The Living Web

by

Luke C. LeGrand

Department of Art, Art History, and Visual Studies
Duke University

Date: ______________________

Approved:

___________________________
Victoria Szabo, Chair

___________________________
Mark Olson, Advisor

___________________________
Paul Jaskot

Thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Arts in the Department of
Art, Art History, and Visual Studies in the Graduate School
of Duke University

2019
ABSTRACT

The Living Web

by

Luke C. LeGrand

Department of Art, Art History, and Visual Studies
Duke University

Date: _______________________

Approved:

___________________________
Victoria Szabo, Chair

___________________________
Mark Olson, Advisor

___________________________
Paul Jaskot

An abstract of a thesis submitted in partial fulfillment of the requirements for the degree of Master of Arts in the Department of Art, Art History, and Visual Studies in the Graduate School of Duke University

2019
Abstract

As the role of Internet Connected Technologies (ICTs) increases exponentially, and as all student populations (highly motivated or not) become increasingly composed of digital natives, it is imperative that the academy adapt to these new challenges. A university is obligated to ensure its students are adequately prepared for the Digital Age. This paper seeks to examine and evaluate the current scholarship of coding pedagogy, digital learning, and information science education, to leverage these evaluations towards the construction of a course of study which is informed by critical thought and current scholarship. It is the author’s hope to provide methods and approaches through which students who may lack an academic background in Computer Science can develop critical and analytic thinking skills alongside essential understandings of the technologies underpinning their daily lives.
Dedication

To the many smiling faces of Duke University,
who have made me feel so at home.
Contents

Abstract ........................................................................................................................................ iv

List of Tables ................................................................................................................................... ix

List of Figures .................................................................................................................................. x

Acknowledgements ........................................................................................................................ xi

1. Introduction ...................................................................................................................................... 1
   1.1 Project Topics .............................................................................................................................. 3
   1.1.1 The Pedagogy of Pedagogy ..................................................................................................... 4
   1.1.2 Web Coding Education .......................................................................................................... 5
   1.2 Goals for Development ............................................................................................................... 5
   1.2.1 Conceptual Courses ............................................................................................................... 6
   1.3 Research Goals ........................................................................................................................... 7

2. Pedagogy .......................................................................................................................................... 9
   2.1 Ethical Use .................................................................................................................................. 9
   2.2 Teaching problem solving ......................................................................................................... 10
   2.2.1 How to Build for Students ..................................................................................................... 12
   2.2.2 Coding Demonstration .......................................................................................................... 13
   2.3 Scalability .................................................................................................................................. 19
   2.3.1 Where is the classroom? ........................................................................................................ 19
   2.3.2 Who are the Students? ............................................................................................................ 20
   2.3.3 Inclusive classrooms .............................................................................................................. 21
5. Conclusion .................................................................................................................. 43

5.1 Looking Forward ..................................................................................................... 43

Appendix A: Syllabus ................................................................................................. 44

Appendix B: Annotated Syllabus ............................................................................... 56

Appendix C: General Code ........................................................................................ 107

Appendix D: Website Documentation ....................................................................... 109

Bibliography .................................................................................................................. 111
List of Tables

Table 1: Prospective Syllabus ..................................................................................................................... 51
Table 2: Annotated Syllabus.......................................................................................................................... 70
List of Figures

Figure 1: ISS-240 Lab #4 render at 400px screen-width (broken) .................................. 15
Figure 2: ISS-240 Lab #4 code at 400px screen-width .................................................... 16
Figure 3: ISS-240 Lab #4 code at default screen-width .................................................. 16
Figure 4: ISS-240 Lab #4 render at default screen-width ............................................... 17
Figure 5: ISS-240 Lab #4 code at 400px screen-width .................................................. 18
Figure 6: ISS-240 Lab #4 render at 400px screen-width (fixed) .................................... 18
Figure 7: Compilation of student free responses ............................................................. 22
Figure 8: Word cloud generated from student free responses ....................................... 24
Figure 9: Student selected answers to "Where did you go for help?" ................................ 25
Figure 10: Research->Test->Implement->Cite ................................................................. 27
Figure 11: Student Troubleshooting Guide ................................................................... 31
Figure 12: Student Boilerplate Example ......................................................................... 107
Figure 13: Student Annotation Exercise ........................................................................ 108
Figure 14: The Living Web Project Site ......................................................................... 109
Figure 15: Student Demo Site based on Syllabus ......................................................... 110
Acknowledgements

I would like to offer my most humble and sincere gratitude to Professors Victoria Szabo and Mark Olson for their enduring support, endless patience, and steady guidance. Also, to Sarah Riazati, Katie King, and Max Symuleski for sharing their ideas, experience, and experiences with me. Finally, to Josh Marmorstein for serving as my tireless sounding board.
1. Introduction

A university's purpose is to prepare students for the world of today, not to mention the world of tomorrow. Understanding the web, how it works, how to dissect it, and how to build onto it are invaluable skills for living much less working.

A student can, with minimal guidance, get a page up on the web in a couple of hours, and advance quickly from there. Further, the learning of HTML, CSS, and JavaScript offers ample opportunities to teach information architecture, data management, computer utilities, internet function, electrical engineering, and more.

Such a program could also introduce students to a wide variety of software platforms that can be useful in personal and professional contexts. The potential for development extends far beyond what can be covered here. Imagine a course designed to make efficient use of time tracking, time management, team working, wireframing, rendering, design, and digital media software that help students manage projects and encourage working in a self-motivated way. A course that empowers students to build diverse, forward-thinking career skills in high energy, social environment. A course informed by scholarship and organized with expertise. A course that focuses on the effective and efficient use of HTML, CSS, JavaScript, information architecture, data management, computer utilities and maintenance, internet function, electrical engineering, and other support technologies essential to any course of study. A course that promotes confidence and creativity in students who arrive in the lab woefully ill-educated about the devices and networks through which they live their very lives. This is an issue of computational literacy; the university has an ethical obligation to arm its students against misinformation, manipulation, and malfunction.
An additional benefit of such a course: it can be run cheaply at a huge scale, the resources are largely static, and those listed here are available to the public for free. This course can be offered anywhere the resources exist to host students with an internet connection.

The hope is to encourage students to view this as an art class rather than a science one—or to understand the fundamental connections between the two. Introducing computational concepts and fun, high energy projects into a social, collaborative learning environment; in the same spirit as hack-a-thons and boot camps which have proven to be highly effective settings for coding training.

Over the last few years the 200-level class in web development at Duke University, out of all students most have been students approaching graduation who find themselves in the lab to fulfill a general education requirement—and they are terrified. The threat of computer operation under direct scrutiny, not to mention the application of math and logic, fills most of the students with dread and despair. However, the most satisfying aspect of my instructional position is seeing students come in meekly on day one and before the second week is finished we are sharing grand celebrations of new successes in code production, design insight, confidence increases, and, most importantly, in the discovery and formulation of new metaphors for understanding the abstraction of code to design.

Indeed, every new trick brings excitement. Keyboard shortcuts, creating directories, data management, and design all light a spark of adventure and understanding in their eyes.
1.1 Project Topics

In brief, Constructivist learning principles focus on teaching through building and building through teaching. That is to say, students at many levels thrive in an environment where their own scholastic goals are aligned with the production of knowledge (or, in this case, digital products).\textsuperscript{1} As the current syllabus for Duke University’s Fundamentals of Web Based Multimedia Communication (ISS-240) has grown organically along with the substantive content of HTML and CSS, the logical conclusion has been to instruct students in the careful construction of a uniform midterm site hosting all of their gradable course content, and then to move on to a more freeform final website based around some personal interest, passion, or product. Such projects are limited only by the students' learned abilities, that is to say limited by what is covered in class.

Here is taken a learner-centered orientation, focusing on the student as learning agent within the paradigm of a Liberal Arts education—where coding plays either a mechanical or literary role. How the course adapts to and accommodates students at a variety of skill levels and interests are central to this approach.\textsuperscript{2}

Developing coding/markup skills is very much like learning to speak a new language; just as with languages, we must allow the individual student’s learning style to guide the instruction. By providing ample resources and patient guidance to students,


we enable them to develop independent approaches to coding the web, that do not rest on improvised workflows (see below) or other inefficiencies of ignorance.

### 1.1.1 The Pedagogy of Pedagogy

Few course structures offer a teaching assistant as much facetime with undergraduates. It is both challenging and fulfilling to see students grow as their skills progress, and it provides graduate students an excellent opportunity to explore the realities of working in educational environs. Here, instructional design has given thought to formatting/formulating syllabus/plans/projects/calendars to maximize students understanding, provide a variety of avenues for students to understand essential concepts, and myriad further possibilities.

Education on the importance of the digital and illustrating how connectivity plays a central role in their daily lives is an essential instrument for the success of students. At stake is the matriculation of well-educated but underprepared inexperts. Former students stepping into the professional or academic sphere without complete comprehensive education in the use and functionality of a machine that has become as essential in 2019 as the book had become in 1819.

Any university would be shocked to learn that it has been accepted as customary for students to have been told to make use of dictionaries, encyclopedias, and thesauri, but that still many graduates were unable or unaware of alphabetization, bookmarks, concordances, or indexes. This is much the situation, however, professors who do not have specialized training in the use of or education on nuanced software platforms and complicated computation do as best they can and demand a high standard from students with incomplete instruction.
1.1.2 Web Coding Education

Code fluency and computational/information literacy is generally lacking among the students who appear in this course. Perhaps chief among the multiple causes of this deficiency is the lack of computer education in primary schools around the world.

Not so long ago, the novelty of internet connected technologies (ICTs), or indeed of computers generally, prompted an educational explosion within the American public education system, students were lined up in front of any functioning machine and tasked with practicing keystroke and click, formatting, word processing, printing, and slide design. However, as computational materials and machines became ubiquitous this emphasis on how to compute got lost in a tidal wave of necessity. The completion of any classwork became screen time, and students' workflow and social life became bound up in computation. As a result, a knowledge gap between has developed between educators who are definitionally not digital natives and their students who definitionally are. Students now center their entire work mode around screen, keyboard, and trackpad uneducated about the machine running them.

1.2 Goals for Development

Here is taken the stance that this course is, at its root, about computational literacy and analytic thinking; it seems necessary that it serve equally as a gateway into higher levels of computational learning, digital media and design, et cetera. But also, as a primer for these directionless digital natives to develop essential skills, knowledge, and habits that lead to more effective and efficient workflows, optimal system use, and comfortable confidence in personal and professional digital settings. To approach this desired outcome, we must begin with the basics. Students may be familiar with the keyboard, but the sealed mac book is a perfect black box. So, students have learned their
particular computer environment through improvisation, feeling it out with neither instruction nor supervision. As a result, students have their own methods, processes, and workflows for using the machine, childhood trial, and error ossified into habitually daisy-chained frustrations.

By making certain insistences in the student structured files/assignments, we can effectively imprint more efficient processes on each class. This begins by providing students the comprehensive overview of their devices’ compulsions, limitations, and capabilities, and extends, onward through the necessary competencies of computer maintenance, upgrading, file system organization, logic, and storage, computer and internet safety, potential political/liability issues, a healthy dose of data terror, basic understanding of networking science, the spatial qualities of digital data, the resource demands of organizing, upgrading, and operating of complex networking systems.

The conceptual approach outlined here is the most important element of this paper. As an educator one must think critically about class time and student interactions in and out of the classroom. The approach taken in my lab sessions has been focused on the creation of a fun, collaborative, sharing environment where those who know or discover a solution are encouraged to distribute it, and those facing challenges are encouraged to look to resources first, peers second, and instructors last.

1.2.1 Conceptual Courses

Discussed extensively below, here are suggested several alternative forms this course material could take, allowing for more expansive educational content to be directed at a more extensive array of users, learners, and educators. These include the implementation of placement testing, the creation of modularized sections to focus on various elements of the design and development process intensively, and even to
recombine these modularities into dev teams tasked with the construction of a collective webpage. By creating and implementing a course based on web development, design, and understanding the internet as an element of basic worldly knowledge, students are far better equipped to tackle computational, digital, and design problems in other areas of their education. As educators there is an obligation to provide students with a robust, accurate, current education that does not rely on gap-filling by professionals who expertise lies elsewhere. Potentially, growing towards a system of courses, camps, educational events, and seminars that empowers students to live their daily networked lives and to find great success in future career aspirations.

