{ } FE Dev Homework

Unit 18 | Iterate on Your Portfolio and Code Responsive Design Elements

~13.5 hours | Individual | 6 Parts

(Assignment Due 19.3)

Background

Last week, you started laying the groundwork for your UX/UI portfolio website during the Unit 16 Challenge by building the wireframe you designed during Week 16. This week we will continue to work on your portfolio by applying CSS Flex, CSS Grid, and media queries to make your UX/UI portfolio responsive. In this assignment, you will use CSS Flex to build a layout showcasing your UX skills. You will also use CSS Grid to create content for the case study you generated during Week 16. On top of this, you will continue to polish your UX/UI portfolio website's CSS to make your site more attractive and employer-competitive.

User Story: As UX/UI professionals, accessibility is critical for your portfolio page. You must not operate under the assumption that your site will only be browsed via desktop. In today's busy world, it's entirely possible that your target audience will not check out your work by desktop and browse on their mobile devices instead. This week is your chance to illustrate your front-end development skills by building flexible content that shows off your work and makes your UX/UI portfolio responsive.

Note: If you are using a content management system like Wix or Squarespace to build your portfolio, you MUST still complete this assignment for submission.

Workflow Tips

The following general developer workflow tips help speed up your development process when building a webpage:

- Start by coding the skeleton of the wireframe first. Use outline: **1px solid red;** to see what you're doing on your page with your CSS. Remember to use the Web Inspector to preview changes to your CSS.

- Style your content inside your wireframe.

- Google a lot! There are many nuances to front-end development, and you will undoubtedly encounter problems when practicing coding. Using Google to search for answers is a normal part of all web developers' workflow.

- Create an account at https://stackoverflow.com/. Stack Overflow is a community forum for developers and a great resource to answer all your development questions.

Resources

Pull up these resources as you are coding to help enforce your code syntax and troubleshoot issues via documentation.

- Best practices for mobile design
- Media Query Tutorial By MDN
- MDN Flexbox basics
- MDN basic concepts of Grid layouts

If you did not complete the Unit 17 homework or were not happy with the outcome, feel free to use this starter template. Make sure you style it to match your visual design from Unit 16.

Be sure to read through this document in its entirety before beginning work.

Objectives

Here is what your instructor wants to see that you're capable of:

- ❏ Practicing redlining a page layout to define HTML/CSS structure.
- ❏ Applying CSS Flex to improve a Responsive webpage.
- ❏ Utilizing and practicing CSS Grid to generate a web page layout.
- ❏ Customizing media queries to make your UX/UI portfolio breakpoints responsive.
- ❏ Using CSS Flex to build a layout.

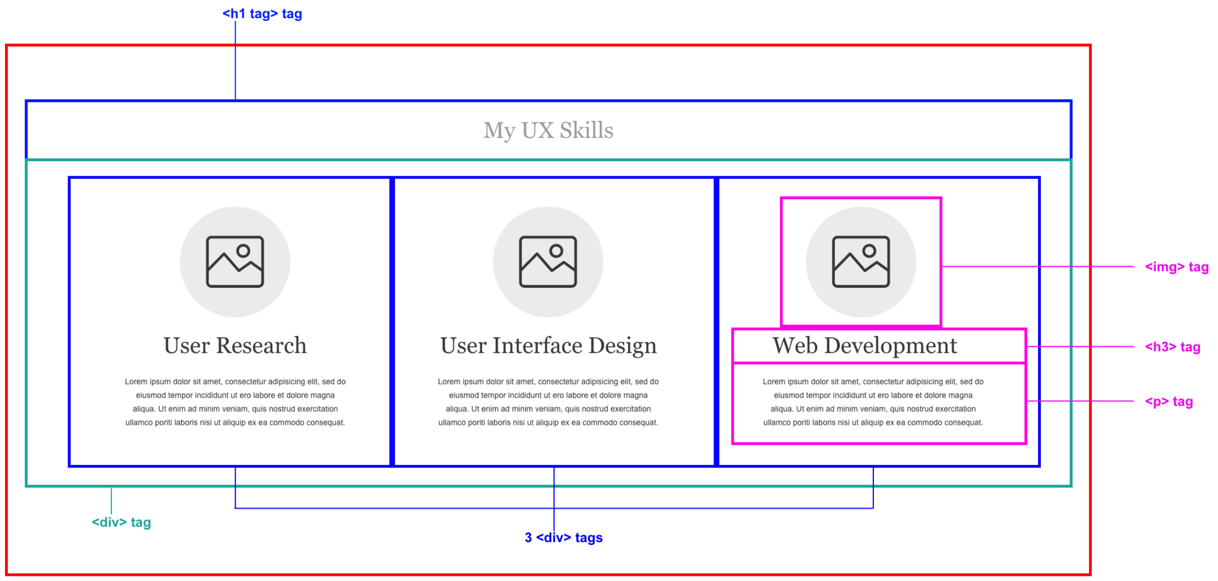
Instructions

Step 1: Redline Your Portfolio's "My UX Skills" and "My Work" Sections (1-2 hour)

