**TASKS**



To center content in the body of the page we'll need to create a container that is positioned and styled in CSS and then this class will be applied to several HTML elements in our index.html.To start, create a .container class in main.css, that sets the top and bottom margins to 0 and the left and right to auto.Set the max-width to 1170px.



This task applies the CSS that you wrote in the previous step so that you can see how it affects the rendered HTML. This is a pattern that you'll be following often in this project - write some CSS, then apply it to HTML.Add the container class that was just created to the following HTML elements in index.html:

* First <div> in the <nav>
* First <div> in the <!-- HERO --> <section>.
* First <div> in the <!-- MAIN CONTAINER --> block.
* The first <section> in the <footer>



In main.css create a rule that selects all <section> elements in the index.html. Adjust the display property in this rule so that each <section> acts as a flex container.



The layout of this landing page will be done using CSS flexbox. Instead of repeating ourselves across our classes we will pull the more commonly used properties into their own classes. Below you will find three classes and the properties they should contain. Create these three classes below the section rule just created.

* .flex - display: flex
* .flex-column - flex-direction: column
* .flex-one - flex: 1



Now, apply the flex utility class to several HTML elements in the index.html file:flex

* First <div> in the <nav>.
* First <div> in the <!-- BRANDING --> block.
* First <div> in the <!-- SEARCH --> block.
* First <ul> in the <!-- SEARCH --> block.
* First and second <div>'s in the <!-- PROFILE --> block.
* First <div> in the <!-- HERO --> <section>.
* First <div> in the <!-- SALE BANNER --> <section>.
* Second <div> in the <!-- FULL BANNER --> <section>.
* First <div> in the <!-- PS DEMO --> <section>.



Now, apply the flex-column and flex-one utility classes to these HTML elements in index.html :flex-column

* First nested <div> in the <!-- PROFILE --> block.
* First <div> in the <!-- HERO --> <section>.
* Second <div> in the <!-- FULL BANNER --> <section>.
* First <div> in the <!-- PS DEMO --> <section>.

flex-one

* First <div> in the <!-- SEARCH --> block.



To adjust the positioning and alignment of individual elements within a flex container, let's create three additional classes in main.css:

* .align-center - align-items: center
* .center - justify-content: center
* .space-between - justify-content: space-between



We will apply these new alignment flex classes to several HTML elements in the index.html file:align-center

* First <div> in the <!-- BRANDING --> block.
* First <div> in the <!-- SEARCH --> block.
* First <div> in the <!-- PROFILE --> block.
* First <div> in the <!-- HERO --> <section>.
* First <div> in the <!-- SALE BANNER --> <section>.
* Second <div> in the <!-- FULL BANNER --> <section>.
* First <div> in the <!-- PS DEMO --> <section>.



We will apply these new alignment flex classes to several HTML elements in the index.html file:center

* First <div> in the <!-- HERO --> <section>.
* First <div> in the <!-- SALE BANNER --> <section>.
* Second <div> in the <!-- FULL BANNER --> <section>.

space-between

* First <div> in the <nav>.
* First <ul> in the <!-- SEARCH --> block.
* First <section> in the <footer>.



Each button will have a consistent look and feel. In main.css, add the class .btn with the following styles:

* add a cursor
* remove any borders
* padding should be 6px top and bottom and 12px left and right
* Round the corners by 3px
* change the line height to 1.4

Also, Create two different button styles btn-default and btn-primary. The color of the text for both should be white. The btn-default background color is #faa541. The btn-primary background color should be #364147.



In index.html, find the only <a> element in the <!-- FULL BANNER --> block. Give this element the btn class and the button btn-default class.Then, Find the sign up <button> element in the <footer>. Give this element the btn class and the button btn-primary class.



We only have a few form controls, but let's make a custom class for them anyway. Add a new class called .form-control below the button classes in main.css.This class should change the display to block, and add padding of 6px top/bottom and 12px left/right. Next, add a 1px solid border with a stroke color of #a0a0a0. The text in the control should have a color of #8598a2. Finally, add a line height of 1.4.



Find the <input> element with the placeholder of Enter email address in the <footer> element in the index.html file. Add the form-control class to the <input>.



