P3 Project Coding (50% of project grade)

Coding Guidelines

Time to build your website! This is where all the hard work put into the design phase pays off. Following your design, carefully assemble the project. Do this one piece at a time, testing it as you go. Be sure to set time aside in your schedule. Start early and work often. Sometimes the simplest task takes the most time to complete.

Here is some information that you will find useful during this phase:

Documentation

Documentation of your work is an important aspect of the development of the project. It will help others, who will possibly try, at some point, to extend or modify your site, to figure out what is done and how it is done. Documentation will also be a big help to you, the creators of the file, when, during the next phase - the debugging phase - you will come back to tackle existing problems.

You insert documentation through comments in all your files. We can distinguish the top-of-file comment from comments spread along the file. The top-of-file comment consists of a paragraph that gives information about the creator of the file, the time of its creation and the purpose of the file, as well as its modification history. Someone should be able to get a good idea of what a certain file is doing, and how it is doing it, just by looking at the top-of-file comment.

A normal, uncomplicated file would just have a comment like this at the top:

```
<!--
FILE NAME: research/index.html
WRITTEN BY:  Michail Giannakos
WHEN:  April 2015
PURPOSE:  describe the general research area and provide links to other sites
-->
```

Comments spread along the file are usually shorter, one line or a few lines long. Most often their purpose is to explain parts of the code that are not completely straightforward, for example code in Javascript or CSS rules that create certain effects in the page.

Dog Tags

Artists sign their work, and most of you will want to as well. Therefore, you may want to put, in a small font, your names and maybe the year, at the bottom of the page. These are called "dog tags," by analogy with the name and address information hung around the necks of dogs, cats, and soldiers.

Using Other People's Code

We teach you a lot, but only a small fraction of the many things you can do with HTML, CSS and JavaScript. We love it when students go beyond what we learn in class, in order to add cool features to their websites, so we encourage you to do so. In addition to the resources on the IT2805 website, there are many websites out there with tricks and tips, tutorials and code to borrow and use. We'll call this **outside code**. Using those is fine, but there are some bounds on this:

- Always give credit to the source. You can do this just with a comment in your HTML, CSS or JavaScript, as appropriate, giving the URL of the website you got the idea, trick or code from.
- Understand the code. Don't just mindlessly copy/paste and cross your fingers that it will work. Take the time to understand what the code is doing and how it works. Simplify it if necessary. Remove extra features that you don't understand. Ask for help in understanding their code; we'll be glad to help. If we were to ask you about any code that you submit as part of your project, you should be able to comfortably explain it to us. If you can't, maybe you should drop that feature.

Our point here is not to scare you. We do want to your expand your capabilities, challenge yourself, build amazing things and have fun doing it.

Browser Compatibility

For the purposes of the course, your pages must display properly using the lab's browsers and versions.

Project Status

Midway through your project coding, we'd like a brief status report. If you think we're trying to check up on you, you're right; we are, but in a good way. If everything is going fine and you're on schedule, you need only say so. If problems have arisen and you've fallen behind schedule, let us know, and tell us how you're planning to remedy the problem.

This is not a graded component, and there will be no penalty if you report that you've fallen behind. However, the coding phase is graded, and so it's to your advantage to let us know if there are problems while there's still time to correct the problems. We will be far more upset if we don't find out that there are problems until just before the coding is due, when there's no time to fix it.

This is also a good time to get together with your partners (if you haven't already), to check on how each other is doing. Some groups work very autonomously, which can be good, but it can work out badly if one team member has fallen behind and her partner doesn't know it. Some of the biggest challenges of doing a big project in a group are coordinating with your partner and staying on schedule; learning how to do this is an important part of why we use group projects in this course.

How to report your status

Email your advisor with a few sentences or paragraphs, saying whether you're meeting the schedule as you put it in your design document. If all is well, this need not be more than a sentence saying so.

If your plans have gone awry, as they so often do, say so and say what you'll do to get back on track. In particular, say what isn't done, and a schedule for getting it done by the coding deadline.

