Module
JavaScript

Academic year 2
JavaScript (NL/INT)

NHL Stenden University of Applied Sciences
Van Schaikweg 94
PO Box 2080
7801 CB Emmen
Phone: +31 (0)591 853100
www.stenden.com

JavaScript

Version 1.2: 1 September 2018

Academic Year 2018-2019

Module Coordinator:	Author:
Name: Niels Doorn Email: niels.doorn@stenden.com	Name: Niels Doorn Email:niels.doorn@stenden.com

Preface

In the "JavaScript" module you will increase your knowledge of programming. JavaScript is a frequently used programming language that is used to add interaction to web pages, among other things. You will increase your knowledge by studying the book "*JavaScript & jQuery*" and by completing assignments. The reason behind this is that software development is a skill that you can only master through practise.

The primary objective of this module is for students to learn to apply JavaScript. The JavaScript programming language will be used and students may choose their own development environment. However, the source code files and other files must be submitted. After this module, the student has a sound foundation to work with JavaScript.

The study load for this module is 84 hours (3 EC).

Experience with various methods used at Stenden University in the past has proven that the method of the "JavaScript" module is the best method for learning.

Niels Doorn

Emmen, 1 September 2018

Contents

1 Introduction

1.1 Rationale of the module

HTML and CSS are the foundation for many websites. Therefore, extensive knowledge of this matter is essential for website builders. In this module you will continue your development in building web applications. Besides HTML and CSS, a number of new techniques will be applied. We will use JavaScript and the Document Object Model. We will also use jQuery, a framework that uses JavaScript.

And we will discuss the Ajax technique (Asynchronous JavaScript and XML) at the end of the module. We will also look at integration with other services and the use of JSON.

Typical professional situation

Typical professional situation

After graduating from her higher education programme Informatics, Karin started working as a junior software engineer at WebSystem. At WebSystem, Karin is responsible for the entire implementation project of content management systems for clients.

JavaScript and PHP play in important role in these content management systems.

One of the projects Karin is involved in is the EcoGoodies web shop. For this project, Karin is responsible for programming the extra desired functionalities within the content management system.

This module uses jQuery.
One example of this is that the client, EcoGoodies, was in need of an extra module for a new concept. This module automatically selects the greenest energy supplier out of all the suppliers in the region.

1.2 Competences

Within this module, the starting professional will develop three competences related to analysing, designing, and producing level 1 applications.

	Managing	Analysing	Consulting	Designing	Realising
User Interaction					• Realising and testing dynamic ICT and/or digital media products, applying relevant graphic elements, sound, picture, and animation. ***(level 1)***
Business Processes					
Infrastructure					
Software		• Collecting and validating functional requirements for a software system with a single stakeholder, following a standard method. • Defining acceptance criteria for the above functional requirements. ***(level 1)***		• Designing a software system using modelling techniques according to a standard method. ***(level 1)***	• Building and testing a simple software system, and making it available. ***(level 1)***
Hardware Interfacing					

1.3 Theme of the module

This module period has no central theme; the JavaScript module is a separate component.

1.4 Objectives of the module

Upon completion of this module, students have the ability to:
1. Apply the topics discussed in the JavaScript & jQuery book, in the design of computer programs;
2. Analyse a given professional situation;
3. Establish a procedure for developing a (web) application;
4. Develop a web application based on JavaScript.

1.5 Prerequisite knowledge

Students must meet the exam requirements corresponding to the "(X)HTML" module to follow the "JavaScript" module.

1.6 Agreements

There are a number of agreements regarding fonts to distinguish text and programming code.

Normal text	Verdana. (font size 10)
Program code	Courier new. (font size 10)
Program output	Courier new. (font size 10)

1.7 Version Management

Version	Date	Author	Description
0.1	23-06-2016	N. Doorn	Initial document
1.0	29-8-2016	N.Doorn	Remarks W. Alsaqaf processed.
1.1	10-11-2016	N.Doorn	Feedback from students processed
1.2	2-9-2018	N.Doorn	Updated and changed jQuery to a JS library

2 Examination

2.1 General

There is a final assignment to assess whether the students have achieved the module objectives.

Table 2.1 outlines the final examination along with relevant information on the standards, marks and credits.

