Variablen (mit Schlüsselwort <i>var</i>) und implizit definierten Variablen (ohne Schlüsselwort <i>var</i>) kennen.
	1.2.2 Datentypen und Werte	1.2.2.1	Einfache (Skalare) und zusammengesetzte Datentypen verwenden können.
		1.2.2.2	Ganzzahlige Werte, Dezimalzahlen, Wahrheitswerte und Zeichenketten verwenden können.
		1.2.2.3	Die besonderen Werte <i>null</i> , <i>undefined</i> , <i>Infinity</i> , <i>NaN</i> , <i>Number.POSITIVE_INFINITY</i> , <i>Number.NEGATIVE_INFINITY</i> und <i>Number.NaN</i> verwenden können.

Kategorie	Wissensgebiet	Ref.	Fertigkeit
	1.2.3 Operatoren	1.2.3.1	Zuweisungsoperatoren anwenden können.
		1.2.3.2	Zwischen Zuweisungen <i>by Value</i> und <i>by Reference</i> unterscheiden können.
		1.2.3.3	Arithmetische Operatoren anwenden können.
		1.2.3.4	Vergleichsoperatoren anwenden können.
		1.2.3.5	Logische Operatoren anwenden können.
		1.2.3.6	Prioritätsregel (Reihenfolgeregel) für Operatoren anwenden können.
		1.2.3.7	Abarbeitung von Operatoren bei gleicher Priorität (Assoziativität von Operatoren) verstehen und anwenden können.
		1.2.3.8	Den Typerkennungsoperator (<i>typeof</i>) anwenden können.
		1.2.3.9	Den <i>delete</i> -Operator anwenden können.
		1.2.3.10	Den <i>void</i> -Operator anwenden können.
		1.2.3.11	Den <i>in</i> -Operator anwenden können.
		1.2.3.12	Den Kommaoperator (,) anwenden können.
	1.2.4 Zeichenketten	1.2.4.1	Escape-Sequenzen in Zeichenketten verwenden können.
		1.2.4.2	Den Verknüpfungsoperator bei Zeichenketten anwenden können.
	1.2.5 Arrays	1.2.5.1	Arrays instanziiieren können.
		1.2.5.2	Arrays mit einem Literal definieren können.
	1.2.6 Objekte	1.2.6.1	Ein Objekt auf Basis der eingebauten (<i>built-in</i>) Klasse <i>Object()</i> instanziiieren und verwenden können.
		1.2.6.2	Ein Objekt mit einem Literal definieren können.
		1.2.6.3	Eine Klasse selbst definieren und Objekte dieser Klasse instanziiieren können.
		1.2.6.4	Den Begriff „Konstruktor“ verstehen und Konstruktorfunktionen richtig anwenden können.
		1.2.6.5	Die Bedeutung des Schlüsselworts <i>this</i> verstehen.
		1.2.6.6	Instanzeigenschaften und –methoden von Klasseeigenschaften und –methoden unterscheiden können.

Kategorie	Wissensgebiet	Ref.	Fertigkeit
		1.2.6.7	Klassenhierarchien und die Begriffe „Superclass“ und „Subclass“ verstehen.
		1.2.6.8	Die Math-Klasse kennen. Die Methoden <i>pow()</i> , <i>max()</i> , <i>min()</i> , <i>random()</i> , <i>floor()</i> und <i>round()</i> der Math-Klasse anwenden können.
		1.2.6.9	Die Date-Klasse kennen. Die Methoden <i>getMonth()</i> , <i>getDate()</i> , <i>getHours()</i> und <i>getMinutes()</i> der Date-Klasse anwenden können.
		1.2.6.10	Die Eigenschaft <i>length</i> der Stringklasse verwenden können.
		1.2.6.11	Die Methoden <i>indexOf()</i> , <i>replace()</i> , <i>substr()</i> , <i>substring()</i> , <i>toLowerCase()</i> und <i>toUpperCase()</i> der String-Klasse verwenden können.
	1.2.7 Kontrollstrukturen	1.2.7.1	Wenn-Dann-Entscheidungen und Switch-Kontrollstrukturen implementieren können.
		1.2.7.2	Schleifen: <i>For</i> -, <i>While</i> -, <i>Do/While</i> - und <i>For/in</i> -Schleifen erstellen können.
		1.2.7.3	Schleifensteuerung und Schleifenkörper erstellen können.
		1.2.7.4	Break und Continue anwenden können.
		1.2.7.5	Die <i>with</i> -Anweisung anwenden können.
		1.2.7.6	Anweisungsblöcke bilden können.
	1.2.8 Funktionen: Aufbau und Erstellung	1.2.8.1	Funktionsnamen, Argumente und Funktionskörper verstehen und anwenden können.
		1.2.8.2	Funktionen mit einem Literal definieren können.
		1.2.8.3	Reichweite von Variablen verstehen und einsetzen können.
		1.2.8.4	Zwischen der Übergabe von Funktionsargumenten <i>by Value</i> und <i>by Reference</i> unterscheiden können.
		1.2.8.5	Funktionen mit und ohne Rückgabewert in JavaScript deklarieren können.
		1.2.8.6	Die Auswirkung von Groß- und Kleinschreibung bei Funktionsnamen kennen.
	1.2.9 Ausgewählte interne Funktionen	1.2.9.1	Die Verwandtschaft und den Unterschied der Begriffe „Funktion“ und Methode“ erklären können.
		1.2.9.2	Die Funktionen <i>parseInt()</i> , <i>parseFloat()</i> und <i>isNaN()</i> verwenden können.

