NM1: *web technology*

date: 25/2/08; author: A. Eliëns; version: 1.0 (discussion)

infomation

project-based course, semester 1, 3 ects

contents

The course presents elementary web technology, primarily from the perspective of content development, with special attention to form, style, dynamic behavior and interaction.

course outline(s) – nm1: web technology

In this part a more detailed discussion will be provided of **topics**, **learning goals**, **materials** used, and the actual **structure of the course**, as well as a sketch of the **assignments** given. Also **references** to releveant literature is provided, including **online resources**. At the end, **advice for students** following the course will be given, as well as **hints for the instructor(s)**.

course topic(s) The NM1 course will cover a great variety of topics. Although the main focus of the course is to bring about the skills needed to effectively use the web in later projects and applications, we will also deal with the web as a societal phenomenon, as an element of our daily life.

- web languages for markup, styling and interaction
- elementary web technology authoring tools
- client-side vs server side solutions
- basic scripting, styling and interaction design
- separating content, form, style, behavior and interaction
- elements of web 2.0 business model(s)
- analysis of (commercial) web-sites and portals
- privacy and security – web applications as attack surfaces

Recently, issues of privacy, security and trust gain increaing attention. Also from a technical perspective, some knowledge about **hacking the web** is worthwhile for understanding the potential and dangers in deploying the web as a computing platform.

learning terget(s) The NM1 course is meant to bring competence(s) and skill(s) at varioys levels. In addition, references will be made to literature for further theoretical study. Small projects will further give the experience needed for using web technology in an effective manner.

- skill(s) – scripting, styling, configuration
- knowledge – html, javascript, css, xml, php
- theory – basic(s) of web 2.0
- experience(s) – small scale multi-language web application development
- attitude – understanding, craftmanship, discovery

Apart from practical skills, the course aims at an intuitive understanding of the complexity of the web as a platform for communication and services. To prfit from the course, must have a sufficient degree of curiosity and lust for discovery.

lesson material(s) Although there are many good books available, the is also a wealth of material online, which should suffice for a first introductory course.

- canonical example(s) – *game* / *calculator* / slogan(s) / lookat(s)
- (online) reference material(s) – standard(s) / javascript / www.w3schools.com / kit(s)

- challenging target(s) – heart(s) / labs.google.com (*edu* / *code*)

In the course, we will take an **example-based approach** to learning, that is by showing a selection of examples that demonstrate essential features of web-technology. A dissection, or discussion of these examples will help the students in understanding the most salient features.

course structure – session(s)
Although the course is essentially **project-based**, and to a great extent relies on the students' activity in completing the assignments, there will be a number of lectures, to assist the student in the assignments, and in understanding both the technical and societal context of web applications.

1. introduction of language(s), tool(s) & technology
2. scripting – basic assignment(s)
3. web standard(s) – client-side
4. styling – basic assignment(s)
5. web standard(s) – server-side
6. interaction – basic assignemnt(s)
7. advanced topic(s) – ajax, dhtml, plugin(s), addon(s)
8. presentation of final assignment(s)

The structure presented here is only indicative, and may differs from the actual sequence of topics treated in the lectures. In particular topics such as privacy and security, and societal issues, will be dealt with as items in the lecture, also dependent on actual news items and developments.

assignment(s)
There will a small number of assignemts, to be made by the students individually. The goal of these assignments is to provide a structure that assists the students in exploring the technology. Basic assignemnts (may) include:

basic(s) – *web technology*

1. style – adapt three basic example(s) in style and functionality
2. form – construct a simple calculator or converter in a domain of choice
3. chaos – create the worst, that is visually confusing, web page possible, in an aesthetic way though
4. portal – make a small information site about some topic of choice
5. mimic – evaluate and mimic, e.g. from best of the web

For the final assignment(s) of the course, students are allowed to work indidually, or in groups of two or three (maximally) students. Work done in groups must be proportionally more challenging and complex. Students can make a choice out of (among possibly others):

final(s) – *web technology*

- health information site – www.digifit.eu
- collection of javascript math games – www.cut-the-knot.com
- javascript (visual) adventure game – www.astoundme.com/scottadams
- alternative(s) – submit a proposal

In effect, students will be encouraged to follow their own ideas, in for example implementing a game using web technology, or a deep exploration in style, corresponding with (parallel) course(s) in design.

reference(s) There are many books dealing in one way or another with **web technology**. In particular the **Programmer to Programmer** series of Wrox (www.wrox.com) is highly recommended, especially for learning specific technologies in a practical way.