1.3 Research Goals

W3Schools is useful and serving rather the function of an encyclopedia, something even more critical in a post paper age. Thanks to the immense depth and detail of the W3Schools indexing and exampling of HTML/CSS and JavaScript we can refer students to a public reference for nearly any web design problem they wish to tackle.

Indeed, we can send students on a journey of self-directed learning by asking them to read and explore the documents provided for education by the W3C itself. So, as this bountiful resource exists, why do we even need teachers? The first answer may seem obvious, but students, again just as in a foreign language classroom, develop at radically different rates, learning through varying methods of experience, trial and error, and self/instructor correction. A video and an encyclopedia cannot provide this carefully coordinated guidance for the student. Beyond this, and less obviously perhaps, is the power of synthesis. Again, referring to the logic of language learning, our hypothetical student might have great success in studying and building vocabulary, and our student
might even, through grappling with the available literature, grasp some rules of grammar in the target language. However, without some more directed guidance, the mesh of particular vocabulary and variants with grammar and personal style will never achieve a level approaching native fluency. While myriad resources do exist for guiding the novice web coder towards markup fluency, it is crucial to retain a focus on personalized education and the patient guidance of a familiar educator. So, rather like our students’ projects themselves, the course becomes responsive to the needs, wants and interests of the individual student and can respond and react in real-time to student input and developments in the field.

Currently the scholarship on web coding and coding pedagogy are of two primary prongs: the scientific and the MOOC. These approaches have achieved pedagogical dominance in last two decades for obvious reasons (access, funding, computational development, and ICT primacy). Similarly, open web courses, boot camps, and hackathons have sought to drag coding into the public sphere, though these events have attracted younger learners, the computer sciences still suffer from a startling lack of diversity. This paper seeks to apply pedagogical thinking form computer education to the diverse student body as it exists, to take the next logical step of implementing computer education within multiple courses of study, and then to extend it beyond the traditional university setting.
2. Pedagogy

A personal favorite is "painting with grammar," which is close to "just like learning a second language," or "drawing with my eyes closed."

2.1 Ethical Use

Students must understand how to copy and cite code responsibly, to account for the safety and security of their users and themselves. Given the recent explosion in the creation and dispersion of malware, ransomware, rat-ware, and other viral software, it is more important than ever for students and professionals to understand that logging on is a calculated risk, one that can be approached in a cautious and educated way. Failing to provide students with at least rudimentary understandings of computer and internet safety and security places them, their future clients, their employers, and their university at risk for victimization through identity theft, loss of intellectual property rights, theft, debt accrual, blackmail, medical fraud, and even danger of great bodily harm.

Further, these are only the wholly illegal entries on the list of ways the unwitting user can be misused by those only slightly savvier on the wild web. Computational algorithms used by market researchers, medical aggregators (for both health and profit), insurance agencies, and now, the federal government.

To this end, students need a grasp of the necessary skills involved in password selection, platform evaluation, resource evaluation, authority valuation, VPNs and DNS settings, protection of personal information, identity monitoring, computer maintenance, and security. These elements must be implemented out of our scholastic duty to students.

However, given the current restraints placed upon this course (maintaining the coveted QS credit), the syllabus leaves little room for substantive interrogation of the
aforementioned topics. To that end, the conceptual syllabi experiment with the original structure of the course to provide opportunities for the implementation of best security practices from day one, and to train the students into adopting these safety protocols as habitual parts of their digital routine.

2.2 Teaching problem solving

The purpose of the coveted course marker (QS-Quantitative Studies) for which this class remains in such demand is a focus on the development of quantitative analytic skills. In order to protect this valuable marker, past iterations of this course have opted to focus increasingly on mark-up in HTML/CSS. So much has this focus persisted that certain vestigial artifacts from the internet salad days have remained, while most of the media has fallen away from the multimedia aspects of Web-Based Communication in favor of a nearly exclusive focus on markup, position, style, and embed. While these are undeniably valuable skills for anyone in the 21st century, this approach asks only for students to replicate or repeat content from other sources—with the possible exception of writing. Of course, there are many ways to create content, to express artistic and academic pursuits, and to make statements on policy and environment that do not rest wholly or even primarily on the generation of original text.

Here the focus ought to shift to one that educates the student in coding logic, web markup and style, as well as digital design and creative problem solving. The teaching of web markup, associated styles, coding of interactivity, and even basic server-side functionalities provides an excellent substrate for the intellectual production of advanced and adaptive problem-solving skills. Indeed, all design projects are about problem solving at their root, just as the display or communication of any information is also about solving the problem of uninformative information. We should not mistake the
challenges and analytical rigors of learning a coding/markup language as being the sole opportunities for quantitative learning in such a course.

While significant portions of the developed syllabus are devoted to the introduction, explanation, and implementation of basic computer/digital maintenance and safety, these themes are threaded throughout the directed instruction on relevant coding practices and languages. In the same way best practices of coding and design are then threaded through the teaching of HTML/CSS. In this way we can implicitly draw parallels between these varieties of problem-solving endeavors and produce a more holistic education. An education which focuses on analytical thinking as a multipronged approach to the assessment, breakdown, and solution of any given design or technical obstacle.

By focusing so exclusively on markup language as the sole analytical element within this course we have produced a class with teaches students to ignore many (more potent) methods of analysis and solution. Here the argument is to be made that the value of the present course material is the opportunity to pursue both creative and logical problem-solving abilities, to approach obstacles and errors from many angles, rather than to suggest that only one problem with one solution exists. From a 2018 article in the journal *Sustainability*,

In the Application theme, a big majority of the students stated that in class activities, they did coding, wrote projects, did technical drawing, animations, produced three-dimensional images, and gave concrete examples of their imaginations and explanations related to their projects etc....participated in doing and learning and this raised their self-efficacy perceptions.  

---

Far too frequently coding culture is dominated by an efficiency standard propagated by a few inexpert economists examining their own work and reducing success to a simple equation of keystrokes and exit codes. While certainly there is something to be said for the application of such metrics as regards the efficiency of any piece of code or complied application as it relates to physical consumption of space, energy, and resources, or indeed the strain which is places upon the user or the machine on which it runs, these concerns seem rather frivolous in the context of introductory web markup.

Instead, let us give students the skills needed to explore and pursue analytic education on their terms; by developing their own solutions to myriad challenges posed by assignments, real world examples, the challenges of their own mistakes and misunderstandings, and those of their peers as well.

Again, to draw the comparison with education in foreign languages let us think that a student who evaluates and selects (after some trial and error) her own methods of retention and organization, develops her own affinities for mnemonics and rhythms, her own metrics for evaluating the performance and artistry of her grammar, is a student who is empowered to experiment and explore with that language in a social setting, rather than one who is prepared only to grapple with introductory texts. So, it is the instructor who should provide the student with knowledge and tools to develop individual solutions, rather than providing rote solutions ad nauseum.

### 2.2.1 How to Build for Students

It is essential that we begin by thinking about how to best construct our course for the students who will present in it. It is also imperative that as the strictures of a course become less intensive that students are guided by other relational strategies.
Faculty members—the research sample—emphasize the importance of providing a new proposal of design and developing E-learning constant based on digital concepts maps. Because it enables learner and teacher to build their own digital content, organize content and establishing a structure involving all content concepts.4

So, we must also think about the resources being presented to students, with careful attention those elements designed by teaching team.

2.2.2 Coding Demonstration

As an example, outlined below is a classroom collective approach to addressing consistent and widespread student issues with a lab assignment from a previous iteration of Duke University ISS-240, illustrated with screenshots and commentary. The instructor treads a fine when demonstrating solutions to course material generally, particularly so when the course structure centers on production and problem-solving in class. However, it is simply unimaginable to tackle the content presented here without demonstrating one’s own proficiency as a web coder through live coding. This is especially central to the approach used by the author in class when gaps in readings, tutorials, or lectures become apparent through widespread student struggles.

The particular assignment in question challenged student to create a responsive layout for a website subpage indexing previous (and future) lab assignments using CSS flexbox and grid. While students were provided with ample code to get started, several changes and application of sizing were required to create a functionally fluid webpage.

Generally, students, though provided with numerous examples of grid and flexbox both through assigned readings/videos and in lecture, struggled to achieve

responsive design without breaking the page. The primary cause here was not a lack of effort on the part of the students, nor of omission by the lecturer, but rather one of detail being subsumed by the necessary generalities of transmission. Though told that CSS grid will function only when all numbers of rows and columns are copacetic, many students failed when presented with grid-based articles within a larger gridded scheme. At issue here was the seemingly innocuous inclusion of a title grid-cell spanning the width of the top-level grid element. In simpler terms, the students were provided with most of a webpage sized appropriately for desktop browsers and arranged using CSS grid. Students were then tasked with using CSS media queries to adjust the number of columns in their top-level grid to produce a responsive design for smaller screens. Students stumbled owing to a gap in knowledge, while able to easily assess and implement view-port breakpoints within the style of the given page, the earlier inclusion of a single celled row at the top of grid caused page layouts to break at any additional screen size. In order the address these issues, common among all students in the lab at the time, it was decided to present the instructor’s own code and to demonstrate live the slight nuance necessary to produce the desired effect.
The method used is outlined below.

First, it is necessary to recreate the issue:

Figure 1: ISS-240 Lab #4 render at 400px screen-width (broken)
Having recreated the issue, and projecting the both code and render, students were asked to examine their own code and to identify potential causes, with specific direction to #page_title selector within their default CSS styles (see below).

```
 fig 2: iss-240 lab #4 code at 400px screen-width

    @media only screen and (max-width: 400px) {
        #labs-grid {
            grid-template-columns: 1fr;
            margin: 0 10% 0 10%;
        }
    }

 fig 3: iss-240 lab #4 code at default screen-width

    .lab_item {
        grid-template-columns: 1fr 2fr;
        grid-template-rows: 1fr 1fr 4fr;
        grid-column-gap: 5px;
        grid-row-gap: 5px;
        grid-template-areas:
            "image number"
            "image title"
            "image description";
        margin: 0;
        background-color: gray;
    }

    #page_title {
        grid-column: 1/4;
    }
```
Figure 4: ISS-240 Lab #4 render at default screen-width

This particular element was produced as part of an initial instruction for students to create a heading for the top-level grid element, by designating the top row as a single grid-cell containing the string “Labs” through the use of the #page_title selector. By presenting the students with a recreation of the issue being faced in a common setting brain-storming a solution to the problem became a class activity, rather than an independent one. And, in so doing, leveraging the class’s attention and retention as a
whole to identify why the problem occurred, the reason(s) certain solutions did not work, and, finally, the correct answer:

```css
122 @media only screen and (max-width:400px) {
123   labs-grid {
124     grid-template-columns: 1fr;
125     margin: 0 10% 0 10%;
126   }
127
128   nav-links {
129     font-size: .51em;
130   }
131
132   page_title {
133     grid-column: 1/1;
134   }
135 }
```

**Figure 5: ISS-240 Lab #4 code at 400px screen-width**

![Figure 5](image)

**Figure 6: ISS-240 Lab #4 render at 400px screen-width (fixed)**

18
2.3 Scalability

By introducing concepts of language classroom pedagogy into the coding classroom we open up many new avenues for student exploration and development, as well as creating new structure for learning that may be perceived as less intimidating than the traditional approach to markup. Such strategies include gamification, a focus on vocabulary, Socratic Method, and communal practice. Research has shown that students in flipped classrooms perform better when challenged with digital content. Further, students’ self-perceptions of efficacy increase when coding is taught through constructivist means. While this makes a strong case for the importance of human interaction to guide students’ classroom experience, this also indicates that coding and computational education benefit students across fields and demographics. The impact of this course material is not solely contained within webpages.

