

INFS3202/INFS7202 Practical 2 – Interactive Web Design

The goal of this practical is to explore client-side Javascript. This Practical counts 5% towards your assessment. You must present this practical to your lab tutor during your scheduled lab sessions in week 5 (week starting 29/03/2010).

This practical builds on the material you have been introduced to in this course:

- In your Lectures Week 1-3 and in Practical 1, you were introduced to building HTML/XHTML and CSS web pages.
- In lecture Week 4, you were introduced to Javascript to enable you to enabling coding advanced actions when visitors visit your pages. You were also introduced to the important skill of making use of existing Javascript frameworks and libraries. Javascript programming, like any complex programming, requires the skill of understanding existing libraries and building on top of them.
- In this Practical, you will learn to use a few Javascript libraries to enabling interesting web site features. You will also learn to explore important aspects of Javascript.

This practical is divided into 4 Tasks:

- Building a web image gallery (1.5 marks)
- Building an interactive menu on a web page (1.5 marks)
- Understanding Javascript client-side issues (1 mark)
- Detecting browsers and coding for browser behaviour (1 mark)

Preparation

Before attempting this practical you should have a good working knowledge of HTML/XHTML & CSS. You should also be familiar with the concept of Web client-side technology, and how Javascript relates to HTML/XHTML and CSS.

Before attempting this practical please ensure:

- You have covered the material in Lectures 1 - 4.
- You have reviewed the work you did in Practical 1.
- You have tried the introductory Javascript code given in lectures in Week 4, including:
 - Creating an HTML page containing embedded Javascript and viewing it on a browser.
 - Creating a separate Javascript file, linking to it in a HTML page, and viewing the HTML page in a browser.
 - Tried and understand the test code provided that makes use of Javascript frameworks and libraries *Prototype*, *Scriptaculous* and *Lightbox*.
- You understand how to view Javascript errors in browsers of your choice, so you can do debugging. Below the settings for some common browsers:
 - In Mozilla browsers, including Firefox, you can view Javascript errors through the error console, accessible through the menu item Tools>Error Console , or Tools>Javascript_Console for older browser versions.

- In Microsoft Internet Explorer, turn on error reporting through Tools->Internet Options...->Advanced>Browsing, unclick "Disable Script Debugging (Internet Explorer)". Errors will show in a dialog box when the browser encounters them.
- In Opera, the errors are displayed in the console accessible through menu item Tools>Advanced>Error Console or Tools > Advanced > JavaScript console.
- If you use a browser different from the above, look up your browser's documentation to determine how it displays Javascript errors.
- You have a basic understanding of the the Javascript language constructs. A good introductory tutorial is available at:
 - <http://www.w3schools.com/js/default.asp> (complete at least from *JS Introduction* to *JS Guidelines*)
- Additionally, have a Javascript online reference ready to look up details of the language constructs:
 - https://developer.mozilla.org/en/Core_JavaScript_1.5_Guide
 - <http://www.w3schools.com/jsref/default.asp>, for Javascript objects.

