**Syllabus**

**Tandon School of Engineering of New York University**

**Technology, Culture and Society**

**Integrated Digital Media**

**DM-UY 2193 Intro to Web Development**

**Fall 2018 – Professor Sarah Dahnke**

*Tuesday + Thursday 2:30 - 4:20, 2 MetroTech Center Room 813*

*Link to class Slack workspace:*[*https://idmnyu.slack.com/*](https://idmnyu.slack.com/)*(Students will receive an invitation to this course's private channel.)*

*To contact professor: on Slack @sarah, email*[*srd280@nyu.edu*](mailto:srd280@nyu.edu)

*NOTE: The preferred communication tool for this course is Slack. You may direct message me short questions and/or code snippets. If you find yourself needing to write long paragraphs, that is an indication you should be setting up an appointment to meet. Email should be used in instances where you may need to cc someone outside of our class or administrative NYU business related to grades, etc.*

*Office hours are by appointment on Tuesdays or Thursdays before 12:20 or after 4:20. Please contact me at least 24 hours in advance to set up a time to meet.*

**A note about this syllabus**

This is a living document, as are the materials in this course repository. There are certain aspects of our course, such as our calendar or grading criteria, that will not differ from this syllabus unless there are extenuating circumstances. However, the pacing of this course may vary depending on student need, comprehension and technologies that are sometimes rapidly changing in the midst of our learning process. I will do everything in my power to update you on any changes ASAP (in class and via Slack) and to stay fully transparent about what you are expected to learn, produce, and share. Please engage in open communication, as I will do the same.

**Course Pre-requisites**

Basic computer knowledge. Basic familiarity with any programming/scripting language is preferred but no required.

**Course Description**

In this course, students focus on client-side programming. Assignments are arranged in sequence to enable the production of a website of professional quality in design and production. This studio stresses interactivity, usability, and the quality and appropriateness of look and feel.

This course will provide a basic understanding of the methods and techniques of developing a simple to moderately complex web site. Using the current standard web page language, students will be instructed on creating and maintaining a simple web site using HTML5, CSS3, JavaScript and various external libraries.

**Program Learning Objectives**

**Students will:**

1 – Students will develop conceptual thinking skills to generate ideas and content in order to solve problems or create opportunities. Students will develop a research and studio practice through inquiry and iteration.

2 – Students will develop technical skills to realize their ideas. Students will understand and utilize tools and technology, while adapting to constantly changing technological paradigms by learning how to learn. Students will be able to integrate/interface different technologies within a technological ecosystem.

3 – Students will develop critical thinking skills that will allow them to analyze and position their work within cultural, historic, aesthetic, economic, and technological contexts.

4 – Students will gain knowledge of professional practices and organizations by developing their verbal, visual, and written communication for documentation and presentation, exhibition and promotion, networking, and career preparation.

5 – Students will develop collaboration skills to actively and effectively work in a team or group.

**Student Learning Outcomes**

By the end of the course, students will be able to:

1 - Design, build, and develop content for a professional-quality website.

2 - Understand and implement the iterative process

3 - Learn how to proactively learn, also known as self-regulated learning. In any learning situation, you should study beforehand, make/do, debug, reflect, adjust, and do it all over again (iteration). Learning happens in a cycle.

4 - Create an internal developer / creative community

**Course Objectives\*\***

* User Interface (UI) / User Experience (UX)
* HTML5 / CSS3
* HTML/CSS/SASS Frameworks: Bootstrap, Materialize, etc.
* The Responsive Web (flexible media & media queries)
* Javascript/Javascript Libraries

**SOFTWARE REQUIREMENTS**

* Browser: Chrome or Firefox
* Github Account: github.com/join
* Developer Tools for Chrome or Firefox
  + Firebug for Firefox, [http://getfirebug.com](http://getfirebug.com/) or
  + Chrome Developer Tools, <https://developers.google.com/chrome-developer-tools>
* HTML Text Editor (Atom, Sublime Text, Brackets.io, etc)
* FTP Application (Cyberduck, Transmit, Fetch, FileZilla, etc)
* Web Server Space
  + I highly recommend that you get your own web server space from an internet service provider. One recommendation is <https://www.siteground.com/>

**Course Structure**

Class time will be spent as a combination of lecture, review, and in class exercises. Homework will consist of online tutorials as well as weekly projects that build on each week’s topics of discussion.