During the first part of this homework, you will continue to think like a developer by redlining the two sections we will be building using CSS Flex and CSS Grid.
Note: If your design does not match the wireframe examples, redline accordingly. Remember, this is your portfolio. You have creative freedom over the visual design of your website. If you are not confident in your front-end development skills, follow the basic steps here to create a wireframe. Then, style it to visually match your design.

1. Redline your "My UX Skills" section. Your redlined design should contain the following tags:
    a. A <section> tag
    b. A <h1> tag
    c. A <div> tag containing the following tags:
        i. Three <div> tags that will serve as our flex children. Each <div> should contain the following tag:
            1. A img tag containing a skill you possess.
        ii. A <h1> tag containing the title of the skill.
        iii. A <p> tag describing how you apply your skillset

Below is a sample of a redlined "My UX Skills" section:



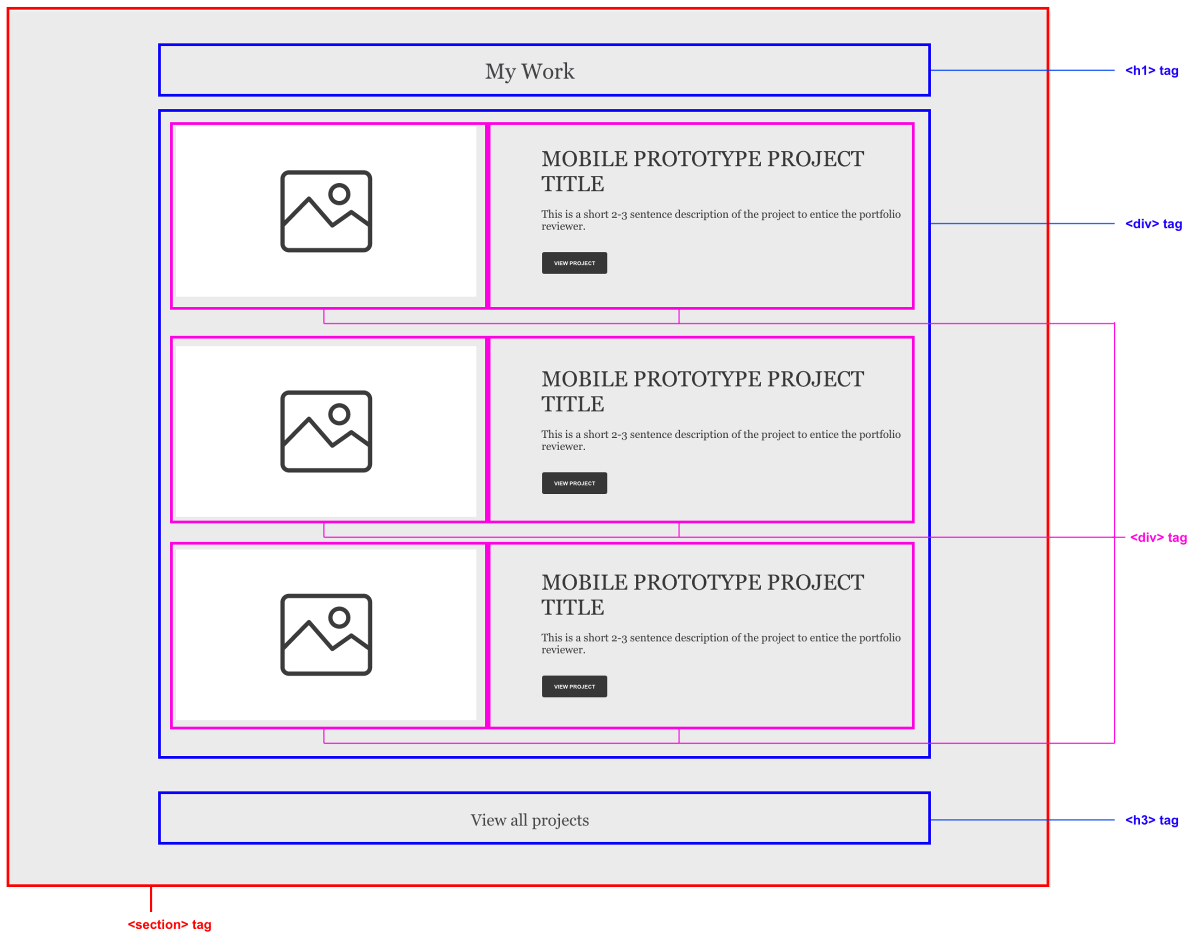
<h1 tag> tag
My UX Skills
User Research
User Interface Design
Web Development
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut ero labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco poriti laboris nisi ut aliquip ex ea commodo consequat.
<img> tag
<h3> tag
<p> tag
<div> tag
3 <div> tags