1. JavaScript: The Definitive Guide by David Flanagan – (amazon)
2. CSS: The Definitive Guide by Eric Meyer
3. Professional Web 2.0 Programming (Wrox Professional Guides) by Eric van der Vlist, Danny Ayers, Erik Bruchez, Joe Fawcett, Alessandro Vernet
4. business model(s) – www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

5. Webbots, Spiders, and Screen Scrapers: A Guide to Developing Internet Agents with PHP/CURL by Michael Schrenk
6. The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws, by Dafydd Stuttard amd Marcus Pinto
7. A. Eliëns, topical media & game development – media.eliens.net

A wealth of material and references can be found at my **topical media & game development** site, including tutorials and examples.

online resource(s)
There is a great, tha is massive, number of online resources, about web technology, including treatments about technology, solutions to HCI issues, style and semantics.

- topic(s) – http://www.digital-web.com/topics
- tutorial(s) – tizag.com / www.w3schools.com
- hci – www.digital-web.com/articles/the_pinball_effect
- style – www.csszengarden.com / webdesignfromscratch.com/web-2.0-design-style-guide.php
- game(s) – nl.youtube.com/experiencewii
- semantic(s) – www.thefutureoftheweb.com/blog/writing-semantic-html

For your basic as well as final assignment(s), the following resources might be useful:

- tool(s) – www.aptana.com / firefox addon(s)
- example(s) – javascript.internet.com / openjsan.org / code.google.com/apis/ajax/playground
- ajax – softwareas.com/ajaxjavascript-8-ways-to-create-graphics-on-the-fly
- graphic(s) – raphaeljs.com/reference.html / me.eae.net/archive/2005/12/29/canvas-in-ie
- css – nubyonrails.com/pages/css_graphs / codepunk.hardwar.org.uk/css2js.htm
- javascript – tool-man.org
- dynamic(s) – www.hunlock.com/blogs/Howto_Dynamically_Insert_Javascript_And_CSS
- menu(s) – www.noupe.com/css/13-awesome-java-script-css-menu.html
- processing – ejohn.org/blog/processingjs
- physic(s) – box2d-js.sourceforge.net
- adventure(s) – zproxy.wordpress.com/2007/11/11/javascript-2d-adventure-game-demo
- dream(s) – jsc.sourceforge.net
- server(s) – www.softintegration.com/webservices

For the final assignment, look for example at the following health site(s)

health

- run – www.runnersworld.com
- nike+ipod – nikeplus.nike.com/nikeplus/?locale=euen_eu
- sport – www.digifit.eu
- moral(s) – morale.erikbenson.com/person/erik

For additional information look at create/resource-web.html.
Particular attention will be given to the complex computational infrastructure provided by the web platform, which is constituted by a wide variety of (scripting) languages and representation formalisms.

prerequisites
CS1 – computer & network architecture(s)

goals and attainment targets

learning terget(s) The NM1 course is meant to bring competence(s) and skill(s) at varioys levels. In addition, references will be made to literature for further theoretical study. Small projects will further give the experience needed for using web technology in an effective manner.

- skill(s) – scripting, styling, configuration
- knowledge – html, javascript, css, xml, php
- theory – basic(s) of web 2.0
- experience(s) – small scale multi-language web application development
- attitude – understanding, craftmanship, discovery

Apart from practical skills, the course aims at an intuitive understanding of the complexity of the web as a platform for communication and services. To prfit from the course, must have a sufficient degree of curiosity and lust for discovery.

place in curriculum
NM1 is an introductory course for all students. It is relevant to NM2 (*interactive visualization*), for publishing the results of CA-projects, as well as for the creation of *individual portfolio(s)*.

application area, motivating examples

The course *web technology* will be based on a collection of online examples, that provide an intuitive illustration of content development issues, as well as online reference material, that will be used for self-study.

lesson material(s) Although there are many good books available, the is also a wealth of material online, which should suffice for a first introductory course.