2.3.1 Where is the classroom?

The goal in the development of this project has been a course that is suited a wide variety of scholastic settings. While Duke University has been used as the framing device for the syllabus itself, the course notes and content are easily adaptable to a wide range of settings and skill levels. There has been, for some decades now, an impetus towards computational literacy gap filling in adult education. An area for which this course of study is totally suited; however, as discussed elsewhere in this paper, the more prescient

---

concern is how to address these gaps in younger generations as curricula fail to adequately address the machines so essential to their construction and completion.

The extremes of the scalable course extend from, as presented here, a university classroom as part of a larger body of study all the way to the purely digital distance learning self-instructed student—with all the stops in-between: primary and secondary schools, continuing/adult education, and online courses, training for new users and for the experienced. The course notes (seen in Appendix B Annotated Syllabus) provide for a wide array of applications, tailored to the students’ ability and desires.

2.3.2 Who are the Students?

The notion of a student persona is actually quite a modern one, now often leveraged by marketing types, its origin is in usability studies. From Usability.gov, “…representations should be based on qualitative and some quantitative user research and web analytics. Remember, your personas are only as good as the research behind them.”

Instead, the focus here is to approach students as individual persons, and to broadly classify the complex motivations that prompt them to arrive in the classroom, and to think about the place of this course within students’ hierarchy of priorities.

We must acknowledge and account for student motivations in enrolling in the class in the first place. Our goals in assigning constructive semester projects must align with students’ priorities for engaging with course content.

Students’ priorities that can be broadly construed as:

- Obtaining credit for course completion
- Developing marketable employment skills
- Developing computer skills for employment or personal enrichment
- In pursuit of other artistical-economic incentives

### 2.3.3 Inclusive classrooms

If code is culture, then the present lack of diversity in STEM represents a social catastrophe. Generally, the overwhelming presence of young women in web coding classrooms at Duke University is seen as a boon. However, many questions remain about supporting and encouraging these students. Further, it is essential to the success of this class both as educational process and as a system for identifying and encouraging talent within diverse student populations to counter the all too similar faces of STEM that the administration think about how the diversity of teaching team reflects and encourages diverse students.

However, the most inclusive classroom is the one that hosts a caring teacher. It is now common in customer service settings for employees to be encouraged to seek out genuine, spontaneous interactions with customers. The reasoning is two-fold, on one hand this personal touch appeals to consumers, giving the commercial setting a more intimate (and exploitable) sense, on the other, this mode of interaction is actually more pleasant for the labourer as it reduces the strain of constant emotional labour.

Relevantly, a similar approach can be instituted in the classroom, too often eager students lack the confidence to initiate friendly conversation with an instructor. This is the ideal product of this course design as implemented at Duke, no matter how large the class grows, sections ought to remain relatively small, enabling the instructor to bond
with students individually, creating a more social, comfortable atmosphere. The hope is to establish new modes of interaction emphasizing interpersonal relationships between instructor/student and between students as peers.

Further, the graphic below, a compilation of responses across categories focused on face to face contact, illustrates the importance of the immediate instructor.

![Pie chart showing the responses to what resources helped the most in the course]

**Figure 7: Compilation of student free responses** to "What resources helped you the most in this course?"

### 2.3.4 Other Contexts

Thought has been given to how this course can adapt to other scenes and settings than that of a university. While the last decades have seen a great deal of attention devoted to closing the technology gap between modern developments and older generations, this project is primarily concerned with addressing digital and internet
literacy within the so-called digital natives. These are students who have lived their entire post digital revolution, screens and files have shaped the context of their existence from earliest memory. However, as mentioned above, for many reasons these students who live a life of screen time are not receiving any formalized education on the subject of computers.

One important alternate context for the implementation of the course material is in elementary schools, important both for students to understand the contexts in which they are growing up, but also to understand digital security which will be paramount to their safety in coming years. Here is an excerpt of a helpful list from Tech & Teaching, “...6 Create hallway displays and bulletin boards with completed coding projects...12. Invite students to teach staff-members how to code and integrate projects with classroom lessons.” These two are highlighted for introducing two seemingly radical ideas for reaching out to younger students. The act of bringing coding explicitly into a space of play and physical decoration represents a more concrete notion of the subject, more easily grasped by young minds. The second point serves not only to reinforce students’ lessons by repeating them to others, but also it places individuals on either side of the digital gap in proximity to each other, focusing on building shared, compatible skills.

2.4 Course revaluation

In the appendices (A and B) at the end of this document, you will find the syllabus that has been prepared printed in full along with an annotated version outlining

---

the logic of various choices made as well as references to syllabi from previous iterations of this course and similar courses at other universities.

2.4.1 Student input/data

Included below are some data visualizations produced from a feedback survey given during the 2018-19 Fall semester, the survey, conducted at the conclusion of the semester had a fairly high response rate (~40%) and was administered via Google Forms.

![Word cloud generated from student free responses](image)

**Figure 8: Word cloud generated from student free responses**

The missing component is clear in the students’ minds. In practice, this course has made use of prefabricated JavaScript to dip students in, by testing their HTML and data management skills. However, students come into the class expecting or fearing the subject of Coding Logic and are uniformly disappointed when it does not arrive.

Smaller trends, less obvious, in visualizations include: a lack of utility in Web VR/AR, confusion over element positioning, and a dearth of visual design. The syllabus reconstruction has tried to incorporate these opinions into the course’s formulation. Further, the graphic below illustrates students’ preferred resources when coding, again the preference is clear. This result makes sense, of course, the W3C has provided ample and generous resources for web design students.
Figure 9: Student selected answers to "Where did you go for help?"

In rather the opposite spirit of the above images, the decision has been made here to remove portions of the course which focus on WordPress, despite some requests, as the use of proprietary platforms is not in the spirit of this course’s design.

2.4.2 Open Teaching

Like Open Scholarship, open teaching seeks to make this course widely available and adaptable not just to instructors and settings, but to students. It is a matter of respect that educators be upfront with students about how the backend parts of the class work, how they are being assessed, what grades mean, and what kind of feedback to expect based on the evaluation process. This strategy also draws on the notion of an open classroom, as space where co-learning and co-teaching can operate together to produce well rounded students.

Today, learners are coming together in different social settings and communities such as in groups in schools, teams in a company, and in informal
communities of learners. They share and co-create educational resources and interact with an ever-increasing number of open educational resources.⁹

This increases student sense of involvement in the course, builds trust between student and teacher, and increases the sense that one has been treated equitably. This also means being honest about the limitations of the course and the real-world applications of these skills, rather than suggesting outdated and uninteresting pseudo career elements.

2.4.3 Learning Strategy

Under Prof. Mark Olson the instruction became more explicit, it is an objective of this course for students to learn how research, test, and implement code/markup from publicly available sources. This key inflection on the course objectives opens up infinite possibilities for student projects, and more acutely demands intellectual engagement with the associated subject matter. Here is proposed an approach to rebuilding this course that focuses less on particularities of project processes and more on a cycle of research, test, implement, and citation, as well as tools/contexts through which these various experimentations can be presented to highlight content relevant to the students personal passions and professional pursuits.

Bearing this in mind, various portions of the course syllabus and semester projects have been reorganized to prioritize personal and artistic expression as essential elements of code mastery, and to emphasize the importance of research and implementation in the building of a website. From *Electronic Journal of E-Learning*, “In a society where information grows at exponential rates, tools change constantly, new apps are created almost virally, and software is in an ever-improving mode, people need to update their skills and knowledge...characterized by life-long learning.”

Rather than administering students a test, set students a goal. Lab exercises are not step-by-step walkthroughs, but rather seek to provide students with attainable goals, some strategies for achieving them, and the necessary tools to learn how to solve the problem.

---

3. Course Logistics

It is important students to understand fundamentals about code, that it is a hierarchy, that a webpage is a series of layers, structured by HTML, styled by CSS, and animated through JavaScript.

3.1 Proposed Learning Objectives

In order to retain the (Duke specific) QS credit, i.e., "Quantitative Studies," we must preserve a strong focus on the development and retention coding languages and logic. Importantly, these specific learning outcomes are increasingly emphasized in universities around the world and at all levels of primary education. However, in the interest of the students and the future applicability of the skills developed in this course, here is included a significant focus on networking sciences and accessible design. Of course, as this is an introductory class only certain sacrifices have been made for the sake of time and clarity.

3.2 Syllabus constructions

A twice weekly lab section, with online lectures, allows macro management, alleviating stress, fatigue, and preventing spread by allowing natural development.

As will seem obvious to those reviewing the included syllabi, this particular course construction falls short on elements of design. It is a simple fact that relevant content to this course far exceeds the possibilities available for limited class time. Indeed, even with a dozen semesters it would be difficult to exhaust the available lessons and exercises. The decision to deemphasize design was made here not because the subject is unimportant, but because the elemental aspects of design are generally intuitive and other areas of the course (CSS generally, positioning, mobile development,
accessibility, et al.) require explanation of these essential aspects. The priority for this syllabus, and for this project, is the lack of technical acumen and the bad habits and coping mechanisms students have developed to mitigate this failing of their educators.

### 3.2.1 Structure

Where this course has, at Duke, historically, focused on the historical processes of development and release that are the hallmarks of web development's developmental trajectory, here it is chosen to streamline the course in favor of a more rapid on-ramp to modern coding platforms, practices, and pitfalls. While it is essential for students to develop an understand of the technical processes and historical circumstances that lead us to now, it is clear that dynamic modeling such outdated kinds of code and structure leads to an increase in final errors rather than increased understanding.

Further, as many of the early assignments in the original syllabus focus almost exclusively on programming structures which are now obsolete, many of which modern browsers will not display. Similar to assigning snippets of unedited Chaucer to the beginner English student. These sparse and incomplete historiographical details are not only of little use to the student, but also provide, as a part of the students first initiation and scholastic task within our subject, information that is incorrect to use outside of the very narrow application of a single introductory assignment.

An inversion of the original structure of the syllabus, excising the outdated and outmoded elements and, instead, providing an introductory boilerplate, that is, a fully formatted and functional, but empty, mobile-ready webpage. With this boilerplate in hand we begin by instructing the students to breakdown each line of the extant code, streamlining the explanatory process of `<head>` and `<body>` etc., as essential elements found in any web code and demonstrating that in a real-world, professional setting.
This approach then lends itself to a natural segue into discussions of boilerplate generators, layout generators, and similar generative tools which can expedite the development process for the minimally experienced user. This is the main thrust of the course redesign, where the patchwork nature of the old archaeological structure of the course required that we instruct students to use incorrect coding elements as a means of demonstrating history, this redesign seeks to create an environment where students actively pursue shortcuts, not only in work but in their thinking. The ability to utilize and leverage these tools into meaningful developmental processes is a far more practicable, practical, and employable set of skills, indeed learning such shortcut thinking is a significant theme of the importance of coding education and essential ability for living and thriving in the Digital Age.

To this end, in the syllabus are replaced many of the job applicant roleplay elements of the course with more freeform exercises which are intended to stretch students’ skills and minds in both coding/analytical ways and into more complex creative dimensions. It is hoped that such an approach will encourage students to see the course as holistic elements of a complete set. Just as code, servers, and layers of networks all mesh together to create a browsable web page, so too do the complementary skills of analytical deduction, problem-solving, shortcutting, and artistic and academic investigation function together as sides of the same multidimensional coin, a wide and varied skillset standard to all people, but incredibly influential to the student, teacher, administrator, or anyone else who can learn to leverage these seemingly disparate elements of thought.
3.3 How it Works

It has been the primary goal in the design and construction of this syllabus and its related course materials to give students the most effective education on the capabilities of modern web design while retaining a strong focus on coding language and logic. To this end, as can be seen in the syllabus, students have been provided ahead of time with templates and outlines to fast track their headfirst dive into web design and coding.