Please also remind us (and/or update us if you've changed this since your design document), what your four required JavaScript applications are, and who is responsible for each. We want to check and make sure that you've chosen applications that meet the criteria (if you haven't, this gives you a chance to fix that before your coding is complete). Now that you've learned more JavaScript, you might want to replace your JavaScript applications with something better; this is a good time to talk to your advisor about that. We only need one email per team, but please cc your partner on the email so that we know all of you are in agreement with the status.

Project Check-up with Advisor

Often, students make simple mistakes in the coding phase of the project that could be easily remedied if they only noticed them. So, just before the coding deadline, you should review your code for errors. Here's how to avoid some common ones that we've seen in the past:

- Pages validate (HTML and CSS)
- No code errors (test each page with the Console)
- Clear navigation structure
- All features work correctly
- No broken links
- Appropriate comments (must be at top of each page, but also within CSS, JavaScript & HTML)
- Proper nesting of code (CSS, HTML, and JavaScript)
- No unnecessary duplication of code from page to page (use JavaScript to avoid this)
- Appropriate ALT attributes for all images
- All pages have appropriate titles
- Aesthetically pleasing layout/color/font
- Properly sized images with correct aspect ratio

You can, of course, check these on your own and help one another, but sometimes a more experienced pair of eyes can help. We are offering the opportunity for a checkup: a quick look-over of your site with your project advisor. In the labs after Lecture 13, your advisor will meet with you and check project and potential problems.

WARNING

This checkup is not intended to be a guarantee of a perfect grade! It is simply a way to help you identify some common defects. Of course, there is more to a great site than just avoiding defects, just as a paper without typos and spelling errors isn't necessarily a great paper. Still, we hope you'll find it valuable to have this checklist and the opportunity for a checkup.

What to Submit

Gather together all the files and folders of your website and place them into a single folder, upload this folder to your web-account in addition zip and submit this on its learning, grading will happen

based on the zipped file, but uploading the is mandatory. Make sure you do this by the deadline given in the schedule.

You must also submit an accompanying HTML document named `P3_changes.html`. This document should detail and justify all substantive changes from your design, so that we won't think it was just a mistake or oversight. Although your site should match the design in most details, we understand that you're still learning web design, so you will probably discover that you'll want to change some of your design decisions by the time you get to the coding phase. Similarly, your client may make some changes during this time. Your document need not be as long and elaborate as your design document, but it should clearly describe what changed.

What specifically should you include in your changes document? Your `P3_changes.html` file should contain a **LINK** to the start page for your website (you'll have many many files in your directory, we need to know where to begin touring your site).

The `P3_changes.html` file should also include:

1. How each of the team members fulfilled their minimum requirements. For example:

   Rachel fulfilled her JavaScript requirements with a user-defined function on intro.html and rollovers on members.html. Tom validated form inputs on orform.html, and implemented the pull-down menu for each page

2. Explicitly state (since it may have changed since the design document), which person worked on which files, including CSS and JavaScript files, and who produced the artwork (banner and button design, etc.).
3. Indicate how the modularity requirement was handled (where and how either Server Side Includes (SSI) or JavaScript were used for modularity, and who was responsible for the implementation).

Your website, the `P3_changes.html` and the presentation documents are due by the coding deadline (Nov 16), and may **not** be changed after the due date.

Grading

Points will be awarded in the following categories:

- Basic site function
- Aesthetics, design, and readability
- Usability, navigation, accessibility
- Documentation
- JavaScript minimum requirements met
- Modularity
- Adequate update (p3_changes)
- Excellency

Excellence

The top 10% (10 out of 100 points) of the coding grade is reserved for teams who demonstrate conspicuous excellence in their project. How can you get those extra points? You must, somehow, go **above and beyond** ordinary achievement. There are many ways, but there is no checklist, any more than there is a checklist for great writing, great science or great design. An excellent website may have clever coding (e.g. Javascript code that you wrote on your own, beyond examples seen in class or taken from the Web); extraordinary beauty and harmony; unusual scale or some combination of all these. It is our expectation that we will award relatively few of these extra points - if you get any of these points, you should be very proud of yourself.

Example

Here is an example of a P3 changes document:

http://www.idi.ntnu.no/~michailg/IT2805/exampleproject/p3_changesexample.html