

MMP 350 CSS Week 3

Table of contents

Goals for Week 3	2
Discussion: Application of Tidwell Takeaways to Websites.....	2
Review: Inline elements, block elements	2
Key concepts.....	2
Review: Margins and padding.....	3
Key concepts.....	3
Review: Positioning.....	3
Readings and Break.....	3
Responsive Design	3
4 Types of Grids and When Each Works Best	3
How to Choose the Right Grid	3
The Grid System	3
Wireframing.....	3
Layouts in CC, CS6	3
Fixed, Fluid and Elastic	4
Adaptive Versus Responsive.....	4
Progressive enhancement.....	4
Wireframe tools	4
Exercise 1: Wirefy	4
Purpose	4
Background	4
Step 1	4
Step 2	4
Step 3	5
Exercise 2: Moqups.....	5
Step 1	5
Step 2: Sign up for Free Account.....	5
Step 3	6
Step 4: Edit the header	6
Step 5: Add an icon to the header	7
Step 6: Insert image	8
Step 7: Add a link	8
Step 8: Preview the Result.....	9
Step 9: Create a login page.....	9
Step 10: Modify the Login Link on your home page to launch the login page.	10
Step 11: Add an Interaction to the Submit button	11
Step 12: Save your Project.....	11
Step 13.....	12
Responsive / Adaptive Design.....	12
Class Exercise: Analysis of an Adaptive Webpage.....	12
Read SDLC Overview.....	13
Homework.....	13

Goals for Week 3

1. Schedule Website Critiques
2. Introduction of Student Instructor
3. Presentation by Philip Cohen
4. Tidwell review – what to consider when critiquing a site
5. HTML review
6. Readings: Responsive design and grids
7. Wire-framing and mockups
8. Moqups exercise
9. Reading: SDLC
10. Discussion: “Waterfall” versus “Agile”
11. Review of Homework

Discussion: Application of Tidwell Takeaways to Websites

Jennifer Tidwell identifies the follows issues to consider when assessing websites:

- Use of color and fonts
- Use of space
- Use of angles and curves
- Use of texture and rhythm
- Use of multi-media assets
- Use of visual motifs

Review: Inline elements, block elements

All elements are block elements, in the sense that the basic unit of space in HTML is a rectangle. Block elements have line breaks before and after, inline elements do not.

Please see the files `inline_and_block_elements_review.html` (plus `inline_and_block_elements.css`) for examples.

Key concepts

- Inline elements have a line-height property
- Block elements have a height property and a line-height property.
- Inline-block elements can have either a height or inline-height property.
- Clear property. Values: left, right, none, both. Adds a break before (left), after (right) the element that has this property.
- Float property. Values: left, right, none. Elements with this property are left aligned, right aligned or not floating.

Review: Margins and padding

Please see the files margins_and_padding.html, plus margins_and_padding.css for examples.

Key concepts

- Margins add space outside an element, ~~and increase the width of their containing element.~~ It does not increase the width of the containing element, though the element with the padding may become wider than its parent container.
- —
- Padding adds space within an element. It does not increase the width of the containing element, though the element with the padding may become wider than its parent container.
-
- Stylings are designated clockwise: top, right, bottom and left, e.g. `margin:10px 10px 10px 10px`
- When two values are set the first applies to the top and bottom, the second to the right and left, e.g. `padding: 10px 5px`
- When one value is set, the top, left, bottom and right properties take this value, e.g. `padding: 10px.`

Please see the files inline_and_block_elements.html (plus inline_and_block_elements.css) for examples.

Review: Positioning

<http://www.barelyfitz.com/screencast/html-training/css/positioning/>

Readings and Break

During this time the Student Instructor will review your Cloud9 setup.

Responsive Design

<http://alistapart.com/article/responsive-web-design>

4 Types of Grids and When Each Works Best

<http://vanseodesign.com/web-design/grid-types/>

How to Choose the Right Grid

<https://vanseodesign.com/web-design/grid-choices/>

The Grid System

<https://www.thegridsystem.org>

Wireframing

<https://www.wirify.com>

Layouts in CC, CS6

<https://helpx.adobe.com/indesign/using/alternate-layouts-liquid-layouts.html>

Fixed, Fluid and Elastic

<https://www.smashingmagazine.com/fixed-fluid-and-elastic>

Adaptive Versus Responsive

<https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/>

Progressive enhancement

<http://alistapart.com/article/understandingprogressiveenhancement>

Wireframe tools

What is wireframing?