This module has one exam component:

- a final assignment (see paragraph 2.2)

The final assignment will be graded according to an assessment form (Scoring Rubrics) as included as appendix 4.

This module will be graded with a pass when the standard has been reached according to the scoring rubrics from appendix 4.

Table 2.1 Examination Overview

Test method	Max. points	Standard %	Standard in points	Credits	Deadline	Resit
Final assignment	100	55%	55	3 EC	Week 8	Week 9
Total	100	55%	55	3 EC		

2.2 Assessment of the group's final assignment/tutorials

All steps (components) of the final assignment must be done and discussed with the lecturer individually. **If during this discussion it is brought to light that the student did not do the assignment himself/herself, the assignment will receive a failing grade.**

The final group assignment consists of three components (also see appendix 2):

1. Proposal of the final assignment
2. Design of the program
3. Programmability of the program.

Students may only proceed with the (following) component after the previous component has been approved by the lecturer.
The final assignment will be graded according to an assessment form (Scoring Rubrics) as included as appendix 4.

Additionally, the following is included in the assessment of the final group assignment:

1. Student attendance;
   Students must attend at least 75% of the tutorials for the entire scheduled time.
2. The assignments:
   Assignments must be signed off by the lecturer in the weeks the corresponding tutorial is scheduled. If it is not signed off in that week, students may redo it in the next week. *For example:* The tasks scheduled in week 6, should be signed off in week 6. The assignment can be redone in week 7. If students have not had a component signed off they may **not continue** with the following components and the assignment will be graded with a 1.

The assignments referred to under item 2 can only be signed off during the tutorials and **not** during consultation sessions. Questions regarding the subject matter can be asked during consultation sessions.

2.3 Active Participation

Students must complete all assignments in the allotted weeks preceding the final result. Students can have their assignments signed off during the scheduled supervised tutorials.

2.4 Module Resit

Students who do not receive a passing mark for their final assignment may do another final assignment. The resit will be in week 9. Students who miss the week-8 deadline will automatically do the resit in week 9.

3 Programme

3.1 Introduction

One of the most common occupations in the ICT field is that of software engineer. The basis for software engineers (as well as other ICT professionals) is a thorough knowledge of programming languages.

As a JavaScript programmer, you (generally) work within a team of (experienced) JavaScript programmers. You are co-responsible for maintaining and developing existing packages and you assist with building and testing new applications for internal and external use.

You acquire an understanding of JavaScript during this module. During the first introductory lecture, you will be informed what is expected of you in the module.

This module contains a number of (weekly) lectures and practical exercises. During the tutorials, students can discuss any issues they might have regarding their assignments with their lecturers and can make sure their assignments are signed off.

3.1.1 Tutorials

Students are to prepare themselves for the tutorials by means of self-study. During the tutorial, they have the opportunity to ask questions. Tutorials are also used to assess student progress.

Students have the opportunity to have the assignments for the corresponding week signed off during the tutorials. Sometimes, students do not complete the week's work; In that case, it is possible to have it signed off the following week. But it is not possible for students to have a considerable number of uncompleted assignments at the end of the period.
Students are required to attend all tutorials. The lecturer is available for parts of the tutorial.

3.2 Curriculum Overview

An overview of the weekly activities is displayed below.

Wk	Assignment no.	Study activity
1	3.3.1	Introductory lecture
1	3.3.2	Tutorial 1 (Study chapters 1-2 of the book) Individual assignment 1
2	3.3.3	Tutorial 2 (Study chapters 3-4 of the book) Resit individual assignment 1. Individual assignment 2.
3	3.3.4	Tutorial 3 (Study chapter 5 of the book). Resit individual assignment 2. Individual assignment 3.
4	3.3.5	Tutorial 4 (Study chapter 6 of the book). Resit individual assignment 3.
5	3.3.6	Tutorial 5 (Study chapter 7 of the book, hand in proposal final assignment). Individual assignment 4
6	3.3.7	Tutorial 6 (Study chapter 8 of the book, feedback on proposal final assignment). Resit individual assignment 4.
7	3.3.8	Tutorial 7 (Study chapter 9 of the book).
8	3.3.9	Hand in final assignment.
9	3.3.10	Resit final assignment.