This leads us to a prominent note on the revised structure of the syllabus, by providing students with a basic mobile web-ready template and detailed instructions for engaging with the complicated initial setup of their webpages and network directories, we are able to front-load the course with similar backend functionalities, such as HTML validation, accessibility checking, search engine indexing, and traffic analytic functionalities. In so doing, students are equipped with a wide range of diverse skills to...
create, post, and monitor web-based content in a way which is readily applicable to the post-graduation job market.

### 3.3.1 Common Student Struggles

Students not much younger than myself have spent their entire lives with screens and the universes of digital media that pour from them. However, by and large, these young people are left without directed education in the efficient use and safety of such devices. As a result, almost all students enter the lab with one significant deficiency: improvised workflows. Without education on the use and function of their devices, students have learned use rather like a native language—by seeing, copying, experimenting, guessing, and improvising.

The result of this is that students develop individual, nuanced, and sometimes ritualistic or superstitious attitudes towards simple tasks like saving or organizing files. Coupled with ignorance of some essential features students daisy-chain commands into a wrote pattern performed repeatedly and extending into their work as developers. This is a severe issue when file structure and pathways are introduced as essential elements of a sites Information Architecture, anxieties amplified by the wasted energy of frustration with inefficient operations.

### 3.4 Practical Considerations

It is the harsh reality that universities must account for certain economies when allocating funding and other resources. Examined below are several such considerations, and their prospective benefits to the university.
3.4.1 Respecting Students

Just as the administration must account for the needs of the market when allocating resources to manage a corporate entity the size of any university of college, educators must also account for the priorities of their students and accept the reality of place their course holds within each learner’s schema of responsibilities and obligations. Here is the first benefit of such a regularly constructed course, students can be directed quickly and efficiently to course content when a class is missed, allowing students adequate time to recover from illness while also ensuring they remain up to speed with the rest of the class. Similarly, athletes demanding schedules can be accommodated with only slight variations in deadlines, by ensuring each student has continuous access to the relevant course resources.

3.4.2 Value to the University

Such a course is also of great value to the University. The approaches outlined here produce student success, save money on instructors, and provides excellent optics, as well as the practical benefits of tech savvy students, known to have excellent computer skills.

3.5 The Conceptual Syllabus

Outlined below are several conceptual forms this course content could take, along with explanations of the thinking behind these decisions, and their prospective benefits to students and the University.

3.5.1 The Class of the Future

Let’s imagine a required learning pathway (or optional certificate program) that sits alongside other majors and certificates. Grafted onto these associated programs as a
means of ensuring student competency without placing undue stress on the resources of the student’s home department or requiring inexpert users to teach well beyond the scope of their field.

Such a course could make extensive use of themes, to bring computational concepts and projects into a high energy atmosphere. While also allowing course runners to emphasize other software platforms (e.g. excel for budgeting a project, word for crafting a document, Creative Cloud, Slack, and more) that will be of personal and professional use to all students.

3.5.1.1 Grading by web scraper

Through the standardization of assignment collection and grading, for example, we remove significant human latency from the system, and can indeed produce reports engineered for human readability and efficiency. However, reinforcement of basic file structure and baking conventions will be essential for both bot and student success. This is another opportunity to introduce the notion of open teaching into our dialogue, by explaining the grading backend system (it’s potential and implications) we reinforce both competencies with these essential elements of web design and code legibility, but we also provide students with valuable information about how to best manage their grades, assignments, and files.

There are distinct trends within the course, and along with each student’s developmental trajectory, course runners can leverage these to improve teaching and grading workflow, as well as student success. This approach reduces emotional and psychological implications of grading. In a 0, check, check+ system there is strong impulse against zeroes for those who have genuinely attempted but failed. By automating
the process to a reasonable degree, we can enforce standards and deadlines, streamline workflow, and reduce stress.

Owing to the extreme demand for placement in this course at Duke University, with similar conditions at comparable universities, the success of such a program requires access to wide array of instructors, capable of managing a semester of independent teaching. Much like the tutoring programs at many universities use an orientation to help new tutors get started, it is here that the concept of a boot camp becomes most relevant. Through a short intensive, likely held during breaks, graduate students of many backgrounds can get web-ready themselves.

Further, there is ample space for the centralization and expansion of many currently disconnected technology seminars. Such potential events include:

- **LaTeX Party**
  - Fast paced night, dinner included, for anyone interested in learning how to use the LaTeX mark-up system.

- **Creative Cloud Workshop**
  - Allows students to bring in an in-progress Adobe project for assistance and critique

Finally, the course of the future looks beyond traditional standards of achievement to produce a more fulfilled and competent student. It is essential to begin exploring gradeless course structures that focus on developing the individual student and focus on growth. One potential area for this expansion is through the application of student experience points for completion of certain credits within the course pathway to unlock access to other seminars or areas of the course.
As the academic landscape changes so too must the educators, institutions, and policies who tend, cultivate, and organize it. It is no secret in the academe that so-called "grade inflation" has slowly but inexorably pushed the average student from an arbitrary "C" to an arbitrary "B" or even "A-". Over the perhaps twenty years, or more, there have been offered numerous attempts to explain this phenomenon, there are apologists for the upwards creep, those who blame debt, or blame faculty, or anxiety.

Though from all this it seems apparent that the extant system no longer functions effectively or efficiently. Both graders and gradees have been swept up in the transitioning system for all of the above reasons and more beyond them. To take the long view, it can be demonstrated that the foundational metrics of Western higher education are no longer effective strategies for evaluating student development and performance. Many departments and universities (even some community colleges) have abandoned the formal Final Exam Period, indeed many forward-thinking professors have dispensed with traditional testing altogether. Much as in the course outlined here, the focus in higher education has begun to shift towards constructive, analytic projects which require the student engage actively and critically with course content on their own terms rather than providing rote answers or factoids under artificial constraints of time and expression.

This question extends far beyond the limitations of this project, for the death of the gradebook is an issue which affects each and every discipline in every classroom. Rapidly the cost of class time inflates for both student and administration, the emphasis here has been on individual education, student progress and improvement, highly personalized content and feedback offered to students in face-to-face meetings, in order to ensure a congruency between cost and value for students and the university. The
implication is that educators must begin to align evaluation policies with minimum standards of achievement paired with individual development to ensure both basic competency and sufficient difficulty appear within a single syllabus.

3.5.2 Concept 1-Modules

Modular courses focus on variations of the course features mentioned above, students can be presented with a menu of mini-course options on a variety of subjects. Something larger than a seminar but smaller than a course, a place to focus intensively on certain elements of digital practice and computational literacy, building their own version of the course across a given semester.

Basic considerations are necessary to ensure such a course would flow smoothly. If each module met in separate locations at the same time, moving students between them would not be difficult. Also, some consideration must be given to placement of students based on ability/existing knowledge, and to how certain modules relate to others.

Essentially, this is a series of short courses on narrow, individual topics; strung together by relation of topic, a set of modules within a larger container, or a sequence within the more extensive course. Such modules could include HTML/CSS, Graphic Design, Computer Literacy, Data Visualization, Web Backend, and more.

The course is cemented by capping lectures at either end, high energy, high impact presentations, trending away from stilted orientations, to get students excited about the program and their own education. Designed to move freely between the web and real-time, students can be provided with the resources, and a human instructor for class, but instruction remains largely self-motivated. The course as a series moves from basic computational literacy, through intro to coding/systems/logic, then into
specialized units (backend/server, UI/UX design, graphic design/multimedia, content design/creation/publishing/layout); the natural extension of this modular structure is to simply remove this course from the overall curricular structure of a given student’s studies and to place it rather within a space between general education requirement and extracurricular activity. This is the course calendar imagined not as a single string of classes culminating in a final, but rather a more freeform elective credit which encompasses mass group lectures, digital coursework and meetings, seminars, social events, and even class meetings and lab training sessions. Giving students the possibility to develop, explore, and optimize specialties within the broader field of computational literacy.

### 3.5.3 Concept 2-Teams and Specialists

Trending in a direction of exploration in line with Bass Connections, this concept focuses on engaging students with diverse skill sets as a team, to cooperatively solve design and technical problems, while also working through and across the essential stages of a web development project. Individuals focused on discrete elements of development and design are brought together to coordinate the building of an extensive group project on a topic of their choice.

### 3.5.4 Concept 3- Placement Testing

By implementing some variation of placement testing within the course structure we account for differing initial levels of ability. The complication are the varieties of differential ability in class. Invariably most students are scared stiff by the prospect of coding, much less working in the computational dimension. However, more than a few students are competently proficient, through either training or self-motivation, and
handle the machine itself with ease. Indeed, often have at least some introductory knowledge of coding and markup.

These students arrive in lab looking for training to develop web coding capabilities in a more advanced academic sense than those who arrive woefully ignorant of the capabilities of their all-important machine. Yet there is a third subset of students, generally included in the more academically invested group mentioned above, who have ample training in the use and maintenance of their machine, education in the production of code and markup, and who come seeking professional development of web coding skills.

Problematically, this professional education subgroup also generally represents students who are disinterested in studying the art/design/social aspects of web design and markup (all the more important for them to learn) and show a preference towards more concrete backend technologies that support the web frameworks on which this introductory course focuses.

Just as graduate students are (generally) sorted into separate classes, and must be confined to unique section numbers, by implementing a brief (optional) placement testing system we can sort certain students into more advanced sections within the overall course. Essentially, we are implementing an application system for advanced sections, in which students who wish to learn at an increased pace (primarily by extreme acceleration of introductory materials) allowing for more in depth exploration of advanced topics. We also prevent the ballooning of overhead course management by not requiring each student to test.

Indeed, depending on university and accreditation restrictions we may be required to qualify these advanced sections as separate courses, but the digital scalability
of the course design allows for a relatively lean teaching team to manage these differing syllabi without undue stress or time consumption. This remains a central question in the application of the present syllabus. Given the radically disparate skill and knowledge levels of past (and prospective) students, it is likely in the best interest of all involved to begin grouping students based on comprehension of basic elements of the course content.

A simple, web-based, placement survey could be quickly implemented, and would easily distribute students into ability-based clines. Thus, allowing for greater tailoring of course content to suit student needs and desires. Such a survey could be easily limited to only a few questions. First, to check the student's ability with and understanding of basic computational methods, e.g. creating directories and running updates. Describing one's practice with HTML/CSS by describing and annotating code and by interpreting basic html objects into raw code. Finally, by examining some basic JavaScript in a similar fashion.

By conducting such introductory survey, offered optionally, students could then be sorted into sections according to ability and proficiency, thus allowing course coordinators to streamline the approach of individual sections, but also this generally aggregates students of similar ability into sections, producing a smoother grading curve at the end of the semester. Such additional sections might include: Athletics, Graduate Students, Computer Science Majors, as well as Basic and Advanced. In this way we can encourage the more advanced students to achieve more, without altering printed standards or expectations.
4. Tips for TA's

This course is of great benefit to graduate students with any interest in teaching, especially if they have a little technical skill. Few other programs offer such extensive time to lecture and work directly with students, much less to work consistently with a limited group of students. Teaching Assistants also build experience managing data, student roles, grading, working in a self-motivated way without direct supervision. What is more, just one semester of teaching this course offers an acute familiarity with HTML/CSS foundational concepts.

4.1 Good Ideas

Here are presented some thoughts from various Teaching Assistants from the past, offered in hopes of helping new instructors:

- Use form emails
  - Luke C. LeGrand

  Using form emails is an excellent was to offer students high availability, with little effort, often several students will contact you with the same technical question regarding some technical or setup issue. Spend some extra time writing the first support message, and then save it for other students.

- Work through labs ahead of time
  - Katie King

  Try to work through the lab a couple of days ahead of time, you will frequently hear from students before your lab section meets with technical questions about that week’s assignment. By and large Duke students are highly motivated and grade oriented, they like specific feedback. Remember that you are guiding the section, you are not
expected to have all the answers, rather just to know how to stop and think about a problem and researching an answer.