**Readings**

There is no required textbook for this course, however students will be required to complete various online tutorials between classes.

**Laptop Usage During Class**

I love having WiFi available and I think it is a valuable tool for use in the classroom. Unfortunately, it can very easily be a distraction as well. Laptop use will be expected in class but should be focused on class-specific material, applications, and research. Whenever classmates or guests are presenting please keep your laptop closed. The quality of the class depends in large part on your attention and active participation, so please respect your fellow classmates and close your lid.

**Grading**

NOTE: While you are expected to document and host your work on Github or another external server, links to assignments, projects, and documentation will be submitted and graded on NYU Classes. This will allow me a space to privately communicate any notes and point deductions connected to each student's assignment. It will also allow you to track your progress in the class. After midterm and final presentations, grades from the NYU Classes gradebook will be migrated to Albert.

Grades will be determined according to the following breakdown:

**Regular Assignments 25%**

We will have weekly assignments that are relevant to material from the previous class. When it is assigned, each assignment will contain a breakdown of how many points it is worth and the grading criteria. Most assignments will include a Learning Log, where you will be asked to provide some assessment of your clarity on a topic after doing a tutorial. Please refer to the Learning Logs file in this repository for more instructions. Late assignments will lose a point for each day they are late.

**Mid-Term Project 20%**

The midterm assignment will be a project that demonstrates a working knowledge of HTML and CSS elements. This project must be completed, published, and presented.

**Participation and Attendance 20%**

Attendance is *mandatory*. Please inform me via Slack direct message if you are going to miss a class. Showing up late for class or an excessive number of absences will adversely affect your grade. This class will be participatory, you are expected to participate in discussions and give feedback to other students both in class and participate with their projects. This (along with attendance) is 20% of your overall grade. Each student will be asked to do a 5-min presentation on a Learning Log throughout the semester, which is part of your participation grade.

**PLEASE NOTE: 3 unexcused absences will result in your grade dropping 5%**

**Final Project 20%**

Class will culminate with final projects. It is expected that these will be both technology and content driven. The final project will be built over the course of several weeks. This project must be completed, published, and presented.

**Documentation/Self-Asssessments/Tickets to Leave 15%**

5% - Students will be expected to document their work on Github. You may also set up a blog on a site like Wordpress or Medium for supplementary narrative documentation. If you already have your own hosted website and would like to blog there, that works, too.

You can receive web server space from IDM’s Technology Manager, Elton Kwok, MAGNET 883 IDM’s FTP server info: [http://sites.bxmc.poly.edu](http://sites.bxmc.poly.edu/) (Use active mode)

5% - At the end of each session, students will need to submit a Ticket to Leave. Please view the Ticket to Leave file in this repo for specific instructions.

5% - Students will perform three self assessments on their work throughout the semester. Specific instructions and deadlines on these assessments will be provided when they are assigned.

**Qualitative Grading Overview**

Each student will be judged on the quality, experimentation, and improvement that their work shows.

**A. Excellent (90-100)**

Performance, participation, and attendance of the student has been of the highest level, showing sustained excellence in meeting course responsibilities. Work clearly differentiates itself from other work, has memorable impact, pursues concepts and techniques above and beyond what is discussed in class. The student thoroughly understands the web design and development process.

**B. Very Good / Good (80-89)**

Performance, participation, and attendance of the student has been good, though not of the highest level. Work demonstrates a better than average web design and development process.

**C. Satisfactory (70-79)**

Performance and attendance of the student has been adequate, satisfactorily meeting the course requirements. Work is average and competent, showing a basic understanding of the web design and development process.

**D. Poor; Below Average (60-69)**

Performance and attendance of the student has been less than adequate. Work is lacking in many or most areas that show any understanding of visual foundation. Problems may include lack of interest, procrastination, poor planning and poor craft.