3.3 Weekly Programmes

Introductory lecture	
Week:	1
Work method:	Lecture
Duration:	1 hour
Objectives:	• Students acquire an overview of the content of the JavaScript module; • Students understand the work method and the assessment in the JavaScript module.
Contents:	During the introductory lecture, the student is briefed on the work methods and practices, assessment, material and contents of the JavaScript module.
Preparation:	
Individual assignments:	

3.3.2 Tutorial 1 (chapters 1 and 2)	
Week:	1
Work method:	Tutorial
Duration:	2 hours.
Objectives:	• *Understanding the main principles of JavaScript and web pages.* • *Learning to apply the basic principles of JavaScript.*
Contents:	This tutorial will handle chapters 1 -2.
Preparation:	Study chapters 1-2. Students should work independently through chapters 1-2 prior to the tutorial in the week these books are on the curriculum. Students are to prepare themselves for the tutorials by means of self-study. During the tutorial they have the opportunity to ask questions. **These questions must be e-mailed to the lecturers no later than 24 hours prior to the tutorial.**
Individual assignments:	Study chapters 1-2. Do individual assignment 1

3.3.3 Tutorial 2 (chapters 3 and 4)	
Week:	2
Work method:	Tutorial
Duration:	2 hours.
Objectives:	• *Understanding and applying functions.* • *Use of Objects.* • *Logical decision and repetition structures in JavaScript.*
Contents:	This tutorial will handle chapters 3 -4.
Preparation:	Study chapters 3-4. Students should work independently through chapters 3-4 prior to the tutorial in the week these books are on the curriculum. Students are to prepare themselves for the tutorials by means of self-study. During the tutorial they have the opportunity to ask questions. **These questions must be e-mailed to the lecturers no later than 24 hours prior to the tutorial.**
Individual assignments:	Study chapters 3-4. Resit individual assignment 1. Do individual assignment 2

3.3.4 Tutorial 3 (Chapter 5)	
Week:	3
Work method:	Tutorial
Duration:	2 hours.
Objectives:	• *Knowledge of how the Document Object Model works.* • *Knowing how to use the Document Object Model for selecting, adding, manipulating and deleting elements.*
Contents:	Chapter 5 is dealt with in this tutorial.
Preparation:	Study chapter 5. Students should work independently through chapter 5 prior to the tutorial in the week these books are on the curriculum. Students are to prepare themselves for the tutorials by means of self-study. During the tutorial they have the opportunity to ask questions. **These questions must be e-mailed to the lecturers no later than 24 hours prior to the tutorial.**
Individual assignments:	Study chapter 5. Resit individual assignment 2. Do individual assignment 3

3.3.5 Tutorial 4 (Chapter 6)	
Week:	4
Work method:	Tutorial
Duration:	2 hours.
Objectives:	• *Understanding the principles of an Event-driven model.* • *Being able to apply Events in JavaScript.*
Contents:	Chapter 6 is dealt with in this tutorial.
Preparation:	Study chapter 6. Students should work independently through chapter 6 prior to the tutorial in the week these books are on the curriculum. Students are to prepare themselves for the tutorials by means of self-study. During the tutorial they have the opportunity to ask questions. **These questions must be e-mailed to the lecturers no later than 24 hours prior to the tutorial.**
Individual assignments:	Study chapter 6. Do individual assignment 4

3.3.6 Tutorial 5 (Chapter 7)	
Week:	5
Work method:	Tutorial
Duration:	2 hours.
Objectives:	• *Understanding what libraries in JavaScript are.* • *Understanding what jQuery is and what you can do with it.* • *Applying jQuery.*
Contents:	Chapter 7 is dealt with in this tutorial.
Preparation:	Study chapter 7. Students should work independently through chapter 7 prior to the tutorial in the week these books are on the curriculum. Students are to prepare themselves for the tutorials by means of self-study. During the tutorial they have the opportunity to ask questions. **These questions must be e-mailed to the lecturers no later than 24 hours prior to the tutorial.**
Individual assignments:	Study chapter 7. Hand in individual assignment 4. Hand in proposal for final assignment.