• There is a gap between reading and practicing, occupied by the lab instructor.
  
  o Sarah Riazati

It is good practice to take a few minutes to annotate each day’s readings at the start of class. In addition to refreshing diligent students and informing those disinclined towards reading, this also reminds students that the readings are important and encourages them to do it.
5. Conclusion

SO, we see that through constructivist approaches to pedagogy and by converting our coding classrooms into places of exploration we can produce empowered students web ready for the digital age. By embracing the proven tenets of gamification and personal expression from the language classroom and engaging students with goal and production-oriented assignments build more inclusive learning environs, transitioning diverse student populations across borders of discipline and experience.

5.1 Looking Forward

The courses described here represent a constellation of opportunities. One that fits nicely into the educational landscape at Duke. Building on this project will require further streamlining and a transition from the Duke-centric model to a capsule sized course, a sleek collection of documents for publication and distribution under open scholarship practices.

In concrete terms, this means moving towards a commercial site for distribution, as well as adding links to Wikipedia, and testing course in new settings (primary/secondary schools, community colleges, and adult education.)
Appendix A: Syllabus

The Living Web

ISS-2048

Meeting Times: TBD

Fall 2048

Luke C. LeGrand

Luke.LeGrand@Duke.edu

Office hours: TTH 2-4pm

Course Description

Learn coding and computational essentials in fun, collaborative and creative environment. Students will meet twice a week in small lab sections with guidance from Instructors to complete assignments and get assistance with milestone projects.

Course Goals

Successful students will be able to:

• Read and write intermediate level HTML/CSS

• Implement and modify basic JavaScript and web coding libraries

• Understand coding logic and the essentials of Information Science

• Construct and execute design and technical plans for a digital project

An Introductory note:
Hello and Welcome! Briefly, before we begin, I would like to introduce myself as your instructor and share some personal feelings about this course. We want this class to serve as a safe space of exploration and development for each and every one of our students. This is a class where you are strongly encouraged to ask any questions that are on your mind, to discuss assignments and solutions with friends and neighbors, and seek help from/offer help to your classmates. I believe that an energetic, social, collaborative workspace produces the best learning outcomes for all students. Whatever it is you may need from me; I'll do my best to help. Please reach out to me by email or speak to me privately so we can ensure your needs are being met. Whatever it is you’re bringing, there’s room for it here.

**Required Equipment and Software**

- It is highly recommended that all students have a personal laptop for use in and out of class, lab computers are available for use during class sessions.

  - **Software**
    
    - CyberDuck or FileZilla (sFTP)
    - Brackets (text editor)
    - Cisco AnyConnect (Duke VPN)
    - Google Chrome, Firefox (browsers)

- While it is not a formal text for this course you will frequently need to reference w3schools.com

**At Home**

Before each lab session students are required to view the short video lectures posted to the course website, and to review the relevant articles.

**In Class**
Each lab session will focus on an individual Lab Assignment as listed on the course website. It is essential that all students review the video lectures and assigned readings before lab in order to make the best of your time with the Instructor and contribute to the class discussion.

Your participation in class and engagement with your instructor and fellow students is essential not only to your personal success in this course, but also to the success of the course as a whole. It is the goal of this class to explore the concrete structures of web coding and the abstract process of design in a collaborative, social environment. Ask and answer questions, compare strategies and notes, review and critique each other's work.

All interactions in this class will be civil, respectful, and supportive of an inclusive learning environment for all students. If you have questions or concerns about classroom participation and classroom dynamics, please speak with your instructor, the department chair, or another advisor.

**Communication policy**

As your instructor I will do my best to respond to emails within 48 hours. Before you email with questions, I ask that you conduct thirty minutes of research on the issue, search/post to the course forum, and include all available context for the issue you are facing.

**Major Assignments:**

There are two (2) milestone assignments for the semester

**Midterm Site**
The **Midterm Site** will serve as your main portal for turning in assignments throughout the semester. Each lab assignment will contribute to the overall complexity of the Midterm Site in addition to requirements listed below.

1. A fully formed website including
   a. A Homepage
   b. A Labs index page
   c. An About Me page
   d. A Final Project page
   e. Two (2) Additional Pages of your choice
   f. A blank template page for you lab assignments

2. Observe basic principles of design and accessibility

3. All **Lab Assignments** must be correctly listed, linked and uploaded to class server space

4. A fully functioning menu system linking all pages to one another

5. Be free of **ALL** validator errors, except those specifically noted by your instructor, if any

**Final Project Site**

The Final Project site must be independent of your midterm site. The Final Project site can be focused on any subject which is of personal use or interest to you. Some examples include but are not limited to: Website for a club or student organization, an art portfolio, an ecommerce platform, a personal blog, a webpage detailing research or other academic pursuits you wish to share with the general public, a professional website detailing your experience and talents, a project of creative expression.
The Final Project site must include the following:

1. A fully formed website including
   a. A Homepage
   b. An About page
   c. AT LEAST 4 additional pages

2. Present a consistent, refined personal style in observation of design and accessibility principles covered in the course

3. A fully functioning menu system linking all pages together

4. At least one (1) imported font

5. At least 2 JavaScript or library implementations not covered in class

6. Be free of all Validator errors

Course Grading

The goal in providing grades and feedback is to help you develop as a coder and web designer. Please review feedback from your instructor carefully.

Explanation of Grading System

All lab assignments and quizzes will be graded on a zero, check minus, check, check plus basis. Rubrics will be provided for grading of the Midterm and Final projects at least 1 month before the listed due date.

Attendance and Participation: 25%
Lab Assignments: 25%
Midterm Site: 25%
Final Project Site: 25%

Course Policies and Information for Students
Duke Statement of Commitment to Diversity and Inclusion

Duke aspires to create a community built on collaboration, innovation, creativity, and belonging. Our collective success depends on the robust exchange of ideas—an exchange that is best when the rich diversity of our perspectives, backgrounds, and experiences flourishes. To achieve this exchange, it is essential that all members of the community feel secure and welcome, that the contributions of all individuals are respected, and that all voices are heard. All members of our community have a responsibility to uphold these values.

ATTENDANCE POLICY

Attendance will be taken at the start of every scheduled lab session, two excused absences are accounted for.

PENALTIES FOR LATE WORK

No Lab Assignments will be accepted late, failure to turn lab assignments on time will result in a 0.

ETHICS/VIOLATIONS OF ACADEMIC INTEGRITY: Ethical behavior is an essential component of learning and scholarship.

Duke Community Standard and Student Conduct

Last updated: August 27, 2018

Duke Community Standard

Duke University is a community dedicated to scholarship, leadership, and service and to the principles of honesty, fairness, respect, and accountability. Citizens of this community commit to reflect upon and uphold these principles in all academic and nonacademic endeavors, and to protect and promote a culture of integrity.

To uphold the Duke Community Standard:
▪ I will not lie, cheat, or steal in my academic endeavors;
▪ I will conduct myself honorably in all my endeavors; and
▪ I will act if the Standard is compromised

If you have any doubts about what constitutes a violation of the Academic Integrity policy, or any other issue related to academic integrity, please ask your instructor. If a student comes to me to discuss or disclose an instance of sexual assault, sex discrimination, sexual harassment, dating violence, domestic violence or stalking, or if I otherwise observe or become aware of such an allegation, I will keep the information as private as I can, but as a staff member of Duke University, I am required to immediately report it to my Department Chair or Dean or directly the University’s Title IX Coordinator. Additionally, you can report incidents or complaints to Student Affairs.

MENTAL HEALTH: Duke University Student Health provides a variety of mental health and emotional support services through the CAPS program, you can contact them here.

**Disclaimer**

The instructor reserves the right to make modifications to this information throughout the semester.
<table>
<thead>
<tr>
<th>DATE</th>
<th>SESSION 1</th>
<th>SESSION 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>LAB 1: Updates, Installations, and Explorations</td>
<td>LAB 2: Breaking down Web Design</td>
</tr>
<tr>
<td></td>
<td>Difficulty: 3/5</td>
<td>Difficulty: 1/5</td>
</tr>
<tr>
<td></td>
<td>Notes: Introductions</td>
<td>Notes:</td>
</tr>
<tr>
<td></td>
<td>Readings:</td>
<td>Everything You Ought to Know About the Internet</td>
</tr>
<tr>
<td></td>
<td>What is the Internet?</td>
<td>Readings:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W3c on Mobile First Design</td>
</tr>
<tr>
<td>Week 2</td>
<td>LAB 3: Hello Worldwide Web!</td>
<td>LAB 4: Oh, What a Tangled Web(site)!</td>
</tr>
<tr>
<td></td>
<td>Difficulty: 1/5</td>
<td>Difficulty: 2/5</td>
</tr>
<tr>
<td></td>
<td>Notes: Responsive Design</td>
<td>Notes: Building a Website Infrastructure</td>
</tr>
<tr>
<td></td>
<td>Readings:</td>
<td>Readings:</td>
</tr>
<tr>
<td></td>
<td>w3schools Responsive Design</td>
<td>CSS</td>
</tr>
<tr>
<td></td>
<td>Responsive Design</td>
<td>W3c CSS</td>
</tr>
<tr>
<td>Week 3</td>
<td>LAB 5: The Elements of Style</td>
<td>LAB 6: Dressing for the Weather</td>
</tr>
<tr>
<td>Difficulty: 1/5</td>
<td>Difficulty: 3/5</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Notes: Coding with Style</td>
<td>Notes: Informed Design: Usability, Accessibility</td>
<td></td>
</tr>
<tr>
<td>Readings:</td>
<td>Readings:</td>
<td></td>
</tr>
<tr>
<td><a href="#">CSS Pocket Reference</a></td>
<td><a href="#">Penn State Accessibility Checklist</a></td>
<td></td>
</tr>
<tr>
<td><a href="#">W3Schools Tutorial on CSS</a></td>
<td><a href="#">Duke Learn</a></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

- CSS Pocket Reference
- W3Schools Tutorial on CSS

**Week 4**

**LAB 7:** A Class-y Webpage

**Difficulty: 2/5**

**Notes: Div, class and ID**

**Readings:**

- [Difference Between ID and Class](#)
- [Relevant and Absolute Types](#)
- [CSS Zen Garden](#)
- [W3Schools sections on the box model](#)

**LAB 8:** Boxes and borders and padding OH MY!

**Difficulty: 3/5**

**Notes: Display and Position**

**Readings:**

- [Flex Box](#)
- [Flexbox Froggy](#)
- [Grid Garden](#)
<table>
<thead>
<tr>
<th>Week</th>
<th>LAB 9: Embedding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 1/5</td>
</tr>
<tr>
<td></td>
<td>Notes: Embedding media and @import</td>
</tr>
<tr>
<td></td>
<td>Readings:</td>
</tr>
<tr>
<td></td>
<td>W3Schools on iframes</td>
</tr>
<tr>
<td></td>
<td>W3Schools on @import and fonts</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Week</th>
<th>LAB 10: De Colores!</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 4/5</td>
</tr>
<tr>
<td></td>
<td>Notes: Decoration vs. Design</td>
</tr>
<tr>
<td></td>
<td>Readings: palleton.com</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Week 6</th>
<th>FALL BREAK</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Week 7</th>
<th>MIDTERM OPEN LAB HOURS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Week 8</th>
<th>LAB 11: It's Alive!</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 4/5</td>
</tr>
<tr>
<td></td>
<td>Notes: JavaScript and Coding Logic</td>
</tr>
<tr>
<td></td>
<td>Readings: JavaScript Tutorial</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Week 9</th>
<th>LAB 12: Motion Potion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 3/5</td>
</tr>
</tbody>
</table>

<p>|        | LAB 13: A hot cup of Java(script)! |
|        | Difficulty: 3/5         |</p>
<table>
<thead>
<tr>
<th>Week 10</th>
<th>LAB 14: Destination: Final</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 2/5</td>
</tr>
<tr>
<td>Notes:</td>
<td>A tour of final projects past</td>
</tr>
<tr>
<td>Readings:</td>
<td>Mozilla CSS Animations</td>
</tr>
<tr>
<td></td>
<td>Keyframe Animations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Week 11</th>
<th>LAB 15: The .gif that keeps on giving</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 1/5</td>
</tr>
<tr>
<td>Notes:</td>
<td>Photoshop Walkthrough</td>
</tr>
<tr>
<td>Readings:</td>
<td>Chris Jackson</td>
</tr>
<tr>
<td></td>
<td>Angelina Liu</td>
</tr>
<tr>
<td></td>
<td>Soren Chargois</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Week 12</th>
<th>LAB 17: Searchin' for Somethin'</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 2/5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Week 11</th>
<th>LAB 16: You are being watched</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 1/5</td>
</tr>
<tr>
<td>Notes:</td>
<td>What are analytics? Your Data Matters</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Week 12</th>
<th>LAB 18: The Great Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 1/5</td>
</tr>
<tr>
<td>Week 13</td>
<td>Notes: How do Search Engines Search?</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td></td>
<td>FINAL OPEN LAB HOURS</td>
</tr>
<tr>
<td>Week 14</td>
<td>BREAK</td>
</tr>
<tr>
<td>Week 15</td>
<td>FINAL</td>
</tr>
</tbody>
</table>
Appendix B: Annotated Syllabus

The Living Web

ISS-2048

Meeting Times: TBD

Fall 2048

Luke C. LeGrand

Luke.LeGrand@Duke.edu

Office hours: TTH 2-4pm

Course Description

Learn coding and computational essentials in fun, collaborative and creative environment. Students will meet twice a week in small lab sections with guidance from Instructors to complete assignments and get assistance with milestone projects.