**F. Unacceptable (59 & Below)**

Performance and attendance of the student has been such that course requirements have not been met. Work shows no overall understanding of the course material on many levels or either a severe lack of interest.

**Accommodations**

If you are student with a disability who is requesting accommodations, please contact New York University’s Moses Center for Students with Disabilities at 1-212-998-4980 or [mosescsd@nyu.edu](mailto:mosescsd@nyu.edu). You must be registered with CSD to receive accommodations. Information about the Moses Center can be found atwww.nyu.edu/csd. The Moses Center is located at 726 Broadway on the 2nd floor.

**Notable Dates**

You can find the official academic calendar [here](https://www.nyu.edu/registrar/calendars/university-academic-calendar.html#1184)

**Monday, September 17, 2018** - Last day to drop a class and receive a refund of 100% of tuition & fees for Undergraduate, Graduate and Diploma students who are dropping classes, but will remain enrolled in at least one course. For Undergraduate, Graduate and Diploma students who 'Completely Withdraw' from ALL courses during the semester, please see the Refund Schedule for Complete Withdrawal. Last day to drop Fall 2018 classes and not receive a grade of 'W.' Last day of active waitlists. Last day to add/drop on Albert.

**Monday, October 8, 2018** - Fall Recess, no classes scheduled

**Tuesday, October 9, 2018** - NYU Legislative Day. Classes meet according to a Monday schedule. (This class will not meet.)

**Monday, November 5, 2018** - Last day to use the Term Withdrawal Form to submit a spring semester term withdrawal request or submit a request to withdraw from a course. Students should consult the academic calendar of their home school for specific deadlines pertaining to course withdrawal for the term

**Wednesday, November 21-Friday, November 23, 2018** - Thanksgiving Recess, no classes scheduled

**Friday, December 14** - Last day of Fall 2018 Classes

## Learning Logs

Students should have 10 Learning Logs by the end of the semester. Plan on doing one every week. Every student will also give at least one five-minute Learning Log presentation for the class.

Learning Logs can be around any client-side web topic we have covered in class where you need more clarity or want a deeper dive. I will update this file with suggestions as the course moves forward.

To submit a Learning Log:

Post to Slack in the logs channel for this course. Begin message with "#LL: descriptive name" (where descriptive name is a title that describes your learning log).

* Identify (name, URL, sections, etc.) at least one tutorial that you did, and respond to the following.
* Instances where you achieved clarity, i.e. understood something or felt pieces of knowledge falling into place towards a greater understanding.
* Instances where you did NOT achieve clarity and why. What can you do to aid your understanding?
* Connections you made among the week’s readings, lectures, class activities, homework assignments, etc.

(The tutorial you do could also be via a book—in that case, list the name of the book instead of the URL, and include the chapter and pages.)

Learning Logs Presentaton schedule:

Oct 23: Sasha

## Weekly Schedule Overview

NOTE: Specific assignments and materials for each class session will be located in their own folder in this repository. This schedule is meant to serve as an overview of the course trajectory.

#### Sept 4 & 6: Hello World, Design Inspo presentations

#### Sept 11 & 13: Intro to HTML, Intro to UNIX/command line & Git/Github, File paths/Directories, Github/Github Pages, Intro to CSS