There are some base typographic styles we'll use throughout the page.In the main stylesheet, below the .form-control class, add three classes: .text-light, .text-secondary, and .text-primary.The light and secondary classes should have a font size of 30px. The light color is #a0a0a0 and the secondary color is #faa541.The primary class should have a font size of 16px and a color of #364147.



These are the required index.html modifications. Add the class listed to the correct element.text-light

* <div> with contents Couples retreat weekend

text-secondary

* <div> with contents Save an extra 20%



Finally, add the class text-primary to these elements in the index.html file:

* <div> with contents when you buy 2 pairs of boots
* <div> in footer with contents Customer Support
* <div> in footer with contents Company Info
* <div> in footer with contents Privacy &amp; Terms
* <div> in footer with contents Follow Us
* <div> in footer with contents &copy;Pluralsight 2018



The header of the page has several elements. In this task, we'll style the general header as well as the logo.Open main.css, and below the other styles create a class called .header-nav that adds a 1px bottom border with the color #a0a0a0



Next, setup a class called .brand below the .header-nav class that sets the element size to 170x85 pixels. Adjust the left margin by negative 15 pixels and the right margin by 60 pixels.



The logo needs to fit within the 170x85 container. To do this, create a .logo class that sets the max-width to 100% and the height to auto.



Switch over to index.html and we'll apply the newly created header and branding classes to the correct elements.Apply the header-nav class to the <nav>.Find the <!-- BRANDING --> block and apply the brand class to the first <div> in the block.Then, apply the logo class to the <img> tag with the src of img/carved-rock-logo.png.



Lets get back to work on the styles on the next header element the search box. Create a new class below the existing styles called .search-box. The search box needs to be sized to 330x30 with 6px padding top/bottom and 12px left/right. Finally, give the search box a full 1px border colored #a0a0a0.



Another element of the header is the search button. Create a class called .search-button. The background color of the button should be #a0a0a0 and the text color should be white(#ffffff).Its size should be 100x30 pixels and the border and padding should be reset to 0. Hint: It is best practice to use 0 without the px unit.



Find the <!-- SEARCH --> block in the index.html file. Apply the search-box class to the text <input> and the search-button class to the <button> element.



For the .nav-list class, zero out the left padding. Adjust the width of the list to 575px. Add a top margin of 10px.



The .nav-list links (li a) should be selected and given the following font properties, 16px, uppercase, 700 weight, with a letter spacing of 0.2px.



In index.html, locate the <ul> in the search block. Give this element a class of nav-list.



To style the profile section of the header we need two different classes. Name them .profile and .account. The only declaration in the .profile class should set the left margin to auto. The .account class should adjust the alignment on the text to the right, have a full padding of 15px, and a font color and size of #364147 and 18px.Finally, select all the links that are descendant of all elements with a class of .account. Overwrite the font-size, set it to 14px.



We'll round the profile picture and size it down.Create a class called .profile-pic in main.css.Make the size of the profile picture container 60x60.Then create a rule that selects the all descendant img elements of .profile-pic.To round the image set the border-radius to 100px. Make sure the image fits in the container set max-width to 100%.



We have three classes to apply in index.html - .profile, .account, and .profile-pic.In the <!-- PROFILE --> block assign the first <div> the class profile. Add the account class to the nested <div>. Finally, give the <div> that surrounds the image with an src of img/profile-pic.jpg the profile-pic class.



Moving further down the page, let's focus on styling the hero image and text.For the image, construct a class called .hero that adjusts several background properties.Set background-image to the img/hero.jpg image (the project has a img at the root), height to 400px, and background-size to cover.Center the image horizontally and vertically using background-position.Finally, transform the text to uppercase.



To create a more visually appealing hero section we'll add some styling to the <h1> and <h2> elements.First, create a rule that selects the <h1> in the .hero <div>. Have that rule adjust the position of that <div> by making it relative and then move it to the left by -30px.Select all <h2> elements in the .hero <div> and change the text color to #faa541.Next, select both the <h1> and <h2> elements in the .hero <div> and set a few properties to make them consistent.Adjust the font size to 63px, bump up the font weight to 700, and set the line-height to 0.9.



Now, let's change the paragraph element to match the <h1> and <h2> elements.Select all <p> elements in the .hero <div> and change these font properties: letter spacing to 2px, font weight to 700, font size to 21px.For this element's margin, zero out the top, left and right. Add 10px to the bottom. Hint: Using the 3 value shorthand for margin.