Course Goals

Successful students will be able to:

- Read and write intermediate level HTML/CSS
- Implement and modify basic JavaScript and web coding libraries
- Understand coding logic and the essentials of Information Science
- Construct and execute design and technical plans for a digital project

An Introductory note:
Hello and Welcome! Briefly, before we begin, I would like to introduce myself as your instructor and share some personal feelings about this course. We want this class to serve as a safe space of exploration and development for each and every one of our students. This is a class where you are strongly encouraged to ask any questions that are on your mind, to discuss assignments and solutions with friends and neighbors, and seek help from/offers help to your classmates. I believe that an energetic, social, collaborative workspace produces the best learning outcomes for all students. Whatever it is you may need from me; I'll do my best to help. Please reach out to me by email or speak to me privately so we can ensure your needs are being met. Whatever it is you're bringing, there's room for it here.

**Required Equipment and Software**

- It is highly recommended that all students have a personal laptop for use in and out of class, lab computers are available for use during class sessions.

- **Software**
  - CyberDuck or FileZilla (sFTP)
  - Brackets (text editor)
  - Cisco AnyConnect (Duke VPN)
  - Google Chrome, Firefox (browsers)

- While is not a formal text for this course you will frequently need to reference w3schools.com

- This syllabus adapted from previous versions prepared by Prof. Richard Lucic, Prof. Raquel Salvatella dePrada, Prof. Mark Olson, and Prof. Victoria Szabo, along with Augustus Wendel and Max Symulueski.
• An easy title that suggests both the importance of the subject of the class, as well as rooting the concept of the course within a framework more accessible to students of the natural sciences and humanities

• Place holder course information for the purpose of display and formatting

• The given course description is meant to offer a brief introductory snippet to inform potential enrollees of the concepts and content explored in the course, as well as important course objectives. This is, in effect, our pitch to prospective students seeking to enroll in the course

• As a general rule, accreditation agencies require the creation and maintenance of certain course objectives which are intended to organize the flow of coursework and motivations of both student and teacher, I have chosen to include here practical elements which are at the heart of this particular project, though the can be modified to the satisfaction of various entities, agencies, and administrators.

• A note: I chose to include this note as a means of, hopefully, encouraging students not only to seek me out first when they encounter obstacles with the course, but also to reinforce the sense that this course is meant to be a collaborative space

• Originally the plan was to produce and provide a collected and compressed assortment of necessary software along with teacher produced readme.txt, pdf textbooks, and even some pre-prepared html boilerplate to get students up and running as quickly as possible. However, several functional challenges persisted, the aggregated software zip files were still of an unreasonable size, further downloading and installation of appropriate versions is best left to the individual software licensors. Also, many students enter this course without the wherewithal to mediate issues on the fly, or indeed for the teaching team to assume current operating systems and active updates. So the functional
operating choice is rather to begin each course offering by mandates that each student perform basic computer maintenance (restarting, installing software updates, cleaning out disused software and bloat files, and installing a reasonable form of proactive antivirus software) and then providing a curated list of required software, withholding certain alternatives for individual troubleshooting later on.
At Home
Before each lab session students are required to view the short video lectures posted to the course website, and to review the relevant articles.

In Class
Each lab session will focus on an individual Lab Assignment as listed on the course website. It is essential that all students review the video lectures and assigned readings before lab in order to make the best of your time with the Instructor and contribute to the class discussion.

Your participation in class and engagement with your instructor and fellow students is essential not only to your personal success in this course, but also to the success of the course as a whole. It is the goal of this class to explore the concrete structures of web coding and the abstract process of design in a collaborative, social environment. Ask and answer questions, compare strategies and notes, review and critique each other’s work.

All interactions in this class will be civil, respectful, and supportive of an inclusive learning environment for all students. If you have questions or concerns about classroom participation and classroom dynamics, please speak with your instructor, the department chair, or another advisor.
**Communication policy**

As your instructor I will do my best to respond to emails within 48 hours. Before you email with questions, I ask that you conduct thirty minutes of research on the issue, search/post to the course forum, and include all available context for the issue you are facing.

**Major Assignments:**

There are two (2) milestone assignments for the semester

**Midterm Site**

The **Midterm Site** will serve as your main portal for turning in assignments throughout the semester. Each lab assignment will contribute to the overall complexity of the Midterm Site in addition to requirements listed below.

6. A fully formed website including
   a. A Homepage
   b. A Labs index page
   c. An About Me page
   d. A Final Project page
   e. Two (2) Additional Pages of your choice
   f. A blank template page for you lab assignments

7. Observe basic principles of design and accessibility

8. All **Lab Assignments** must be correctly listed, linked and uploaded to class server space

9. A fully functioning menu system linking all pages to one another

10. Be free of **ALL** validator errors, except those specifically noted by your instructor, if any
• A brief overview of the kinds of work students can expect to be consistently responsible for outside of the classroom, and the importance of student preparedness not only for themselves but for their instructor and colleagues.

• A brief description of the kinds of work and problem-solving skills that will consistently appear as key points of the course

• In much the same way as the above comments, by including a statement on (ungraded) necessity of class participation and the standards of conduct required, we set students expectations for what "being in class" will require.

• The explicit establishment of a Communication Policy both encourages students to solve problems independently or along with their peers, and sets a standard for how much time solving any one problem should take

• Midterm: The **Midterm Site** will serve as your main portal for turning in assignments throughout the semester.
  
  o Your project must have:

  i. A fully formed **website** including
     1. A Homepage
     2. A Labs index page
     3. An About Me page
     4. A Final Project page
     5. Two (2) Additional Pages of your choice

  ii. Observe basic principles of design and accessibility

  iii. All **Lab Assignments** must be correctly listed, linked and uploaded to class server space

  iv. A fully functioning menu system linking all pages to one another
v. Be free of **ALL** validator errors, except those specifically noted by your instructor, if any.

- **Rubric**
  - **Validation and Functionality 50/100**
    - No validator errors, no extraneous code
    - All links/images/videos load as intended
  - **Meets Basic Requirements 30/100**
    - See assignment
  - **Design 20/100**
    - The styles and layout of the site must show:
      - **Consistency**
        - Each page is clearly a part of the whole
      - **Accessibility**
        - Text and images are of a reasonable size and contrast
        - Basic conventions of accessibility have been observed
      - **Intentionality**
        - Each element of the design contributes to the purpose of the page/site
      - **Thoughtfulness**
        - There is a logic behind particular design choices
Final Project Site

The Final Project site must be independent of your midterm site. The Final Project site can be focused on any subject which is of personal use or interest to you. Some examples include but are not limited to: Website for a club or student organization, an art portfolio, an ecommerce platform, a personal blog, a webpage detailing research or other academic pursuits you wish to share with the general public, a professional website detailing your experience and talents, a project of creative expression.

The Final Project site must include the following:

7. A fully formed website including
   a. A Homepage
   b. An About page
   c. AT LEAST 4 additional pages

8. Present a consistent, refined personal style in observation of design and accessibility principles covered in the course

9. A fully functioning menu system linking all pages together

10. At least one (1) imported font

11. At least 2 JavaScript or library implementations not covered in class

12. Be free of all Validator errors
Course Grading

The goal in providing grades and feedback is to help you develop as a coder and web designer. Please review feedback from your instructor carefully.

Explanation of Grading System

All lab assignments and quizzes will be graded on a zero, check minus, check, check plus basis. Rubrics will be provided for grading of the Midterm and Final projects at least 1 month before the listed due date.

Attendance and Participation: 25%
Lab Assignments: 25%
Midterm Site: 25%
Final Project Site: 25%
• Final Project
  o The Final Project site must be independent of your midterm site and will be focused on a subject which is of personal use or interest to you. Some examples include but are not limited to: Website for a club or student organization, an art portfolio, an ecommerce platform, a personal blog, a webpage detailing research or other academic pursuits you wish to share with the general public, a professional website detailing your experience and talents, a project of creative expression.

• Assignment

  The Final Project must have:
  a. A fully formed website including
    i. A Homepage
    ii. An About page
    iii. At least 4 additional pages
  b. Present a consistent, refined personal style in observation of design and accessibility principles covered in the course
  c. A fully functioning menu system linking all pages together
  d. At least one (1) imported font
  e. At least 2 JavaScript or library implementations not specifically covered in class
  f. Be free of all Validator errors

• Rubric
o Validation and Functionality 30/100
  o No validator errors
  o No extraneous code
  o All links/images/videos load as intended

o Meets Basic Requirements 35/100
  o See assignment

o Design 35/100
  o The styles and layout of the site must show:
    o Consistency
      • Each page is clearly a part of the whole
    o Accessibility
      • Text and images are of a reasonable size and contrast
      • Basic conventions of accessibility have been observed
    o Intentionality
      • Each element of the design contributes to the purpose of the page/site
    o Thoughtfulness
      • There is a logic behind particular design choices
Course Policies and Information for Students

Duke Statement of Commitment to Diversity and Inclusion

Duke aspires to create a community built on collaboration, innovation, creativity, and belonging. Our collective success depends on the robust exchange of ideas—an exchange that is best when the rich diversity of our perspectives, backgrounds, and experiences flourishes. To achieve this exchange, it is essential that all members of the community feel secure and welcome, that the contributions of all individuals are respected, and that all voices are heard. All members of our community have a responsibility to uphold these values.

ATTENDANCE POLICY

Attendance will be taken at the start of every scheduled lab session, two excused absences are accounted for.

PENALTIES FOR LATE WORK

No Lab Assignments will be accepted late, failure to turn lab assignments on time will result in a 0.

ETHICS/VIOLATIONS OF ACADEMIC INTEGRITY: Ethical behavior is an essential component of learning and scholarship.

Duke Community Standard and Student Conduct

Last updated: August 27, 2018

Duke Community Standard

Duke University is a community dedicated to scholarship, leadership, and service and to the principles of honesty, fairness, respect, and accountability. Citizens of this community commit to reflect upon and uphold these principles in all academic and nonacademic endeavors, and to protect and promote a culture of integrity.
To uphold the Duke Community Standard:

▪ I will not lie, cheat, or steal in my academic endeavors;
▪ I will conduct myself honorably in all my endeavors; and
▪ I will act if the Standard is compromised

If you have any doubts about what constitutes a violation of the Academic Integrity policy, or any other issue related to academic integrity, please ask your instructor. If a student comes to me to discuss or disclose an instance of sexual assault, sex discrimination, sexual harassment, dating violence, domestic violence or stalking, or if I otherwise observe or become aware of such an allegation, I will keep the information as private as I can, but as a staff member of Duke University, I am required to immediately report it to my Department Chair or Dean or directly the University’s Title IX Coordinator. Additionally, you can report incidents or complaints to Student Affairs.