#### Sept 18 & 20: More CSS, Wireframing, Layout, Image Format, Google Fonts

#### Sept 25 & 27: CSS Transform/Transition/Animate, Media Queries and Responsive Design, Project Moodboards

#### Oct 2 & 4: Midterm Project Proposals, Oct 4 special session on accessibility with Claire Kearney-Volpe

#### NO CLASS OCT 9

#### Oct 11: Flexbox, CSS Grid, Bootstrap

#### Oct 16 & 18: Midterm Project Presentations

#### Oct 23 & 25: Javascript variables, arrays, functions, conditionals

#### Oct 30 & Nov 1: JS math, events, loops, DOM manipulation

#### Nov 6 & 8: JS Objects

#### Nov 13 & 15: JS Libraries, Jquery

#### Nov 20: Continued Jquery

#### NO CLASS NOV 22

#### Nov 27 & 29: Final Project Proposals, student request topics

#### Dec 4 & 6: Final Project Workshop

#### Dec 11 & 13: Final Project Presentations

### STUDENT RESPONSIBILITIES

* Schedule your time (Keep a calendar of some sort.)
* Come to class on time and participate (Be present and engaged.)
* Study outside of class (ideally with classmate(s) i.e. Form a study group or team), fulfilling homework assignments, reading, and studying concepts covered in class.
* **Back up your work constantly** (That's what Github is for!)
* Consult Slack and Github at least once a week for up to date info
* Complete all assignments by due dates

### STUDENT VALUES

* Action – do your absolute best
* Strive for continuous improvement
* Pay attention to detail & craft
* Have desire amounting to enthusiasm (to learn, to explore)
* Have self-motivation, proactiveness, and focus
* Have patience, persistence, and discipline
* Be creative
* Have self-confidence and pride in your work
* Have fun!

## Tips for Progress

* Get ready to be frustrated. This is an integral part of learning. Everyone makes mistakes; you will too. It is ok.
* **Keep calm and focus your study.**
* Add one thing at a time to your code until you know what the code is doing.
* Don’t give up when something gets hard. Keep at it.
* Read and/or watch everything you can. (Multiple resources on the same topic give you different perspectives). Watch and learn from the mistakes you make. These are prime ways to improve your skills. You will continue to learn even if you have been at it for a long time.
* View Source. Look at other people’s code and ways that they organized it. How easy is other’s code to read through?
* Learn Relative Paths.
* Acquire and Use a HTML, CSS, Framework tag list or cheat sheet.
* Make sure you have closed tags and closed quotes.
* Make sure you have spelled tags correctly
* index.html in each folder (except images folder). why?
  + Server can be setup so that this is the default page within a directory
  + Shorter/Cleaner URL
  + Don’t see all of the files within a directory
* .html extension
* **Lowercase filenames with no spaces**
* Be consistent in the style that you write your code.
* Title (descriptive text descriptions for browser bookmarking)
* Cross Browser and Device Compatibility
  + Test on as many browsers and devices as possible.
* Validate your code: [HTML5 Validator](http://validator.w3.org/) and [CSS Validator](http://jigsaw.w3.org/css-validator)
* Comment your code!
* Make sure your links work. Test! Test! Test!

## TICKET TO LEAVE

At the end of each class period, before you leave:

Post to Slack in the class channel.

Begin message with "#TTL".

Respond to 1 or both of the following:

* List at least 1 thing you learned today in class.
* What concept(s) confuses you from today’s lecture, discussion, or exercises? (This will give you a clue as to what you should study before the next class.)

## Basic Unix Commands

cd < directory name >

Change directory (e.g. cd pictures brings you to the sub-directory “pictures”)

cd ..

Change directory up one level to the parent directory

cd

Return to the home directory

touch

Create a new file

chmod

Modifies the permissions on a file or directory

cp < current filename > < new file name >

Copy (e.g. cp cats1.jpg cats2.jpg makes a copy of the picture “cats1.jpg” and calls it “cats2.jpg”)

cp -i < current filename > < new file name >

Use cp -i to be prompted before over-writing another file. This is a safer method.

ls

Lists a directory of your files

ls -l

Lists a directory with more information about the files

ls -a

Lists a directory including the hidden files man This is the online Unix help documentation (as in “manual”). Type man + the command (e.g. man ls) for a detailed explanation of the command. Type q to leave the manual pages.

mkdir < directory name >

Creates a new directory (e.g. mkdir pictures creates a directory called “pictures”)

pico

Use pico < filename > to edit a text file

pwd

See the current directory and path

rm < filename >

Remove (or delete) a file (e.g. rm cats1.jpg deletes the file “cats1.jpg”)

rmdir < directory name >

Removes an empty directory (e.g. rmdir pictures removes a directory called “pictures” as long as that directory is empty)