3.3.7 Tutorial 6 (Chapter 8)	
Week:	6
Work method:	Tutorial
Duration:	2 hours.
Objectives:	• *Understanding and applying Ajax and JSON.*
Contents:	Chapter 8 is dealt with in this tutorial. The student also receives feedback on the proposal for the final assignment.
Preparation:	Study chapter 8. Students should work independently through chapter 8 prior to the tutorial in the week these books are on the curriculum. Students are to prepare themselves for the tutorials by means of self-study. During the tutorial they have the opportunity to ask questions. **These questions must be e-mailed to the lecturers no later than 24 hours prior to the tutorial.**
Individual assignments:	Study chapter 8. Resit individual assignment 4

3.3.8 Tutorial 7 (Chapter 9)	
Week:	7
Work method:	Tutorial
Duration:	2 hours.
Objectives:	• *Understanding what APIs are.* • *Applying APIs for integration with other platforms.*
Contents:	Chapter 8 is dealt with in this tutorial. The student also receives feedback on the proposal for the final assignment.
Preparation:	Study chapter 9. Students should work independently through chapter 9 prior to the tutorial in the week these books are on the curriculum. Students are to prepare themselves for the tutorials by means of self-study. During the tutorial they have the opportunity to ask questions. **These questions must be e-mailed to the lecturers no later than 24 hours prior to the tutorial.**
Individual assignments:	Work on the final assignment.

Tutorial	
Week:	8
Work method:	Tutorial
Duration:	2 hours.
Objectives:	
Contents:	
Preparation:	
Individual assignments:	Hand in final assignment

3.3.10 Individual coaching by appointment	
Week:	9
Work method:	Individual coaching by appointment
Duration:	
Objectives:	
Contents:	
Preparation:	
Individual assignments:	Resit of the final assignment

4 Structure & Organisation

4.1 Module contact hours

The schedule below is an overview of all contact hours for this module.

In addition, students are expected to plan their own (project) meetings to work on their assignments. This also applies to the time students need to prepare and complete (individual) assignments. The schedule also gives a good idea of the expected study load per student.

Table 4.1: Student contact hours (SCH) per week:

Module JavaScript: Student contact hours and study load hours																					
Work method	Number of groups	Week		Week 2		Week 3		Week 4		Week 5		Week 6		Week 7		Week 8		Week 9		Total	
		SCH	SLH	SCH	SLH	SCH	SLH	SCH	SLH	SCH	SLH	SCH	SLH	SCH	SLH	SCH	SLH	SCH	SLH	SCH	SLH
LE	8 to 12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	9
TU	8 to 12	2	9	2	9	2	9	2	9	2	9	2	9	2	9	2	9	0	0	16	72
FE		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SS		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		3	9	2	9	2	9	2	9	2	9	2	9	2	9	2	9	0	0	17	72

LE = Lecture
TU = Tutorial
FE = Final Exam
SS = Self Study hours

SCH = Student Contact Hours (45 minutes)
SLH = Study Load Hours (60 minutes)

5 Literature/software

5.1 Mandatory Study Material

- This module book
- English Book APA:
- Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*: Wiley.

ISBN details:
ISBN 13: 9781118531648.

5.2 References

- Mozilla Developer Network : https://developer.mozilla.org/
- ECMAScript: http://www.ecmascript.org/

5.3 Software

- Sublime Text
- Firefox web browser
- Chrome web browser

6 Module Evaluation

At the end of the module period, the module will be evaluated by means of a questionnaire. This questionnaire covers all components of the module including organisational aspects, content, quality of teaching staff, etc.

We would appreciate your participation in this evaluation. The results will be used to improve the next version of this module.

7 Appendices

Appendix 1:Individual assignments

Assignment 1

a

Write a program that displays 'Hello World' (or your own name) on a web page. Use an HTML page, a CSS file and a JavaScript file (that is, three files).

b

Write a program that performs three different mathematical calculations on two numbers that you saved as variables. Display the calculations and the outcomes on a web page. Again, use separate files for JavaScript, HTML and CSS, if applicable.

c

Write a program that links and displays multiple strings.
Bonus: edit your program so that visitors to your web page can enter the strings.

d

Write a program that can keep a shopping list current. Display all shopping list items on a web page. Your list must have at least 10 items. Change item # 8 (for example, sugar instead of salt).
Bonus: make the shopping list editable directly from the web page.