A wireframe is a schematic diagram or blueprint for a website. Wireframes are used by designers to visualize, and thereby understand, the relationship between screen elements.

Exercise 1: Wirefy

Purpose

To use the wirefy tool to examine the structure of webpages.

Background

Wirefy is a [bookmarklet](#). A bookmarklet is a bookmark stored in a web browser that contains JavaScript commands that add new features to the browser.

Wirefy lets you turn any webpage into a wireframe.

Step 1

Go to wirefy.com:

<https://www.wirify.com/about/wirify-overview-and-why/>

Step 2

Make your bookmarks toolbar active:

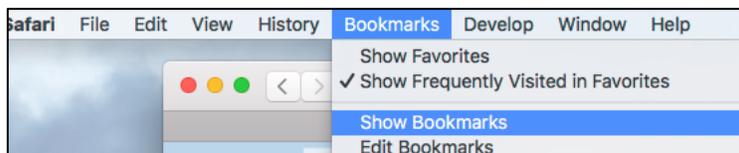


Figure 1: Activating the bookmarks toolbar in Safari.

Drag the Wirefy link (Wirify by Vokside) to your bookmarks toolbar (figure 2)

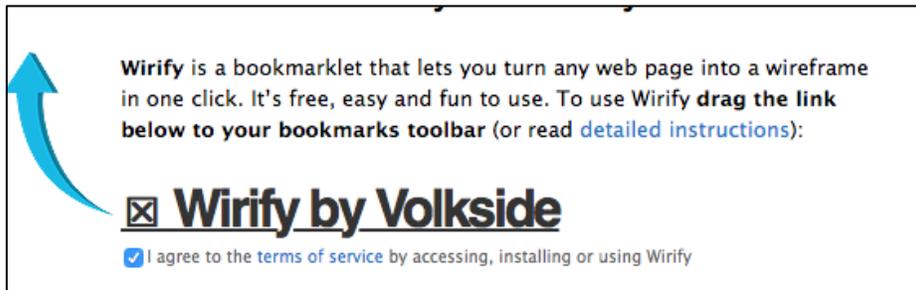


Figure 2

Step 3

Go to a favorite website and click on the Wirify by Volkside link on your bookmarks toolbar (figure 3).

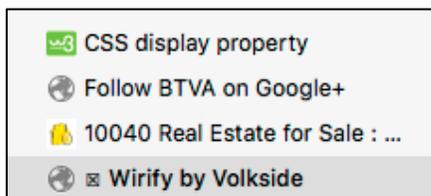


Figure 3

Exercise 2: Moqups

Purpose

To explore common features of the most common collaborative wireframing tools, moqups.com

Step 1

Go to moqups.com

<https://app.moqups.com/edit/page/ad64222d5>

Step 2: Sign up for Free Account

To activate your free account, click on the [Free Plan](#) link at the bottom of the moqups.com homepage (figure 4).

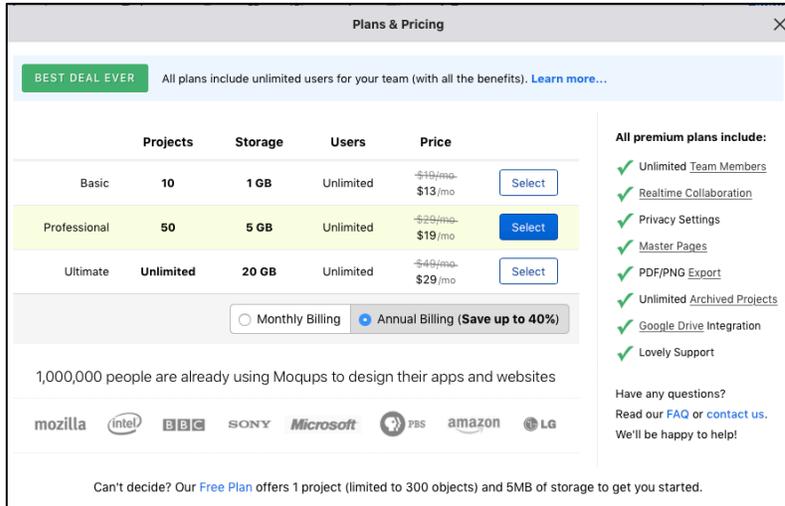


Figure 4: The moqups.com homepage

Step 3

Once you have created a project, review the instructions on the main page and then delete them.

