

# ~~NM1~~: ~~web technology~~

6700010 / project-based course, semester 1, 3 ects

*Æliens*

## course description – nm1: web technology

The course description(s) are taken from the accreditation report Creative Technology (version 2.0)

**content(s)** The course presents elementary web technology, primarily focussed on the use of HTML, XML, CSS, and Javascript, needed for authoring dynamic web pages.. Recommended literature: JavaScript: The Definitive Guide by David Flanagan

Online reference(s):

- [www.w3schools.com/](http://www.w3schools.com/)

**prerequisite(s)**: CS1 – computer & network architecture(s)

### goal(s) & attainment target(s)

The course aims at providing

- awareness of web 2.0 business model(s)
- familiarity the complex computational infrastructure provided by the web platform
- fluency with (X)HTML, CSS styling, javascript and forms
- full literacy with authoring dynamic web pages

Students are expected to have a sufficient degree of curiosity, and will be stimulated to develop their skills in the actual production of playful dynamic web pages.

**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).. In relation to DE-courses, the focus of NM-courses Is primarily on technical issues and programmatic authoring.

**application area & motivating example(s)** Dynamic web technology is used in a variety of ways, for example in simple online games, advertisement, and online shopping sites. Another interesting domain is web art, from which students may get inspiration for more advanced applications..

**teaching method(s)** The course will be organised around lectures, which will introduce basic examples and which will provide an in-depth explanation of the technologies. The assignments will consist of a series of basic exercises and a final exercise in which the students are required to develop a moderately complex dynamic web application.

Regular feedback will be given in classroom sessions where students present their work as well as via online comments or email. Grading will be based on basic assignments, the final assignment 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.

**special facilities:** computer lab & presentation facilities

## 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 relevant 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 increasing 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 target(s)** The NM1 course is meant to bring competence(s) and skill(s) at various 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, craftsmanship, 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 profit from the course, must have a sufficient degree of curiosity and lust for discovery.

**lesson material(s)** Although there are many good books available, there 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](http://www.w3schools.com) / kit(s)
- challenging target(s) – heart(s) / [labs.google.com](http://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 assignment(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 differ 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 be a small number of assignments, 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 assignments (may) include:

*basic(s) – web technology*

1. style – adapt three basic examples 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 individually, 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](http://www.digifit.eu)
- collection of javascript math games – [www.cut-the-knot.com](http://www.cut-the-knot.com)
- javascript (visual) adventure game – [www.astoundme.com/scottadams](http://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**.

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](http://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 and Marcus Pinto
7. A. Eliëns, topical media & game development – [media.eliens.net](http://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, that is massive, number of online resources, about web technology, including treatments about technology, solutions to HCI issues, style and semantics.

- topic(s) – [www.digital-web.com/topics](http://www.digital-web.com/topics)
- tutorial(s) – [tizag.com](http://tizag.com) / [www.w3schools.com](http://www.w3schools.com)
- hci – [www.digital-web.com/articles/the\\_pinballEffect](http://www.digital-web.com/articles/the_pinballEffect)
- style – [www.csszengarden.com](http://www.csszengarden.com) / [webdesignfromscratch.com/web-2.0-design-style-guide.php](http://webdesignfromscratch.com/web-2.0-design-style-guide.php)
- game(s) – [nl.youtube.com/experiencewii](http://nl.youtube.com/experiencewii)
- semantic(s) – [www.thefutureoftheweb.com/blog/writing-semantic-html](http://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](http://www.aptana.com) / [firefox addon\(s\)](#)
- example(s) – [javascript.internet.com](http://javascript.internet.com) / [openjsan.org](http://openjsan.org) / [code.google.com/apis/ajax/playground](http://code.google.com/apis/ajax/playground)
- ajax – [softwareas.com/ajaxjavascript-8-ways-to-create-graphics-on-the-fly](http://softwareas.com/ajaxjavascript-8-ways-to-create-graphics-on-the-fly)
- graphic(s) – [raphaeljs.com/reference.html](http://raphaeljs.com/reference.html) / [me.eae.net/archive/2005/12/29/canvas-in-ie](http://me.eae.net/archive/2005/12/29/canvas-in-ie)
- css – [nubyonrails.com/pages/css\\_graphs](http://nubyonrails.com/pages/css_graphs) / [codepunk.hardwar.org.uk/css2js.htm](http://codepunk.hardwar.org.uk/css2js.htm)
- javascript – [tool-man.org](http://tool-man.org)
- dynamic(s) – [www.hunlock.com/blogs/Howto\\_Dynamically\\_Insert\\_Javascript\\_And\\_CSS](http://www.hunlock.com/blogs/Howto_Dynamically_Insert_Javascript_And_CSS)
- menu(s) – [www.noupe.com/css/13-awesome-java-script-css-menu.html](http://www.noupe.com/css/13-awesome-java-script-css-menu.html)
- processing – [ejohn.org/blog/processingjs](http://ejohn.org/blog/processingjs)
- physic(s) – [box2d-js.sourceforge.net](http://box2d-js.sourceforge.net)
- adventure(s) – [zproxy.wordpress.com/2007/11/11/javascript-2d-adventure-game-demo](http://zproxy.wordpress.com/2007/11/11/javascript-2d-adventure-game-demo)
- dream(s) – [jsc.sourceforge.net](http://jsc.sourceforge.net)
- server(s) – [www.softintegration.com/webservices](http://www.softintegration.com/webservices)

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

health

- run – [www.runnersworld.com](http://www.runnersworld.com)
- nike+ipod – [nikeplus.nike.com/nikeplus/?locale=euen\\_eu](http://nikeplus.nike.com/nikeplus/?locale=euen_eu)
- sport – [www.digifit.eu](http://www.digifit.eu)
- moral(s) – [morale.erikbenson.com/person/erik](http://morale.erikbenson.com/person/erik)

For additional information look at [create/resource-web.html](http://create/resource-web.html).

## advice for the student(s)

many of you already have experience with tools and even programming web applications. Keep in mind though that **html** is not a programming language. Both **javascript** and **php** are, in effect extremely powerful, programming languages. Although **copy&paste** is an acceptable way of creating applications, make the effort to understand what is going on, or at least be aware of what you **do not know**, that is where you rely on the functionality of a give library or tool.

In the course, you are encouraged to follow an **exploratory approach**, meaning that you can go along by discovering what the technology can do for you, step by step. Try to keep **focus** though, by selecting both for the basic and final assignment(s) a **challenge** that suits you. This may be a game, or related to a topic that interests you, for example music or health, or why not, even romance. Remember, though, you have to take in a loth of **math** in your curriculum. One way to cope with this is by applying **learning by teaching**, that is developing simple **math games**, not only as a means to challenge and teach others, but to improve your own **understanding!**

## hint(s) for the instructor(s)

Although there is a wealth of examples online, as well as many tutorials, it is important to have a small collection of working examples, to show as what has been referred to as **canonical examples**, that show some of the complexity of **web applications** in a compact, easy to understand way. Some of the simple games are suitable for that, but also the examples producing visual effects, such as color changes or moving page elemnts.

More important than learning specific rules of style and content layout, is to make students familiar with the **dynamics** of web applications, and the programming model underlying these dynamics. Hands-on experience is crucial here. Even for advanced programmers, the **web** as an **application platform** is too complex to understand in its entirety. Some probing with technology, pushing the limits of your skills is essential in gaining a deeper understanding of the mechanics of web-applications, an understanding which is needed to ultimately create new **web services**, combining the real and virtual world(s), as expected from the student in later **creative applications**.