- canonical example(s) – *game* / *calculator* / slogan(s) / lookat(s)
- (online) reference material(s) – standard(s) / javascript / www.w3schools.com / kit(s)
- challenging target(s) – heart(s) / labs.google.com (*edu* / *code*)

In the course, we will take an **example-based approach** to learning, that is by showing a selection of examples that demonstrate essential features of web-technology. A dissection, or discussion of these examples will help the students in understanding the most salient features.
Although the final assignment will be a small scale project, students are encouraged to find inspiration in current developments and projects on the web.

teaching methods

The course will be organised around lectures, which will introduce basic examples and which will provide an in-depth explanation of the technologies.

course structure – session(s)
Although the course is essentially **project-based**, and to a great extent relies on the students' activity in completing the assignments, there will be a number of lectures, to assist the student in the assignments, and in understanding both the technical and societal context of web applications.

1. introduction of language(s), tool(s) & technology
2. scripting – basic assignment(s)
3. web standard(s) – client-side
4. styling – basic assignment(s)
5. web standard(s) – server-side
6. interaction – basic assignemnt(s)
7. advanced topic(s) – ajax, dhtml, plugin(s), addon(s)

8. presentation of final assignment(s)

The structure presented here is only indicative, and may differs from the actual sequence of topics treated in the lectures. In particular topics such as privacy and security, and societal issues, will be dealt with as items in the lecture, also dependent on actual news items and developments.
In addition, students will be invited to give presentations on selected topics, as wll as concepts and implementations of their own projects.

Grading will be based on basic assignments, a small project with documentation, as well as an essay in which a topic of choice, either technical or in relation to the business model of the web or its societal impact, is discussed in more depth.

assignment(s)
There will a small number of assignemts, to be made by the students individually. The goal of these assignments is to provide a structure that assists the students in exploring the technology. Basic assignemnts (may) include:

basic(s) – *web technology*

1. style – adapt three basic example(s) in style and functionality
2. form – construct a simple calculator or converter in a domain of choice
3. chaos – create the worst, that is visually confusing, web page possible, in an aesthetic way though
4. portal – make a small information site about some topic of choice
5. mimic – evaluate and mimic, e.g. from best of the web

For the final assignment(s) of the course, students are allowed to work indidually, or in groups of two or three (maximally) students. Work done in groups must be proportionally more challenging and complex. Students can make a choice out of (among possibly others):

final(s) – *web technology*

- health information site – www.digifit.eu
- collection of javascript math games – www.cut-the-knot.com
- javascript (visual) adventure game – www.astoundme.com/scottadams
- alternative(s) – submit a proposal

In effect, students will be encouraged to follow their own ideas, in for example implementing a game using web technology, or a deep exploration in style, corresponding with (parallel) course(s) in design.

reference(s) There are many books dealing in one way or another with **web technology**. In particular the **Programmer to Programmer** series of Wrox (www.wrox.com) is highly recommended, especially for learning specific technologies in a practical way.

1. JavaScript: The Definitive Guide by David Flanagan – (amazon)
2. CSS: The Definitive Guide by Eric Meyer
3. Professional Web 2.0 Programming (Wrox Professional Guides) by Eric van der Vlist, Danny Ayers, Erik Bruchez, Joe Fawcett, Alessandro Vernet
4. business model(s) – www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
5. Webbots, Spiders, and Screen Scrapers: A Guide to Developing Internet Agents with PHP/CURL by Michael Schrenk
6. The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws, by Dafydd Stuttard amd Marcus Pinto
7. A. Eliëns, topical media & game development – media.eliens.net

A wealth of material and references can be found at my **topical media & game development** site, including tutorials and examples.