MENTAL HEALTH: Duke University Student Health provides a variety of mental health and emotional support services through the CAPS program, you can contact them here.

Disclaimer

The instructor reserves the right to make modifications to this information throughout the semester.
### Table 2: Annotated Syllabus

<table>
<thead>
<tr>
<th>DATE</th>
<th>SESSION 1</th>
<th>SESSION 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>LAB 1: Updates, Installations, and Explorations</td>
<td>LAB 2: Breaking down Web Design</td>
</tr>
<tr>
<td></td>
<td>Difficulty: 3/5</td>
<td>Difficulty: 1/5</td>
</tr>
<tr>
<td></td>
<td>Notes: Introductions</td>
<td>Notes:</td>
</tr>
<tr>
<td></td>
<td>Readings:</td>
<td>Readings:</td>
</tr>
<tr>
<td></td>
<td>What is the Internet?</td>
<td>W3c on Mobile First Design</td>
</tr>
</tbody>
</table>

*What is the Internet?*

*W3c on Mobile First Design*
• Lab1
  • Video 1: Introductions
    o about your instructor
    o Basic computer functions
    o Shortcuts
    o Updates
    o Antivirus
    o Search/Spotlight
    o Computer Terminology/hardware
    o What the machine can do
    o File structure + consequences
    o Intro to what html/CSS looks like
    o Installation and start up
    o Overview of the syllabus and major course assignments
    o Installs
• Before Class
  o Updates and Downloads:
    o Make sure updates have been installed before proceeding,
      either way give your laptop a restart before we get started.
    o Then, download and install: CyberDuck, Firefox, Chrome,
      Cisco AnyConnect, and Adobe Brackets
• In Class
  o SFTP and VPN Set up
o Website Setup (if necessary)
  o Review Website home directory setup for public_html folder
  o Duke VPN at http://portal.duke.edu if off-campus
  o Moving Files Between Your Computer and the Web Server
    o Mounting your home directory: OIT instructions for Windows | for Mac https://software.duke.edu/node/86
    o SFTP into your site using CyberDuck or another FTP program
  o Backup strategies for the semester (local copies, server)
    o Log into your personal Duke Box to back up your files (create a folder for your coursework and share with your section leader and lecturer ves4 if you want to be able to share code for troubleshooting
    o Also recommended: external hard drive or big USB key

▪ After Class
  ▪ Finish installs

▪ Lab2
  ▪ Video 2: Everything You Ought to Know About the Internet but were too Afraid to Ask
    o Shorts on computational history
    o Getting intimate with html, CSS, and JS (what these say and do)
    o Browser as context
      o Through the browser we view content that is:
        ▪ Structured by HTML
          o <div> the basic unit of our page
- Some specialized names like
  - <body>
  - <header></header>
  - <nav></nav>
  - <main></main>
  - <footer></footer>

- In fact, we can use any name we want:
  - <watermelon></watermelon>
  - <kitten></kitten>
  - <the_abstract_concept_of_hope>
  - </the_abstract_concept_of_hope>

  - Styled by CSS
  - Animated by JS

- In Class
  - Deconstructing a website (*group work; in class*)
  - In groups of three select and analyse the design of any website
    - Focus on use of colour and image
    - Effective display and communication of information
    - Intuitive controls/interface/navigation

- After Class
  - Lab 2a
    - See (*Appendix C*) and follow the instructions to annotate the given code.
<table>
<thead>
<tr>
<th>Week 2</th>
<th>LAB 3: Hello Worldwide Web!</th>
<th>LAB 4: Oh, What a Tangled Web(site)!</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 1/5</td>
<td>Difficulty: 2/5</td>
</tr>
<tr>
<td>Notes:</td>
<td>Responsive Design</td>
<td>Notes: Building a Website Infrastructure</td>
</tr>
<tr>
<td>Readings:</td>
<td><a href="#">w3schools Responsive Design</a></td>
<td>Readings: CSS</td>
</tr>
<tr>
<td></td>
<td><a href="#">Responsive Design</a></td>
<td>Readings: W3c CSS</td>
</tr>
</tbody>
</table>
Lab 3

- Video 3: Using search engines
  - File structure redux
  - How to use W3
  - How to google
  - Assistances!
    - Using the inspector
      - Making intuitive design choices to scale across devices
  - In Class:
    - In this lab assignment you will build your very first webpage
    - Using the principles of mobile first design and template file provided, complete the outfitting of this basic mobile website, use comments to explain each discrete section of code
      - Set up FileZilla using the Website FTP Instructions
      - Launch the Brackets text editor
    - Use the given boilerplate (See Appendix C) code to build a mobile site.
    - Your site must include
      - A title in the head tag
      - 3 headings of at least 2 sizes
      - 2 displayed paragraphs with some text that is: bold, italic, superscript
- 1 ordered list
- 1 unordered list
- Make a link to another webpage
- 2 Horizontal lines
- Centered footer
- End the page with a copyright symbol, the year, and your name.
- Finally, specify a background color for your webpage
- 3 headings of at least 2 sizes
- 3 images
  - At least one (1) linked relatively
  - At least two fonts
  - At least two font colors
- Check your finished site for accessibility and errors, Upload mypage.html to your public_html folder following the SFTP/File Upload instructions.

- Test out your page on the web by going to your personal web folder from a browser.

- After Class:
  - Validators
  - Run your website pages through the W3C Validator: [https://validator.w3.org/](https://validator.w3.org/). What do you discover?
  - Fix any problems you can on your pages.
See also: http://cs-people.bu.edu/sworst/CommonValidatorErrors.html

**GENERAL VALIDATION NOTES:**

- Use the alt= option in your <img> tags to avoid validation errors.
- Also be aware that technically you should not put block elements (like <ul> or <ol>) inside <p> tags - you will get an error. To avoid validation errors, you use <div> tags for such formatting instead.

**Lab 4**

- **Video 4: Building a Website Infrastructure**
  - File order to Site infrastructure, menus, pages, sites, and subpages
  - Best uploading practices
  - Using Brackets and the file tree

- **In Class:**
  - This lab assignment focuses on the construction of your website infrastructure, this is the frame on which all of your subsequent lab assignments (and your midterm!) will rest!
  - In your class web_folder create and save the following files (it is advised to start each file with the given template from Lab 3)
    - index.html – your home page
    - About_me.html
    - subfolder for labs and index.html for the labs subfolder
    - subfolder for final_project and index.html for final project
    - Subfolder for storing media
  - Next update these pages:
o On the home page place a header, an image, and a short introduction

o Link all the subfolder index pages back to your main home page using relative links. You should be able to get back from any page in your file structure without ever hitting the “back” button.

o Create headers on the labs home and final_project home pages in anticipation of future content. Be consistent in your sizing choices!

o Create links for the Labs and Assignments we have already completed on the labs home page. Use bullet lists to organize them.

o Update your "About me" page with details and images of personal/professional interest

o If you have time, continue practicing with HTML to edit your pages and customize their appearance. Review the W3Schools tutorials for additional options.

o Upload everything to the web. Be sure you have included any images in your upload.

o Check your work and make sure everything is functioning from the web browser (not just the desktop!) This means checking your URL: http://people.duke.edu/~netID

o Check your pages for errors (correct and reupload)

o Check pages for accessibility and make a note of any given warnings)

- After Class:

  o Finish Lab 4
<table>
<thead>
<tr>
<th>Week 3</th>
<th>LAB 5: The Elements of Style</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 1/5</td>
</tr>
<tr>
<td>Notes: Coding with Style</td>
<td></td>
</tr>
<tr>
<td>Readings:</td>
<td></td>
</tr>
<tr>
<td>CSS Pocket Reference</td>
<td></td>
</tr>
<tr>
<td>W3Schools Tutorial on CSS</td>
<td></td>
</tr>
</tbody>
</table>

|       | LAB 6: Dressing for the Weather |
|       | Difficulty: 3/5               |
| Notes: Informed Design: Usability, Accessibility |
| Readings: |
| Penn State Accessibility Checklist |
| Duke Learn |
| WAV accessibility checker |
| Color Contrast Checker |
| Accessible Color Combinations |
Lab5

Video 5: Coding with Style
  - Fonts, colors, sizes, and centring
  - Intro to layout and positioning
  - Thinking about data transfer
    - *It is an essential element of accessibility that we remember not everyone has private school internet speeds. We must be economical about our data consumption.*

In Class:
  - Part 1
    - Now that you have spent some time with HTML, we will meet a new language CSS. Cascading Stylesheets (CSS) are external files linked to our html code which allow us to dictate every aspect of our webpages' appearance, from colour and font to size and position! In this lab we will create a single page-stylesheet combo so we can play with style!
    - Create a new sandbox.html file (using the template) and a sandbox.css file
    - Save both files to a new subfolder within the Labs folder called Sandbox
    - Update the `<link>` tag in sandbox.html to link it relatively to sandbox.css
    - Add a link from your Labs page to the sandbox.html page
Add a comment at the top of your CSS

- /* Lab 5 Sandbox style sheet - today’s date */
- /* Site wide styles */

Part 2

- Make an h1 heading on your webpage and color it with an h1 selector on your stylesheet.
- Similarly, make a bold paragraph using a “p” selector on the stylesheet.
- Increase the font size of another paragraph to 200%.
- Make an h2 heading with a monospace font.
- Make an h3 and right align it
- Now add another paragraph and transform a few words in the sentence to italics with an extra-small font size using the <span> selector.
- Change the page background color to something other than white.
- Upload the sandbox.html page and sandbox.css style sheet to your web folder, check for accessibility and errors

After Class:

- Finish Lab 5

Lab6

Video 6: Informed Design: Usability, Accessibility

- Intro to usability and UI, Accessibility
Accessibility checker

HTML and CSS validators

- In Class:
  - Goal: Create a simple, attractive, and responsive webpage for checking the weather.

- Guidance
  - Create a new html file from the template, and save it as Lab 6
  - Add a <style> tag inside the <head> of your new file
    - Go to: https://weatherwidget.io/ and create a new widget, with appropriate font and color choices, for the location of your choosing and paste the embed code into the body of your page
    - OR
      - Select another copy and paste html widget
  - Use CSS inside the <style> tag to center the widget
  - Add headings, body text, graphics, and additional widgets to create something meaningful, expressive, or artistic
  - At minimum
    - Add a pattern or gradient background
    - Change the background color of your widget
    - Add a gif or video
<table>
<thead>
<tr>
<th>Week 4</th>
<th>LAB 7: A Class-y Webpage</th>
<th>LAB 8: Boxes and borders and padding OH MY!</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 2/5</td>
<td>Difficulty: 3/5</td>
</tr>
<tr>
<td>Notes:</td>
<td>Div, class and ID</td>
<td>Notes: Display and Position</td>
</tr>
<tr>
<td>Readings:</td>
<td></td>
<td>Readings:</td>
</tr>
<tr>
<td>Difference Between ID and Class</td>
<td>Flex Box</td>
<td></td>
</tr>
<tr>
<td>Relevant and Absolute Types</td>
<td>Flexbox Froggy</td>
<td></td>
</tr>
<tr>
<td>CSS Zen Garden</td>
<td>Grid Garden</td>
<td></td>
</tr>
<tr>
<td>W3Schools sections on the box model</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lab 7

- Video 7: Class and id

- In Class:
  - Create a new html file from your template and save it as lab 7
  - Add an <h1> to identify the page, and use <h2> and <p> tags to write a few sentences on when would you use CLASS and ID and when would you simply redefine your basic HTML elements? Make reference to the idea of "cascading" styles in your answer. Write a paragraph - or more - explaining your reasoning on this topic and present it on this index_classy.html page using formatting specific and unique to that featured paragraph.
  - Add some images and other decoration to show off your skills with class
  - Then copy your lab_7.html file and rename in 7_classy.html. (Fix the page to address any errors that may have cropped up with the validator earlier if you haven't already.)
  - Restyle your 7_classy.html page using the CLASS and ID functions in CSS to define how content appears in your site. Keep it classy - and creative. Use multiple text, color, font effects as desired. The purpose here is to experiment, so change everything you can possibly think to change.