Step 4: Edit the header

Edit the main heading to say [Your Name] Portfolio (figure 5)



Figure 5: Adding a header to a moqups.com page

Double click on the text to change its formatting (figure 6)

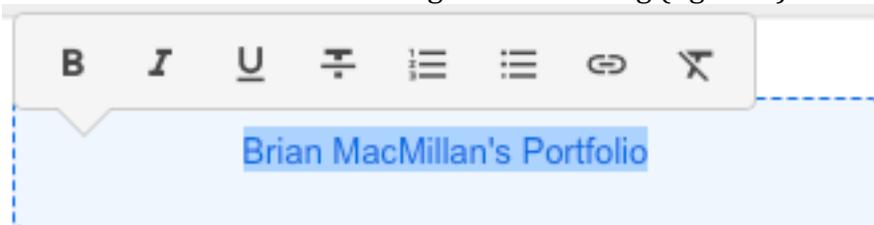


Figure 6: Applying basic text formatting



Figure 7: Changing to Format mode

Change the color and font size of your heading text. The settings in figure z result in the output in figure 8.

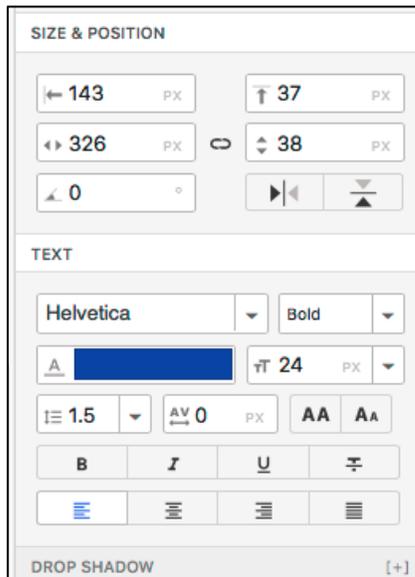


Figure 8: The text format in moqups.



Figure 9: The results of the text formatting decisions illustrated in figure 8.

Step 5: Add an icon to the header

Click on the [Icons](#) icon on the left hand panel (figure 10).



Figure 10: The Icons menu item

Select an icon from the fly-out panel that appears, for example the Android icon (figure 11).

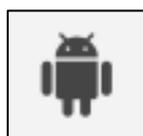


Figure 11: The Android icon from the moqups icon fly-out panel.

Drag the icon to your header and format the results. They should look something like figure 12.



Figure 12: header with icon

Step 6: Insert image

Click on the Images menu item on the left hand menu and then [Upload images](#) link to add an image to your project (figure 13).

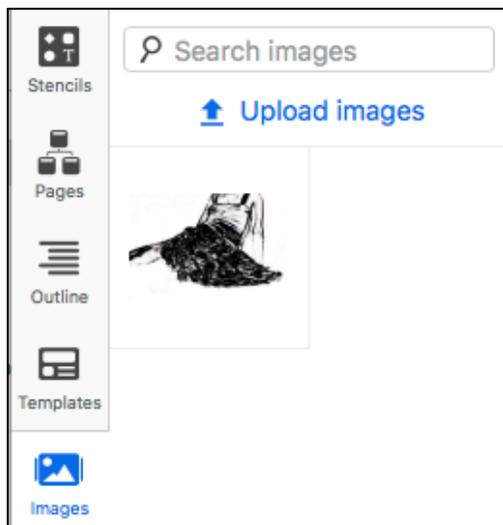


Figure 13: Once you have uploaded an image it will appear in the images palette Only jpgs, png and gif formats are supported.