2. Redline your “My Work” section. Your redlined design should contain the following tags:
    a. A <section> tag
    b. A <h1> tag
    c. A <div> tag that contains the following tags:
        i. Two <div> tags, side-by-side:
            1. The first <div> should contain an <img>.
            2. The second <div> should contain a <h4> tag, a <p> tag, and a <button> tag.
        ii. Two more <div> tags, side-by-side.
            1. The first <div> should contain an <img>.
            2. The second <div> should contain a <h4> tag, a <p> tag, and a <button> tag.
        iii. Two more <div> tags, side-by-side.
            1. The first <div> should contain an <img>.
            2. The second <div> should contain a <h4> tag, a <p> tag, and a <button> tag.

    In total, we should have three rows of two.

    d. A <h3> tag

Below is a sample of a redlined "My Work" section:


My Work
<h1> tag
MOBILE PROTOTYPE PROJECT TITLE
This is a short 2-3 sentence description of the project to entice the portfolio reviewer.
VIEW PROJECT
<div> tag
<div> tag
View all projects
<h3> tag
<section> tag

Now that we have visually mapped out your layout, let's create a repo and start coding our sections!

Step 2: Create a New GitHub Repository for This Unit's Homework (30 minutes)

1. Before we start coding, create a new repository on GitHub and name it "UX_UI_HW_18." If you need a reminder of how to create and clone a repository, please reference last week's homework.
2. Clone the blank repository to your desktop using GitHub Desktop.
3. Copy and paste your homework files from UX_UI_HW_17 into UX_UI_HW_18. You want to be able to show how your code abilities have grown during the course of the code section. Having multiple versions will allow you to show employers how your skills have grown.

4. Update your UX_UI_HW_18 repository with the new files.
5. Publish your site on GitHub pages. Click Settings on your repository's homepage.
6. Scroll down to the GitHub Pages area in Settings and select the master branch.
7. Check the URL after the page refreshes. You're ready to code!

Step 3: Code the HTML for Your "My UX Skills" and "My Work" Sections (1-2 hours)

Code the "My UX Skills" HTML

1. Create a <section> tag with the class of mySkills. Inside it, create the following tags:
    a. An <h1> tag that contains the text "My UX Skills"
    b. A <div> tag with the class of flexContainer that will serve as our flex container. This div will contain the following tags:
        i. Three <div> tags that will serve as our flex children. Each div should contain:
            1. An <img> tag. Pick or create an icon that will fit the skill.
            2. An <h3> tag that contains text to describe the skill.
            3. A <p> tag containing a paragraph about how you use the skill.

That's it for the HTML of our "My UX Skills" section. Let's move on to the "My Work" section.

Code the "My UX Skills" HTML

1. Create a <section> tag with the class of myWork. This section should contain the following tags:
    a. An <h1> tag containing the text "My Work".
    b. A <div> tag with the class of workGrid. This div should contain the following tags:
        i. Two <div> tags. The farthest first <div> will contain an <img> tag and the second will contain:
            1. An <h4> tag
            2. A <p> tag
            3. A <button> to view the project
        ii. Two more <div> tags underneath the other two. The farthest first <div> will contain an <img> tag and the second will contain:
            1. An <h4> tag
            2. A <p> tag
            3. A <button> to view the project
2. Under our grid, create an <h3> tag tag that contains the text "View all projects."