- After Class:
Finish Lab 7

- Lab8
- Video 8: Display and position
  - In Class:
    - Layouts and the Box Model
    - *Important: For this assignment, you will duplicate your entire website and put it inside a Labs #8 folder.*
  - Using CSS, lay out your homepage using `<header> <nav> <main> <footer>` and other nested `<div>` tags along with other box model elements to create a pleasing and effective interactive environment. Be sure to assign and specify attributes for borders, padding, spacing etc. to your preferred specifications, erring on the side of experimentation. Take full advantage of color and other stylistic options.
    - Use *grid areas* to arrange content on your page, *flexbox* to make a responsive menu, and incorporate at least fixed/sticky element
    - Flexbox should be used to create a horizontal layout for the navigation links. The "Home" link should be positioned on the left-hand side of the navigation element, with the remaining links on the right side.
  - Test out your page by resizing it with the browser to make sure it works properly.
  - Link your pages from your Labs page as Lab #8.
• After Class:
  • Finish lab 8
| Week 5 | LAB 9: Embedding  
|        | Difficulty: 1/5  
| Notes: | Embedding media and @import  
| Readings: |  
|         | [W3Schools on iframes](#)  
|         | [W3Schools on @import and fonts](#)  

**MIDTERM OPEN LAB HOURS**
• Lab 9
• Video 9: Imports and Embedding
  o Importing fonts
  o <link> vs @media
  o In Class:
    o In this lab we will experiment with adding content from other sources into our websites. In many cases this will be a desirable alternative to HTML embedding of content on your own server. You will also want to arrange this content effectively and attractively on your site.
    o Create a new web page for Lab #13.
    o Organize your page into four divs. In each div place one of the following:
      o Embed a YouTube video using an iframe. Select Share>Embed option and copy the appropriate embed code from the YouTube website:
        A Google or Bing map. To do this, select a location to embed first, then select Share>Embed Map. A Twitter stream. To do this, check out the Twitter Publication generator, enter a Twitter account You can also just go to an individual tweet and use the embed code you get when hovering over the 3 dots- as long as you update the URL for the platform link: https://platform.twitter.com/widgets.js”. More on Twitter embedding here.
o Now use an iframe to embed another website of your choice. Beware of embedding something big! Also note the limits on what will work in HTML5.

o Customize your web page CSS so that everything displays nicely. You may wish to use float and clear, for example, and/or some positioning. Try using some CSS3 features to customize the look of your site.

o Link from your Labs page as Lab #13. Upload and test
<table>
<thead>
<tr>
<th>Week 6</th>
<th>FALL BREAK</th>
<th>FALL BREAK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MIDTERM OPEN LAB HOURS</td>
<td>MIDTERM DUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In class critique!</td>
</tr>
<tr>
<td>Week 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAB 10: De Colores!</td>
<td>LAB 11: It's Alive!</td>
</tr>
<tr>
<td></td>
<td>Difficulty: 4/5</td>
<td>Difficulty: 4/5</td>
</tr>
<tr>
<td></td>
<td>Notes: Decoration vs. Design</td>
<td>Notes: JavaScript and Coding Logic</td>
</tr>
<tr>
<td></td>
<td>Readings: palleton.com</td>
<td>Readings: <a href="#">JavaScript Tutorial</a></td>
</tr>
</tbody>
</table>
Lab 10

Video 10: Decoration vs. Design

- Design furthers/enhances the purpose of the page
  - synthesis
- Color pallets and platforms

In Class:

- Goal: Create an aesthetically pleasing and well-designed single page website based on any noun of your choice.
- Use CSS and the elements of design discussed in the lecture video and readings to create a new, single page website that emphasizes your Noun in new and interesting ways.

After Class:

- Lab 10 part 2: Write a couple of paragraphs reflecting on the design, decoration, and color pallet of your midterm site as compared to lab 10, and add to your lab10.html, upload and validate.

Lab 11

Video 11: JavaScript and Coding Logic

- What JS looks like
- Basic coding logic
- JS syntax
- Explanation of Lab 11 (responsive menu) mechanics

In Class:
Goal: create a responsive drop-down menu

Go to:
https://www.w3schools.com/howto/tryit.asp?filename=tryhover_js_responsive_navbar_dropdown

Adapt this code to your Lab 10 page and

- Customize your menu to mesh with your site
- Add a breakpoint to separate the standard menu and dropdown
- Replace the hamburger menu icon □ with a different HTML symbol.

After Class:

- Finish Lab 11
<table>
<thead>
<tr>
<th>Week 9</th>
<th>LAB 12: Motion Potion</th>
<th>LAB 13: A hot cup of Java(script)!</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 3/5</td>
<td>Difficulty: 3/5</td>
</tr>
<tr>
<td></td>
<td>Notes: CSS animations and Motion Design</td>
<td>Notes: JS libraries</td>
</tr>
<tr>
<td></td>
<td>Readings:</td>
<td>Readings:</td>
</tr>
<tr>
<td></td>
<td>Mozilla CSS Animations</td>
<td>jQuery Tutorial</td>
</tr>
<tr>
<td></td>
<td>Keyframe Animations</td>
<td></td>
</tr>
</tbody>
</table>
- Lab 12
- Video 12: CSS animations and Motion Design
  - Examples
  - Animatable functions
  - Transitions
  - @keyframes
  - In Class:
    - Goal: Highlight your mastery of web design with animations
    - Create a new directory and webpage, this page must include
      - One button with animation
      - 3 <div> that change on :hover
      - An image with motion effects
  - After Class:
    - Finish lab 12

- Lab 13
- Video 13: JS libraries
  - Using a library
  - Modifying and creating scripts
  - jQuery, P5, etc
  - In Class:
    - Goal: Implement a JavaScript library on a new webpage
    - Simple: Lightbox
o Complex: Choose a library of your own and inform your instructor, experiment implementing and using this library on your lab13 page(s)

o After Class:

  o Finish Lab 13
<table>
<thead>
<tr>
<th>Week 10</th>
<th>LAB 14: Destination: Final</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 2/5</td>
</tr>
<tr>
<td></td>
<td>Notes: A tour of final projects past</td>
</tr>
<tr>
<td>Readings:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chris Jackson</td>
</tr>
<tr>
<td></td>
<td>Angelina Liu</td>
</tr>
<tr>
<td></td>
<td>Soren Chargois</td>
</tr>
</tbody>
</table>

FINAL PROPOSAL
Lab 14

Video 14: A tour of final projects past

- In Class:
  - As you think about your own final project, think about the purpose of the site, the audience, and the message you are trying to convey through your design. What features might you implement in CSS to help convey those ideas? Do your design and content correlate? How might the design of an NGO site differ from a tech startup? From an artistic portfolio site? From a web designer’s showcase? A sports team site? A recipe site? Travel and recommendation site? Look for examples that of other site that are similar to what you hope to do in concept, purpose, look and feel.

- After Class:
  - Finish Final Project Proposal
<table>
<thead>
<tr>
<th>Week 11</th>
<th>LAB 15: The .gif that keeps on giving</th>
<th>LAB 16: You are being watched</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 1/5</td>
<td>Difficulty: 1/5</td>
</tr>
<tr>
<td></td>
<td>Notes: Photoshop Walkthrough</td>
<td>Notes: What are analytics? Your Data Matters</td>
</tr>
</tbody>
</table>
• Lab 15
• Video 15: Photoshop Walkthrough
  o Install if you haven't already
  o Introduction to platform
    o Several paths to same solution
    o Workspaces
    o Tools
    o Layers
      • Charting motion path
    o Timeline
    o Exporting

• In Class:
  o Goal: Create and upload an animated .gif of your own design
  o Go to: https://software.duke.edu/node/272 and download Adobe Photoshop, free to Duke students
  o Referring to the lecture video, complete the following steps:
    o Create a background
    o Import or create an image with transparency (your character)
    o Use layers to duplicate your character and create the motion path
    o Convert layers to frames through the timeline feature
    o Export your new .gif and upload it to a new page

• After Class:
○ Finish Lab 15

▪ Lab 16
▪ Video 16: What are analytics? Your Data Matter.
  ○ What (and where) is your (meta)data?
  ○ Who wants it and why?
  ○ In Class:
    ○ Create a Google Analytics account
    ○ Fill out the relevant information.
      ▪ Account Name: “First name Last name Web Comm 2018”
      ▪ Website Name: “First name Last name”
      ▪ URL: people.duke.edu/~yourNetID
      ▪ Complete the remainder of the information. You may decide to choose an industry that is most similar to your final project. When you’re finished, click “Get Tracking ID” and review the Terms & Conditions.
    ○ Add the embed code to the header of your midterm homepage (index.html). Google Tag Manager recommends you add the script right after your opening <head> tag.
    ○ Upload your updated index.html file to FileZilla. To see if your code is tracking, visit your site in another tab or on mobile.
View Reports > Real Time > Overview to see if website visits are being tracked.

- After Class:
  - Finish Lab 16
<table>
<thead>
<tr>
<th>Week 12</th>
<th>LAB 17: Searchin' for Somethin'</th>
<th>LAB 18: The Great Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficulty: 2/5</td>
<td>Difficulty: 1/5</td>
</tr>
<tr>
<td>Notes:</td>
<td>How do Search Engines Search?</td>
<td>Notes: Ruling your Domain</td>
</tr>
</tbody>
</table>
Lab 17

Video 17: How do Search Engines Search?

- Indexing and algorithms
- Robots.txt
- Key term padding
- Rankings and conversions

In Class:

- Create a **Custom Search Engine** using Google CSE.
- Enter in the URL etc. for your site or another site on the web if indexing doesn't work properly for your own site. Note the use of the * to index the whole site.
- Test your search engine using the public URL generated after your search engine is created. (Search on something you know is there!)
- Add it to your semester website using the embed code provided. Remember to add it in a place where you would want the search bar to show up. Upload and test from the web server.

After Class:

- Finish lab 17

Lab 18

Video 18: Ruling Your Domain

- Buying a domain and hosting services
- Migrating your site
- In Class:
  - Q&A on Migration
  - Final work
- After Class
  - Final Project
  - Course Evals
<table>
<thead>
<tr>
<th>Week 13</th>
<th>FINAL OPEN LAB HOURS</th>
<th>THANKSGIVING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 14</td>
<td>BREAK</td>
<td>FINAL OPEN LAB HOURS</td>
</tr>
<tr>
<td>Week 15</td>
<td>FINAL</td>
<td>PROJECT</td>
</tr>
</tbody>
</table>
- **FINAL PROJECT OPEN LAB HOURS AND THANKSGIVING BREAK**
- **FINAL PROJECT DUE**
  - Full website uploaded to your web folder (FTP folder)
  - Archive copy of entire site (class folder for the semester) on Duke Box
Appendix C: General Code

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Student Demo Site</title>
  <meta name="author" content="Luke C. LeGrand">
  <meta name="description" content="The Living Web Student Demo Site">
  <link rel="stylesheet" type="text/css" href="main.css">
</head>
<body>
  <h1>This site is still being built! </h1>
  <h2>Check back soon!</h2>
</body>
</html>
```

Figure 12: Student Boilerplate Example
1: Using comments like this one below, annotate the function of EACH LINE (or group if all the same) of the HTML below. If you don't know what something means... LOOK IT UP! -->

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <!--EDIT THESE TAGS-->
  <title>TITLE</title>
  <meta name="author" content="YOUR NAME HERE">
  <meta name="description" content="YOUR DESCRIPTION HERE">
  <link rel="stylesheet" type="text/css" href="main.css">
</head>
<body>
<header> </header>
```

Figure 13: Student Annotation Exercise
Appendix D: Website Documentation

To access the full site archive, including updates visit:

- https://github.com/lcl20/TheLivingWeb
- http://people.duke.edu/~lcl20/

Figure 14: The Living Web Project Site
Figure 15: Student Demo Site based on Syllabus
Bibliography