Step 7: Add a link

Click on the [Stencils](#) icon to get access to basic HTML elements (figure 14)



Figure 14: the [Stencils](#) menu item.

Click on the Link menu item and drag a link to your page (figure 15).



Figure 15: The [Link](#) menu item

Double click on the link and change its text to read [Login](#).

Step 8: Preview the Result

Click on the [Preview](#) button in the upper right hand corner of your screen to preview your work as if it were a functioning webpage.



Figure 16

Click on the [Edit Project](#) button to exit from preview mode.

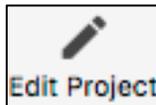


Figure 17

Your page should now look something like what is illustrated in figure 18.



Figure 18: Moqup page with icon, header, image and link.

Step 9: Create a login page

Click on the [Pages](#) icon and then on the [New page](#) option (figure 19). Name the page [Login](#). While you're at it, rename your first page to [Home Page](#).

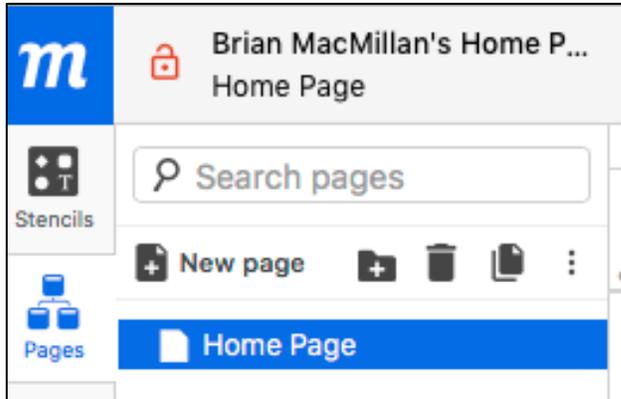


Figure 19: The pages icon.

Now edit the login page so that it looks like figure 20. This page will use Heading, Label, Text Input, Text Area, Checkbox and Button elements.

 A screenshot of a login form. At the top left is a heading 'Login'. Below it are two rows of labels and text inputs: 'User Name' with an input field containing 'Email Address', and 'Email address:' with an input field containing 'Email Address'. Below these is a section titled 'Terms and conditions' followed by a paragraph of Lorem ipsum text. At the bottom left is a checkbox labeled 'Agree to terms and conditions', and at the bottom right is a 'Submit' button.

Figure 20

Step 10: Modify the Login Link on your home page to launch the login page.

Navigate to your home page, select the [Login](#) link and then click on the [Interactions](#) button on the upper right hand corner of the screen. It has a lightning bolt icon (figure 21).

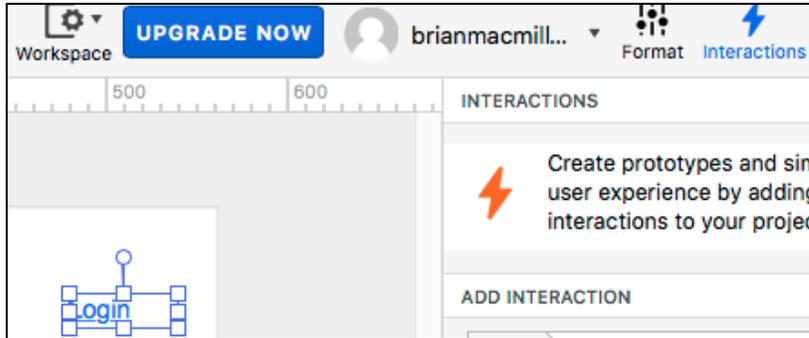


Figure 21

To add an interaction you need to select an event (Click), and action (Go to Specific Page) and a target page (Login) (see figure 22)

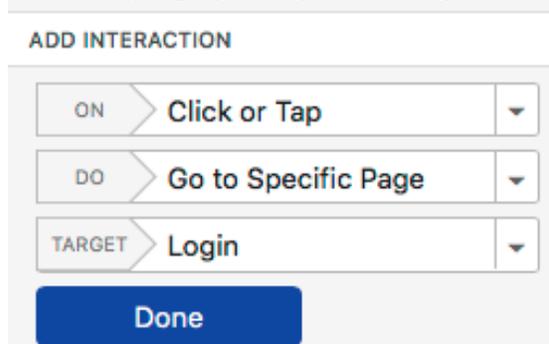


Figure 22

Step 11: Add an Interaction to the Submit button

Add an interaction to the Submit button that returns the user to the Homepage.