Assignment 2

a

Write a program that creates an object structure of a book store, an airport or your favourite TV series. The structure must be comprised of at least four classes. Also include the functionalities of instantiation of objects, modification of properties of objects and retrieval of values of objects.

b

Create an object-oriented calculator with functionalities for adding, subtracting, multiplying and dividing.

c

Add the functionalities of involution and extraction of roots to the calculator. Additionally, call all functions with meaningful test values.

d

Write a program that displays the lyrics of the famous song: "99 Bottles of Beer" on a web page. Use meaningful linguistic constructions to do this efficiently in JavaScript (loops).

e

Write a program that checks whether a string is a palindrome.

f

Write a program that converts a string to Morse code.

Assignment 3

a

Write a program that adds new elements to the DOM that include the first 20 numbers of the Fibonacci series.

b

Create one of the four games below with JavaScript:
Four in a Row
Memory
Hangman
Draughts
Ensure a dynamic construction of the playing field by adjusting the DOM from JavaScript.

Assignment 4

a

Write a fortune cookie program that displays a random quote when a button is clicked

b

Create a slide show that allows you to manually browse the images.

c

Create a slide show that displays a new slide every 10 seconds.

d

Create a to-do application with jQuery.

Appendix 2:Final assignment

The final assignment is an open assignment for which you have to develop an **object-oriented game** in JavaScript. You must use **a JavaScript library** and **JavaScript**. The game must be complex enough in order to qualify for the final assignment. Please draw up a **proposal** for your lecturer. If the lecturer approves your proposal, you draw up a **technical design** and implement the game. The code must comply with the coding conventions and general programming requirements of your program.

Examples of games that could be eligible for the final assignment:

- Frogger
- Multi player (single keyboard) Pong
- Pac-man
- Four in a Row

Appendix 3:Example of a Final assignment

Kim decides to make a network-based variant of Pong. Because she wants to use JavaScript, she is looking at the possibilities for running JavaScript on a server. She decides to use Node.js. She uses the Socket.io library as a communication tool. She wants to use Canvas to build Pong itself.

She describes her idea in a document and asks the lecturer for advice. When discussing her idea, the lecturer suggests to use jQuery for a start screen of the game.

Kim creates a design and substantiates her technical choices. She also draws up a UML class diagram of the software for the server, and of the classes of the client section.

Next, she starts implementing the game. She begins with installing the required software, such as Node.js and Socket.io. She watches a tutorial on Socket.io and the use of Canvas. She works towards her final goal in a structured manner. She wants to have effective communication between client and server first. Once that works, she implements the game Pong. She carefully tests every step and uses version management to have the option to revert to a previous version.

She sets up an appointment with her lecturer for submission of the final assignment well in advance.

Appendix 4:Scoring rubrics

Individual Final Assignment							
Component	**0-2 points**	**3-5 points**	**6-7 points**	**8-10 points**	**Points**	**Weight**	**Score**
Design	The design is incomplete. The choices made are not substantiated.	The design is not complex enough. The choices made are insufficiently substantiated.	The design is complete. The main choices are substantiated.	The design is almost flawless and all choices made are substantiated.		2	
Completeness of the code	The code is incomplete and/or does not function properly.	The final product does not function properly and has little connection to the design.	The final product functions technically well and it is based on the design.	JavaScript was used to create a complex and fully functioning program based on the design.		3	
Use of a JavaScript Library	Insufficient use was made of a library.	Use was made of a library, but inconsistently or ineffectively.	A library was used in such a way that it has a significant added value within the program.	A library was used in such a way that a major achievement was made in the functionality and maintainability of the program.		1	
Work attitude	The final assignment was not completed. Insufficient active participation.	The work has been completed as planned but not all details have been completed as described in the assignment. Insufficient active participation.	The work has been completed as planned and in accordance with the assignment. Sufficient active participation.	The work has been completed creatively, innovatively, as planned and in accordance with the assignment. Good active participation.		1	
						Total score final assignment:	
						Mark final assignment = Total score final assignment: / 7	

Component	**Mark**	**Weighting factor**	**Score**
Final assignment		1	
Final mark for JavaScript			