NM1 – resource(s) / inspiration(s) inspiration(s) / resource(s)

- UI design? – www.useit.com/papers/anti-mac.html
- html5/game(s) – www.intermediaware.com/blog/1141/a-html5-game-written-in-4-hours

- video – praegnanz.de/html5video
- network(s) – www.disruptivetelephony.com/2010/12/understanding-todays-skype-outage-explaining-supernodes.html
- font(s) – www.zdnet.com/blog/open-source/free-fonts-forever/7995
- jquery – savedelete.com/75-most-useful-jquery-plugins-of-year-2010.html
- (do not) read – blog.reybango.com/2010/12/15/what-to-read-to-get-up-to-speed-in-javascript
- javascript & PHP – phpadvent.org/2010/javascript-for-php-developers-by-stoyan-stefanov
- javascript – dailyjs.com/2010/12/02/framework-review
- CSS reset(s) – sixrevisions.com/css/the-history-of-css-resets
- XSS – en.wikipedia.org/wiki/Cross-site_scripting
- (js) book canvas – www.timenolthof.nl/projects/bookcanvas
- javascript/xml – www.ajax.org
- future(s) – chrome.blogspot.com/2010/09/back-to-future-two-years-of-google.html
- semantic markup – msdn.microsoft.com/en-us/scriptjunkie/ee730276.aspx
- experiment(s) – www.chromeexperiments.com/arcadefire
- (html5) element(s) – joshduck.com/periodic-table.html
- zombie(s) – sixrevisions.com/web-technology/the-webs-undead
- mentor(s) – blogs.techrepublic.com.com/five-tips/?p=212
- javascript – developer.mozilla.org/en/a_re-introduction_to_javascript
- experiment(s) – chrome.blogspot.com/2010/07/100-chrome-experiments-and-counting.html
- myth(s) – uxmyths.com
- style(s) – blogs.bnet.com/smb/?p=1009&tag=nl.e713
- women – www.smashingmagazine.com/2010/05/11/women-in-web-design-group-interview
- learn – smart.fm
- browser(s) – www.armorsurf.com
- standard(s) – www.indelv.com
- javascript game(s) – www.eyehook.com/games / www.javascriptgaming.com
- must read – www.smashingmagazine.com/2009/10/05/mastering-css-coding-getting-started
- scalable page(s) – bullit(s) / www.alistapart.com/articles/sizematters
- javascript include(s) – webdeveloper.earthweb.com/repository/javascripts/2003/06/195171/jsinc.html
- js/cc grammar(s) — online / jscc.jmksf.com
- inheritance – www.crockford.com/javascript/inheritance.html
- cloud computing – myfreecloudhost.com/static/what_is_cloud_hosting
- history of the web – www.dejavu.org
- about google wave – blogs.zdnet.com/BTL/?p=25972&tag=nl.e539

- (do not) read tutorial(s) – PHP / XML
- kill flash – www.stevenwei.com/2010/01/31/the-best-way-for-adobe-to-save-flash-is-by-killing-it
- stream(s) – activitystrea.ms
- advanced javascript – www.smashingmagazine.com/2010/04/20/seven-javascript-things-i-wish-i-knew-much-earlier-in-my-career
- jquery – www.smashingmagazine.com/2010/04/27/45-useful-jquery-techniques-and-plugins
- startup(s) – blog.wepay.com/2010/04/27/6-things-a-non-engineer-should-know-before-founding-a-web-startup
- html5 – www.tbray.org/ongoing/When/201x/2010/05/05/HTML5-and-the-Web
- chrome – tutorialzine.com/2010/06/making-first-chrome-extension
- script(s) – msdn.microsoft.com/en-us/scriptjunkie
- local news – www.zdnet.com/blog/open-source/the-big-open-source-news-opportunity/6707
- session(s) – css.dzone.com/articles/all-you-have-know-about
- browsing – www.conkeror.org
- HTML5 – www.smashingmagazine.com/2010/09/23/html5-the-facts-and-the-myths
- demo(s) – html5demos.com
- CSS – webdesignernotebook.com/css/are-css-frameworks-evil
- IDE js – www.cloud9ide.com
- CSS column(s) – www.kmsm.ca/2010/an-almost-complete-guide-to-css3-multi-column-layouts
- history – diveintohtml5.org/past.html
- information – theaporetic.com/?p=228