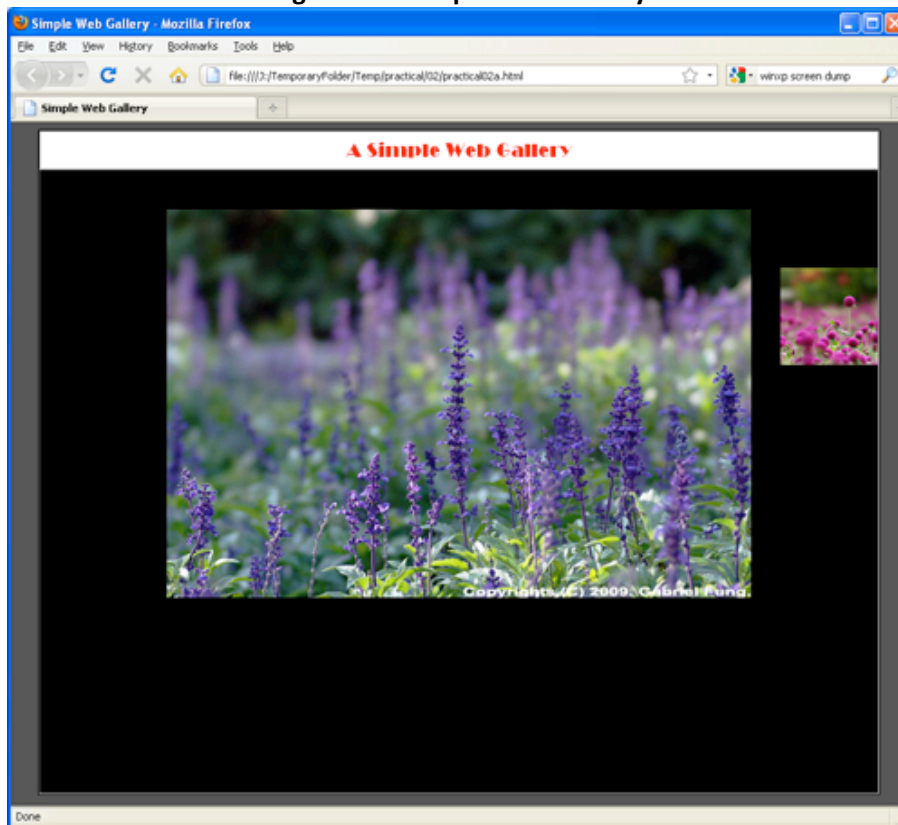
Task 1 – An Image Gallery (1.5 Mark)

In your lectures, you were introduced to the Javascript frameworks and libraries *Prototype* and *Scriptaculous*. In this task, you will be required to build a web image gallery using a third-party library *FrogJS* that is built on top of *Prototype* and library *Scriptaculous*.

Requirements

1. Follow the instructions on the source on downloading and using the required scripts:
 - <http://www.dynamicdrive.com/dynamicindex4/frogjs/index.htm>
2. Create a simple web gallery with the following specification:
 - The title of the gallery is “Simple Web Gallery”.
 - The gallery contains at least 3 images of your choice. A visitor to the gallery can view and navigate through each of the images. The gallery should be based on the navigation provided by the *FrogJS* library and look similar to Figure 1.

Figure 1: A Simple Web Gallery



Task 2 – An Interactive Menu (1.5 Mark)

In this task, you will construct an interactive web page menu using a third-party Javascript library called *Accordion*, that is build on top of another popular Javascript library *JQuery*. You will be required to create a menu that contains the same elements as the ITEE School’s homepage (<http://www.itee.uq.edu.au/>), but make the menu interactive instead of static.