Open the index.html file and locate the first <section> after the <nav> and apply a class attribute of hero.



The sales banner is the first thing in the main content area of the page. Let's draw attention to it.First, create a class called .banner and adjust the margin of the container set it to 55px 15px.The banner is the full width of its parent so set width to 100%.To set the banner apart from the rest of the content set the top border to 1px solid #a0a0a0 and the bottom border to 1px solid #a0a0a0.Give each element some spacing by setting padding to 25px 0.Fix the width of the images to 250px by selecting all <img> elements that are descendants of the element with a class of .banner.



Transform the text in the sales banner by constructing a rule that has a selector of .banner-text. Change these text properties:

* font-size - 30px
* font-weight - 700
* text-transform - uppercase
* line-height - 1
* text-align - center



The text of the last div in the sales banner is a bit smaller than the other lines.Create a rule that selects the <div> using the > selector and the :last-child pseudo selector, and make sure it is a descendant of the .banner-text element.The rule should set the font size to 16px, font weight to 400 and add a top margin of 5px.



Add the class banner to the first <div> in the <!-- SALES BANNER --> <section>. In this <div>, find the first <div> after the image and give it a class of banner-text.



After the sales banner we'll create a four column grid using the flex property.Create a rule with a selector of .category, set flex to 0 0 25%. This will divide the space evenly in four columns.We need to add a gap between the columns so set padding to 0 15px.The images in these columns should be full width. Create a rule that selects the <img> elements that are descendants of any .category element and set width to 100%.



The text of each category should have the following properties declared in a rule with the selector .category-text:

* color - #364147
* padding - 30px 0
* text-transform - uppercase
* text-align - center



All <div>'s that are children(>) of a .category-text element should have the font properties:

* font-size - 16px
* font-weight - 200
* color - #faa541



All <h3> headings that are descendants of a .category-text element should have the font properties:

* font-size - 30px
* font-weight - 700
* line-height - 1



Open index.html and locate the <!-- CATEGORIES --> block.There are four top-level <div> elements in a <section>. Give each of these a category class.Next, find the <div>'s after each image in the category <div>'s and apply a class of category-text.



After the categories, we'll create a Full Banner for Trail Reviews. It will have a picture covering 60% of the main container and a sidebar covering the rest.To start, create a .full-banner class that sets some box model properties. Set padding to 0 15px,and margin to 50px 0. Set the position to relative so that we can create an inset border next.



For the border, we'll use the .full-banner class selector and the pseudo-class :before. Because we're using :before, set content to "" so there is something to style.Because we used relative positioning for the .full-banner, we can use absolute positioning in this rule. Set position to absolute. Next, set the border to 1px solid #ffffff. With this set, we can adjust the top, bottom, left and right. Use, 30px, 30px, 50px, 50px respectively. Bring the border to the front using z-index: 1.



Next, let's use the flex property to have the image take up 60% of the width of the banner. Set flex to 0 0 58.333333% in a rule with a selector of .full-banner-image.Set, background image, size, and position to img/vista.jpg, cover, and top center, respectively. Set the minimum height of the image to 400px.



The text of the banner is on the right and takes up 40% of the width of the banner, so use flex: 0 0 41.666667% in a rule with a selector of .full-banner-sidebar to adjust this.Set the background-color to #364147, transform the text to uppercase, add padding of 30px and change the text color to white(#ffffff).



Style the <h4> heading in the full banner by creating a rule with a selector of .full-banner-sidebar h4 and adding these font declarations - size(40px), weight(700), style(italic), color(#637f94), and overwrite the line height(1).



Let's create a decorative element after the heading. This element will be after the .full-banner-sidebar h4 element. Add the :after pseudo class to this selector to create a rule with the following declarations:

* content - ""
* width & height - 100x1 pixels
* background-color - #faa541
* margin - 20px auto
* display - block



For the text below the heading we're going to adjust a few properties. Select the first <div> in the side bar using the group of selectors .full-banner-sidebar > div:nth-of-type(1), and set the following:

* font-size - 60px
* font-weight - 700
* line-height - 1

Select the second <div> in the side bar using the group of selectors .full-banner-sidebar > div:nth-of-type(2):

* font-size - 25px
* font-weight - 200
* letter-spacing - 3px



Finally, let's create a rule to style the button in the sidebar.Select the button using .full-banner-sidebar .btn.btn-default - a top margin of 50px, make the text uppercase and change the letter spacing to 1px.Make sure the button is first in the stacking order by changing the z-index to 1.