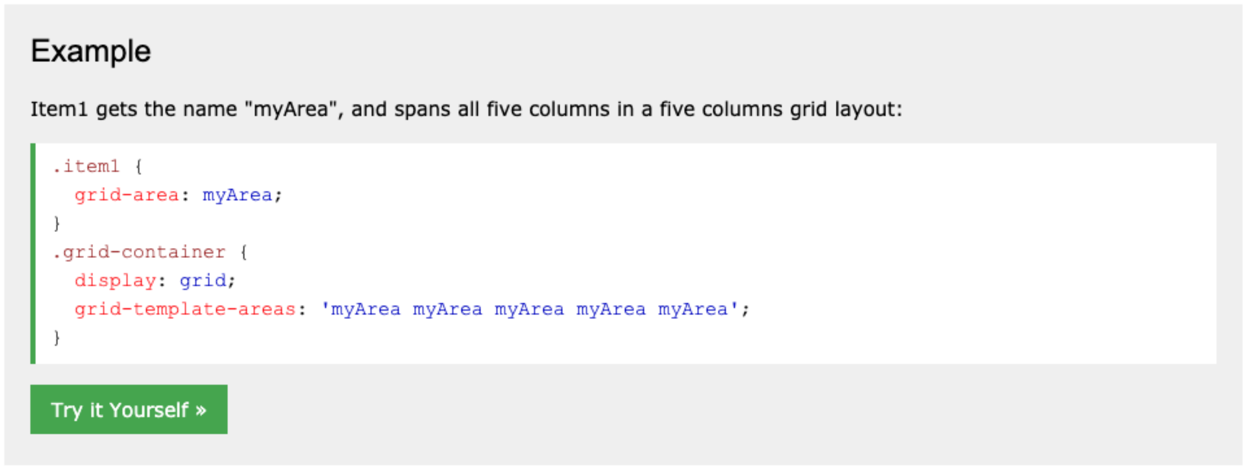
Step 4: Code the CSS Style for the “My UX Skills” Section Using CSS Flex (4-6 hours)

1. If your design differs from the HTML content we constructed earlier, create HTML tags to match your redlined design file for your CSS flex container. This will vary from design to design; each design is different.
2. The first step to creating your “My UX Skills” section is to set a flex container. Set display: flex to your <div> with the class of .flexContainer.
3. Now that you have a flex container created, we need to set the widths of the three <div> tags wrapped inside of it.
    a. Apply width 33% to each <div> wrapped inside to create a fully responsive row. *Note: Don’t forget to apply margin and padding to these <div> tags, creating a layout that is evenly spaced.*
4. Write an HTML comment above your section tag that says:
    <!-- My UX Skills section coded using CSS flex -->

5. Now that you have the basic structure and style created for your “My UX Skills” section, your job is to style it to match your visual design! Here are a couple of tips to get you working:
    a. Use flex-direction on your flex container to set it to display as either a column or a row.
    b. If you are using the placeholder content for this section, fill it with your own content from the design you created during the Week 16 homework.
    c. If your design differs from the placeholder content, add in your HTML elements to make it match your design and style accordingly.

Step 5: Code the CSS Style for the “My Work” Section Using CSS Grid (4-6 hours)

1. If your design differs from the HTML content we constructed earlier, create HTML tags for your CSS Grid. This will vary from design to design; each design is different. *Note:* For a comprehensive guide on how to use CSS Grid, check out this tutorial.
2. Apply display: grid; to your div with the class of workGrid to convert this <section> tag into a grid container.
3. Apply named grid areas to your divs wrapped inside your grid container.


Example

Item1 gets the name "myArea", and spans all five columns in a five columns grid layout:

```
.item1 {
  grid-area: myArea;
}
.grid-container {
  display: grid;
  grid-template-areas: 'myArea myArea myArea myArea myArea';
}
```

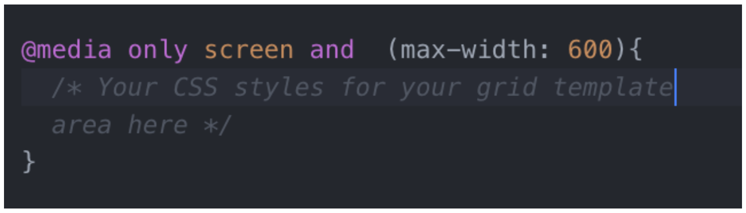
Try it Yourself »

4. On your grid container, create grid-template-rows and grid-template-columns to match the visual design of your page. Keep in mind: you can use percentages, pixels, or fraction units to accomplish this.
5. Build out your grid-template-areas in your grid container to match your design.
6. Apply CSS styles to your CSS Grid to make sure the visual design matches the wireframe that you created during the Unit 16 homework.
7. Create an HTML comment above your "My Work" section that reads:
   <!-- My Work Section Constructed using CSS Grid -->

Step 6: Code a Mobile Media Query to Make Your Design Responsive for Mobile Devices (1-2 hours)

1. Write a mobile media query at the bottom of your style.css file. Set this mobile media query to take effect at 600 pixels or less using max-width.