Requirements

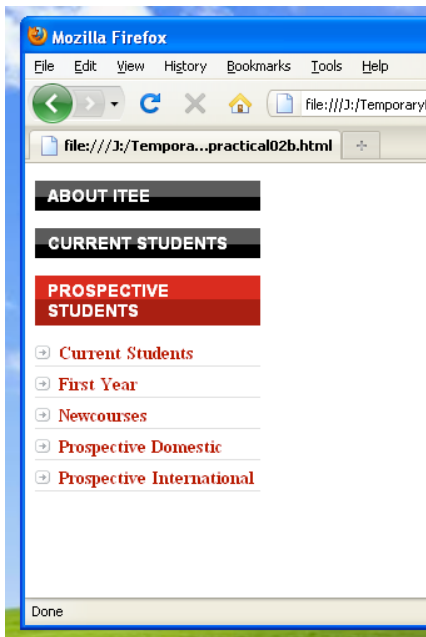
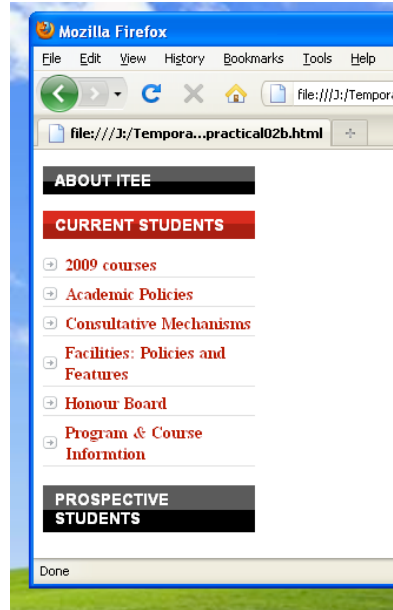
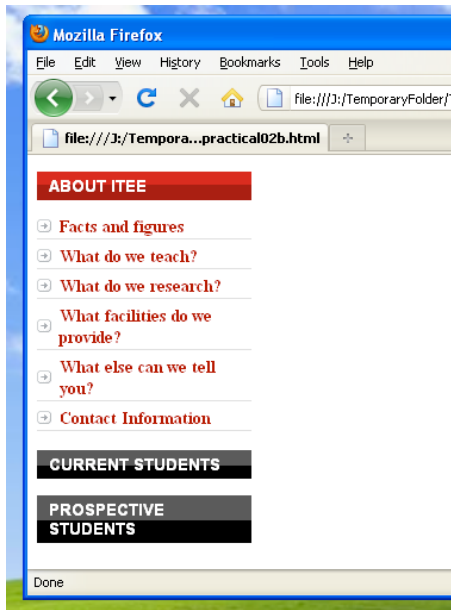
1. Follow the instructions on the Accordion library’s source on downloading and using the scripts:
 - <http://www.dynamicdrive.com/dynamicindex17/ddaccordionmenu-bullet.htm>
2. Create an interactive menu on a HTML/XHTML page the following specification:
 - The menu is interactive and contains 3 headings, with menu choices in each heading. The 3 headings and the choices within them are the same as the first 3 items in the ITEE School website main menu (“About ITEE”, “Current Students” and “Prospective Students”). When one of the headings is clicked, the choices for the other headings collapses, and the choices for that heading expands and the choices for the other headings collapses.
 - The expected output of the menu is shown in Figure 3.
 - You must make use of the *Accordion* library.

Figure 2: ITEE School Homepage



You will create a new menu system containing first 3 parts of this menu, but make it interactive rather than static.

Figure 3: Expected Output



Task 3 – Understanding Javascript client-side issues (1 Mark)

In this task, you will explore the issues involved in using client-side languages like Javascript to implement web site behaviour.

The following link is the front page of an online system with password protection:

<http://www.hiewenterprises.com.au/temp/pw1.html>

Requirements

1. Using your knowledge of HTML and Javascript, try to bypass the password protection and gain access to the codeword contained in the system. What is the system password, and what is the secret codeword? (0.5 Mark)
2. What mistake(s) did the developer of this system make in developing his system? What should the developer have done differently? (0.5 Mark)

Task 4 – Challenge Task: Detecting browsers and coding for browser behaviour (1 Mark)

One of the essential parts of coding in Javascript is handling *objects*. In this task, you will use Javascript to access the *Navigator* object, to detect the properties of your web page visitor's browser and manage the visitor experience accordingly.

Preparation:

1. Review and understand this tutorial page with examples of Javascript accessing the Navigator object: http://www.w3schools.com/js/js_browser.asp

Requirements:

1. Create a Javascript that displays **only** the navigator properties that are supported by the visitor's browser. The output should be similar to the last example script given in the preparation tutorial above (http://www.w3schools.com/js/tryit.asp?filename=try_dom_navigator) but **without** the lines where the properties are «undefined». (0.5 Mark)
2. Select two different browsers you have available on your computer. Modify your solution in Task 2 so that the colour of the headings are different for the two browsers. For example, one browser may show black-coloured headings while the other shows red-coloured headings. (0.5 Mark)