Step 12: Save your Project

To save your project click on the main menu, which is activated by clicking on the mockups logo on the upper left hand corner of the page (figure x) and selecting Save. Note that unless you pay, you can only save one project and you cannot export your work.

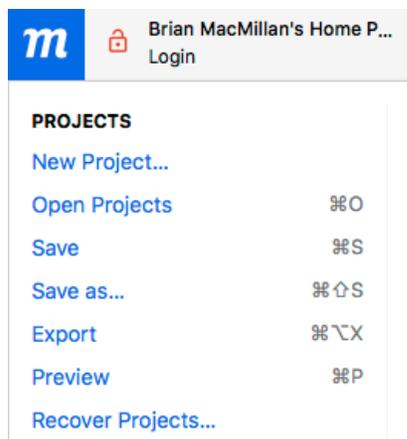


Figure 23: the Mockups Main menu.

Step 13

If you finish this exercise before the rest of the class, either read the articles linked to at the bottom of this document, or begin your homework assignment.

Responsive / Adaptive Design

Responsive design is the practice of designing websites that change style and layout to fit a variety of devices with different dimensions, screen densities and functionalities to create an optimal viewing experience based on a single template. In responsive design, the rules will move fluidly between sizes, applying new dimensions to every dimension.

Like responsive design, **adaptive design** changes based on screen size and density, but adaptive design generally pinpoints specific breaking points or thresholds and creates designs for those points.

Both concepts are rooted in the idea of **progressive enhancement**, the idea that websites should be designed to provide the basic content to any device, while progressively introducing enhanced design possibilities for modern devices.

Class Exercise: Analysis of an Adaptive Webpage

Go to <http://www.houstonchronicle.com>

Change the size of the screen. What happens?

Activate the Inspect Element tool (figure 24)

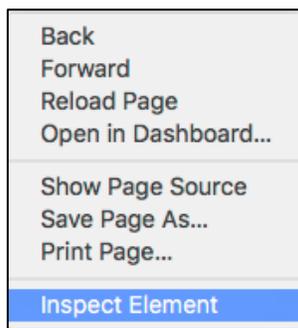


Figure 24: Right click to inspect elements of a web page.

If you are using Firefox change the location of the Inspect Element frame clicking on the left hand button, illustrated in figure 25, below.

Question 1: What happens when you resize?



Figure 25. In Firefox you use the icon on the left to move the position of your inspection frame.

Question 2: What is the width of the header element?

Question 3: Why a percentage and not an absolute number?

Question 4: How does the width of the #content element vary as screen width changes?

Question 5: The site uses a number of common libraries. Which ones?

Answer 1: The width of the content adjusts dynamically.

Answer 2: 100%

Answer 3: Liquid layout

Answer 4: It doesn't change. It is fixed at 93.75%.

Answer 5: jQuery (1.11.1), Google Analytics, Chartbeat, Hearst.js

Read SDLC Overview

https://en.wikipedia.org/wiki/Systems_development_life_cycle

Discussion of the Project Life Cycle

Homework

Create a features table based on the handout *Examples of features tables and mockups*. Submit it before class in pdf, Word or Open Office (odt) format. Do not submit it using pages.

Read the SDLC Quick Guide

https://www.tutorialspoint.com/sdlc/sdlc_quick_guide.htm

Read the twelve principles of Agile development

<http://agilemanifesto.org/principles.html>

Read about the difference between “Waterfall” and “Agile” requirement gathering

<https://dzone.com/articles/why-do-we-capture-requirements-differently-between>

Download Bootstrap

<https://getbootstrap.com/docs/3.3/getting-started/>

Download Bootstrap Examples

<https://github.com/twbs/bootstrap/archive/v3.3.7.zip>

Review the Bootstrap grid system

<https://v4-alpha.getbootstrap.com/layout/grid/>