Kategorie	Wissensgebiet	Ref.	Fertigkeit
		1.2.9.3	Die Funktion <i>eval()</i> verwenden können.
1.3 Browser-seitiges (client-seitiges) JavaScript	1.3.1 Zusammenspiel mit (X)HTML und CSS	1.3.1.1	Script-Tags für die Einbindung von JavaScript in Dateien verwenden können.
		1.3.1.2	JavaScript-Code aus externen Dateien in eine Datei einbinden können.
		1.3.1.3	Den Begriff „Namespace“ verstehen und die Verwendung von Namespaces begründen können.
		1.3.1.4	Objekte in externen JavaScriptdateien als Namespace benutzen können.
		1.3.1.5	JavaScript Code als Wert von HTML-Event-Handlerattributen verwenden können.
		1.3.1.6	JavaScript-Code mit dem Pseudo-Protokoll <i>javascript:</i> in einem URL verwenden können.
		1.3.1.7	JavaScript Code innerhalb eines Script-Tags in einer (X)HTML-Datei in einem CDATA-Abschnitt verpacken können.
	1.3.2 Level 0 DOM (Alt-DOM)	1.3.2.1	Das Window-Objekt kennen.
		1.3.2.2	Die Status- und Location-Eigenschaft des Window-Objekts verwenden können.
		1.3.2.3	Die Methoden <i>alert()</i> , <i>close()</i> , <i>print()</i> , <i>prompt()</i> , <i>confirm()</i> , <i>open()</i> , <i>focus()</i> , <i>setInterval()</i> und <i>setTimeout()</i> des Window-Objekts verwenden können.
		1.3.2.4	Das Document-Objekt kennen und in die Klassenhierarchie einordnen können.
		1.3.2.5	Die Eigenschaften <i>title</i> , <i>lastModified</i> und <i>URL</i> des Document-Objekts verwenden können.
		1.3.2.6	Die Methode <i>write()</i> des Document-Objekts verwenden können.
		1.3.2.7	Das <i>links[]</i> Array des Document-Objekts kennen und verwenden können.
		1.3.2.8	Das <i>forms[]</i> Array des Document-Objekts kennen und verwenden können.
		1.3.2.9	Das <i>elements[]</i> Array des Form-Objekts kennen und verwenden können.
		1.3.2.10	Den hierarchischen Aufbau des Level 0 DOMs kennen.

Kategorie	Wissensgebiet	Ref.	Fertigkeit
		1.3.2.11	Mit Level 0 DOM den Focus auf Formularelemente setzen können.
	1.3.3 Level 1 DOM, Level 2 DOM	1.3.3.1	Die DOM-Darstellung als Baum verstehen.
		1.3.3.2	Die Knotenarten Document, Element und Text im Baum unterscheiden können.
		1.3.3.3	Die Bedeutung des (X)HTML id-Attributes im Zusammenspiel mit browserseitiger JavaScript-Programmierung verstehen.
		1.3.3.4	Die Methode <i>getElementById()</i> des Document-Objekts verwenden können.
		1.3.3.5	Die Methoden <i>createAttribute()</i> , <i>setAttributeNode()</i> , <i>createElement()</i> , <i>createTextNode()</i> und <i>appendChild()</i> verwenden können.
		1.3.3.6	Mit Level 2 DOM den Focus auf Formularelemente setzen können.
	1.3.4 Ereignisbehandlung	1.3.4.1	Den Begriff „Ereignis“ richtig erklären können.
		1.3.4.2	Das ursprüngliche Ereignisbehandlungsmodell in JavaScript erklären können.
		1.3.4.2	Die HTML-Attribute <i>onmouseover</i> , <i>onmouseout</i> , <i>onload</i> und <i>onclick</i> zum Reagieren auf Events verwenden können.
		1.3.4.3	Die Hauptpunkte der Weiterentwicklung des ursprünglichen Ereignismodells darstellen können.
		1.3.4.4	Browserunterschiede bei der Behandlung von Ereignissen im weiterentwickelten Ereignismodell kennen.
	1.3.5 Ajax (Asynchronous JavaScript and XML)	1.3.5.1	Den Grundgedanken von Ajax erklären können.
		1.3.5.2	Die Methoden und Eigenschaften des <i>XMLHttpRequest</i> Objekts kennen.
		1.3.5.3	Das <i>XMLHttpRequest</i> Objekt für synchronen Datentransfer verwenden können.
		1.3.5.4	Das <i>XMLHttpRequest</i> Objekt für asynchronen Datentransfer verwenden können.



ZERTIFIKATE

BASIS

ECDL Core

Der Standard für Computeranwendung

ECDL Web Editing

Der Einstieg ins Webdesign

ECDL Image Editing

Der Einstieg in die Bildbearbeitung

ECDL CAD

Der Einstieg ins technische Zeichnen

OCG Typing Certificate

Zertifikat für 10-Finger-Tastaturschreiben

pm basic

Der Einstieg ins Projektmanagement

LERNUNTERLAGEN ZUM EINSTIEG

PC-Start

Der Computereinstieg für alle

you start IT

Der Computereinstieg für Kids

PROFI

ECDL Advanced

Professionelles Arbeiten im Office

Textverarbeitung

Datenbank

Tabellenkalkulation

Präsentation

OCG Web Management

Professionelle Gestaltung von Websites





**OESTERREICHISCHE
COMPUTER GESELLSCHAFT**®
**AUSTRIAN
COMPUTER SOCIETY**

Österreichische Computer Gesellschaft (OCG)
Wollzeile 1-3
A-1010 Wien

Tel: +43 1 512 02 35-0

Fax: +43 1 512 02 35-9

E-Mail: info@ocg.at

Web: www.ocg.at