   Scroll to the next page.

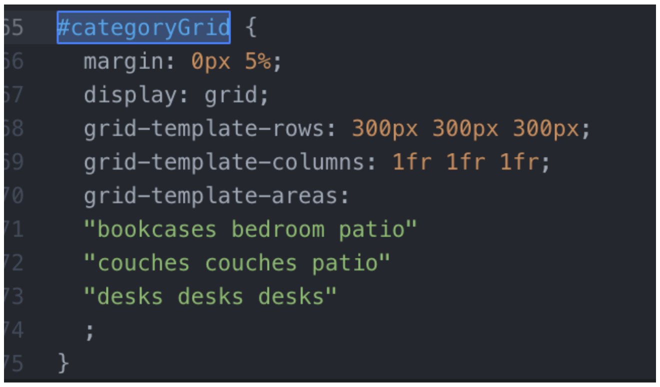


```
@media only screen and  (max-width: 600){
  /* Your CSS styles for your grid template
  area here */
}
```

2. Inside your mobile media query's brackets, rewrite the flex-direction of your flex container to display as a column.
3. Rewrite the width of your flex children to make them take up 100% of the width of their container.
4. Rewrite the grid-template-areas, grid-template-rows, and grid-template-columns of your

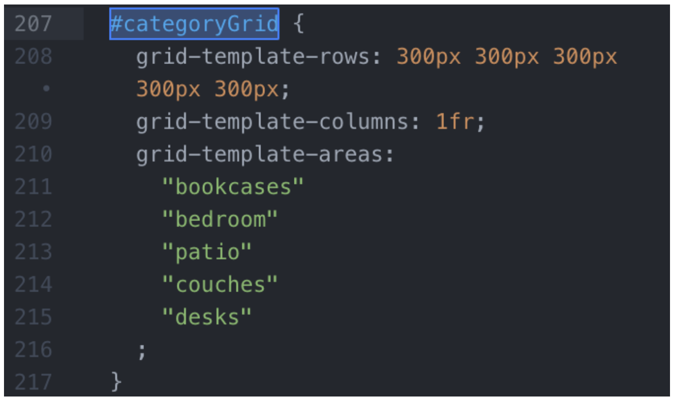
grid container to display for mobile views.
Note: See below for an example from Unit 18.2 for an example of how you might accomplish this.

Here is a sample grid container set to display on the desktop:



```
#categoryGrid {
  margin: 0px 5%;
  display: grid;
  grid-template-rows: 300px 300px 300px;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-areas:
  "bookcases bedroom patio"
  "couches couches patio"
  "desks desks desks"
  ;
}
```

And here, it is rewritten for mobile browsers. You will have to edit the grid-template-rows and grid-template columns and the grid-template-areas for your grid to display correctly.



```
#categoryGrid {
  grid-template-rows: 300px 300px 300px
  300px 300px;
  grid-template-columns: 1fr;
  grid-template-areas:
    "bookcases"
    "bedroom"
    "patio"
    "couches"
    "desks"
  ;
}
```

Step 7: Upload Your Coded Website's index.html, style.css, and Images Folder to GitHub to Update GitHub Pages (1-2 hour)

1. Upload your modified webpage to GitHub using GitHub Desktop.
2. Publish your site using GitHub Pages.
3. Open your published GitHub Pages URL to see your changes applied.
4. Make sure you copy the URL. You will need it to submit your homework via Bootcamp Spot.

Required Deliverables

Google doc with the following sections:

1. A link to a G-drive folder containing images of your portfolio wireframe and the redline wireframe mapping out the HTML elements.
    a. The original design of your portfolio website
    b. The redlined “My Ux Skills” and “My Work” sections
2. A link GitHub repository
3. A link to your Github pages website URL.

Submission

Use Google Drive (G-drive) and GitHub to turn in your homework assignment (make sure to double-check that your sharing permissions are set to “Anyone with the link can view”).

Submit the following links to https://www.bootcampspot.com:

Here’s what to submit to Bootcamp Spot:

1. **A link to your Google Doc with required deliverable links**