IST 263

# Lab 10

## This lab covers:

1. Responsive Design
   1. Viewport
   2. Flexible Image Sizes
   3. Flexbox Flexible Grids
   4. Media Queries

## SETUP

Create a folder in your Github repository called lab10. Place a copy of your latest skills, bio, contact pages and stylesheet in the new folder. Please make sure to keep the same file names in the new folder.

## overview

The goal of this lab is to help you understand how to make a website layout and work well on Mobile, table and desktop devices.

## Viewport

1. Pull out your cell phone, load this lab's index page and note what the home page looks like currently. Or you can use the simulator in the web browser to view your site by right clicking on your page -> choosing inspect -> pressing the screens icon and using the drop downs to select a phone screen.

To make sure cell phones don't try to shrink your entire webpage down to fit in the viewport we are going to add a meta tag. Remember that content goes in the body element. Data about the page goes in the head element. Therefore, our meta tag is going inside the head element. We already have one meta tag on our pages. This second meta tag can go on the line under it.

1. Add the viewport meta tag on all three of your portfolio site's pages - index, skills and contact. Consult this unit's slides if you need help remembering the syntax.
2. Let's check to see if your viewport meta tag is working. Reload your portfolio site on your cell phone and note the change after adding the viewport meta tag.

## Flexible image sizes

My index page image is too big to fit on my cellphone screen and is disproportionately large now that I've put the viewport meta tag on the page. We are going to add a max width property to this image. This will make sure the image fits without scrolling left to right.

1. Add a class to the img tag on the index page. You can call it anything you want.
2. Add a class selector, the max-width property and 100% value to your stylesheet. You don't need to load this on your phone, you can check to see if this one worked by scaling down your browser window.

## flexbox flexible grids

Next, we are going to work on the flexbox grid on the contact page. If you had troubles with the flexbox layout last week, please let me know and I will help you fix the CSS. It might be hard to get the next direction working if your flexbox layout from last week is off.

If you resize your browser down to cell phone size, you will see that the form and map get squished. We can easily fix this with the following…

1. Locate the parent container holding both the form and map. Assign a property of flex-wrap and value of wrap to that container.

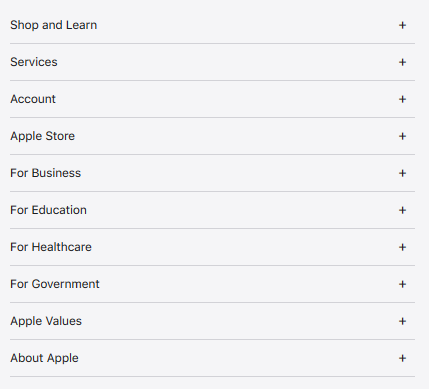
If you map is overflowing the container, we can fix that. Flexbox automatically resizes all it's child elements. If the iframe is a direct child of the flexbox container the map will resize. If you have a child div tag (or other semantic tag) around the iframe get rid of it and move that element's class to the iframe tag.

## media queries

For our media query practice, you are going to create stacking navigation for smaller viewports. If you go to apple.com you will see the footer navigation looks like this:



When we size the browser down, the navigation does the following:



Go the apple.com site, scroll down to the bottom and see for yourself.

We're going to replicate stacking navigation on our main navigation. We will only have links running across the top from left to right on desktop size viewports. On tablet and cell phone viewports we will display stacking links.

Your gut might be to create two totally different navigations and turn them off and on with the media query. Resist the temptation to do this. We can style the same HTML with different CSS to accomplish this task. If we chose to duplicate the navigation, we'd double our work every time we had to change it.

1. You already have the navigation working for the desktop size. The next step is to create a media query to hold the styles for tablet and smaller size viewports. Add the media query to your **external** stylesheet since it's going to be used for the nav on all three pages. Consult this week's slides for media query syntax. Make a judgement call based on the data about tablet widths to decide what size you want to use for your breakpoint.
2. We are going to size the links inside our navigation to 100% when the viewport is below the breakpoint. Put the style to change the width inside your media query. Use a **descendant selector** (don't use a class) to style only the links inside the nav tag.   
     
   Note: The a tag is inline (not block). By default, inline elements only take up the space from left to right that they need to display their content. To make an inline tag take up 100% of the horizontal space we need to convert it to block. Add the property display and the value block to the descendant selector you created above.
3. What about those pesky | pipe characters? We don't want them in our stacking navigation. The span tag is perfect for this. Span is an inline container element, similar to div which is a block container element. Wrap the | character in span tags like this <span>|</span>.
4. This is where the display: none property and value come in handy. You don't want the pipe character to display when the viewport size is small. Add a selector and display: none to the stylesheet to make the pipes disappear when the viewport is below your breakpoint.
5. Last but not least… If your nav items are sitting one right on top of the other, add some space around them.

This is what my finished nav looks like at tablet and smaller size:



This is what my finished nav looks like at desktop computer size:



1. Make sure your navigation is working the same way on all three of your pages.

## What will You hand IN?

Create a word document, pdf or use the "write submission" option in blackboard to provide the following:

1. Submit the urls for the index page.

<https://mbpaninski.github.io/ist263/lab10/index.html>

1. Validate the index and contact page and submit the validation links.

https://html5.validator.nu/?doc=https%3A%2F%2Fmbpaninski.github.io%2Fist263%2Flab10%2Findex.html

1. Submit answers to the following:
   1. What questions did you have about the lab? What didn't you fully understand?

I had trouble with my flexboxes and wrapping them correctly in the viewport.

* 1. What was the hardest part of the lab?

Using the media query

* 1. Rate your comfort level with this week's topics.  
     1 ==> I can do this on my own and explain how to do it.  
     2 ==> I can do this on my own without any help.  
     3 ==> I can do this with help or guidance from others.   
      If you choose this level, please indicate HOW this person helped you.

I had trouble with how my flexboxes I created last week were wrapping and Eleni helped me figure out where the mistake was in my code.   
4 ==> I don't understand this at all yet and need extra help.