Locate the <!-- FULL BANNER --> block in the index.html file. Give the <section> in this block the class full-banner. Give the first <div> the class of full-banner-image and the second <div> a class of full-banner-sidebar.



The footer completes the landing page. It is divided into two sections - the footer links and a demonstration message.To style, the footer select the <footer> element and make it a flex container with flex-wrap enabled(wrap). Set position to relative, z-index to 0, and minimum height to 330px. Make the background color black.



Dynamically set the content after the footer to "". Hint: Use the :after pseudo-class. This will create a sidebar.In the same rule(footer:after), Set the background color of the sidebar to #a0a0a0 and set the width to 75%.Set the position of the sidebar to absolute. Now that the element is absolute, set top, bottom, left and right to 0. Finally set z-index to -1.



All of the content contained in the footer is in unordered lists. To equally space these lists, create a .lists rule and set flex to 0 0 75%. Add a gap between these columns that adds up to 30px(left/right - 15px). The top and bottom padding should be set to 40px.



Each list has a header, so create a rule with a selector of .link-header and transform the text to uppercase. Set the font size to 16px and font weight to 700. Change the bottom margin to 15px.



The anchor tags in the footer should receive the styling: white text color, uppercase, font size of 12px, and the padding should be set to 0 0 4px. Whenever the link is hovered over set text-decoration to underline.



The first <section> in the <footer> should be given a class of lists. Add the link-header class to the following elements:

* <div> in footer with contents Customer Support
* <div> in footer with contents Company Info
* <div> in footer with contents Privacy &amp; Terms
* <div> in footer with contents Follow Us



To help style the social links in the footer, create a rule that has a selector of .social ul li and sets the display to inline-block.The links in the social list (.social ul li a), should have the following styles:

* display - block
* width and height - 23px
* background size and position - 23px 23px center



There are four anchor tags in the <ul> in the Follow Us list. These links will have icons for social media platforms.Create four classes .twitter, .facebook, .instagram, and .pinterest.Each rule should have a single declaration of background-image. Set this to the correct URL for each image:

* img/twitter.svg
* img/twitter.svg
* img/instagram.svg
* img/pinterest.svg



Give the last <div> in the <footer> <section> the class social. Then, find the <ul> under the Follow Us <div>. To the first <a> apply the twitter class, second apply facebook, third apply instagram, and fourth apply pinterest



Below the social icons there is a newsletter section. This block has two <div>'s that need styling. Create two rules, one that has the selector .newsletter .link-header and has the properties:

* font-size - 12px
* font-weight - 400
* margin-bottom - 5px

The second should have a selector of .newsletter .link-subheader and declarations that adjust the following: font style(italic), size(12px), weight(400), and a bottom margin(15px).



In this newsletter section there is also an <input> and a <button>. To select just this <input>create use the selector .newsletter input. The <input> should have a border of 1px solid #364147, width of 200px and bottom margin of 10px.Select just this <button> with .newsletter button and style it with these properties:

* width - 150px
* min-width - auto
* text-transform - uppercase
* letter-spacing - 1px



In index.html find the <div> after the social icons <ul> and apply a class of newsletter. Give the nested <div> with the content Email Updates a class of link-header. In the <div> immediate below, apply a class of link-subheader.



The final section of the page is a notification that this site was made for demonstration purpose only. To style it, create a rule that has a selector of .ps, and add the following properties:

* flex - 0 0 25%
* padding - 40px 0
* color - #e5e5e5
* font-size - 16px

The Pluralsight logo in this section should have a max-width of 230px.



Create a rule with the selector .demo and add a padding of 0 30px. Before this demo section (.demo:before) add these properties:

* content - ""
* display - block
* width - 100%
* height - 1px
* margin - 30px 0
* background - linear-gradient(to right,#f05a28 0,#e80a89 100%)



The final piece of the footer is the copyright. Create this one last rule, the selector .copyright, with the properties - flex(0 0 75%), padding(10px 0), font size(12px), and center align the text.



Locate the <!-- PS DEMO --> block in the index.html file. Give the <section> in this block the class ps. Give the nested <div> the class of demo. Give the copyright <div> the class of copyright.