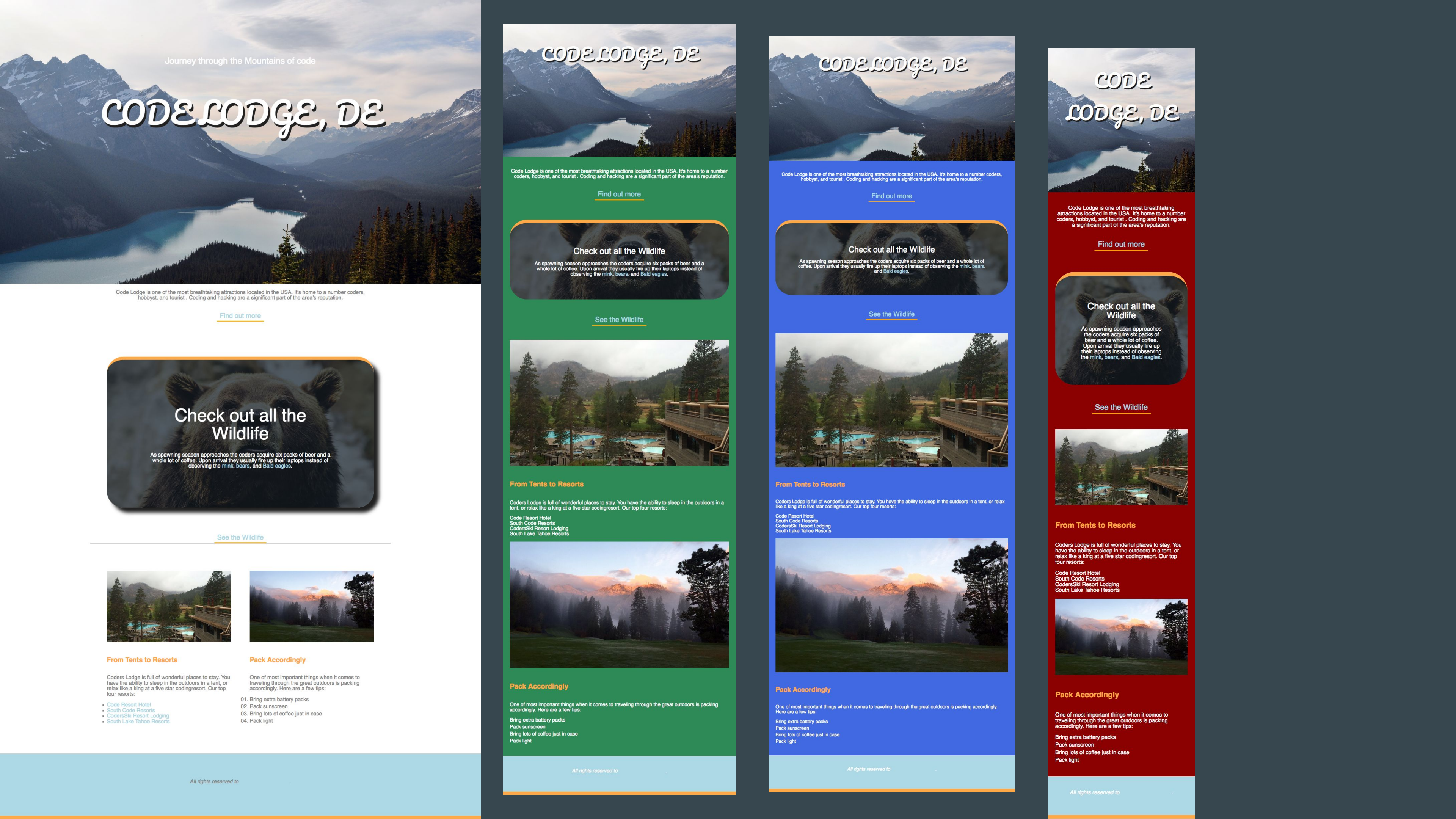
Intro to HTML, CSS & JavaScript

By Mikaila Akeredolu
Email: mikaila@zipcodewilmington.com
LinkedIn: https://www.linkedin.com/in/mikailaakeredolu

Course

- Making a web page with HTML, CSS and JavaScript
- Writing HTML tags and CSS rules
- Laying out a web page with multiple sections
- Working with Images on the web
- Links / URL
- Navigation
- Customizing fonts
- CSS3 animations
- Intro to JavaScript


Journey through the Mountains of code
CODE LODGE, DE
Code Lodge is one of the most breathtaking attractions located in the USA. It's home to a number coders, hobbyst, and tourist . Coding and hacking are a significant part of the area's reputation.
Find out more
Check out all the Wildlife
As spawning season approaches the coders acquire six packs of beer and a whole lot of coffee. Upon arrival they usually fire up their laptops instead of observing the mink, bears, and Bald eagles.
See the Wildlife
From Tents to Resorts
Coders Lodge is full of wonderful places to stay. You have the ability to sleep in the outdoors in a tent, or relax like a king at a five star codingresort. Our top four resorts:
Code Resort Hotel
South Code Resorts
CodersSki Resort Lodging
South Lake Tahoe Resorts
Pack Accordingly
One of most important things when it comes to traveling through the great outdoors is packing accordingly. Here are a few tips:
01. Bring extra battery packs
02. Pack sunscreen
03. Bring lots of coffee just in case
04. Pack light
All rights reserved to

What is HTML?

HTML is the language of the web.

HTML stands for **H**yper **T**ext **M**arkup **L**anguage

Simple definition:
It's a way to markup a document to specify attributes like different font sizes, list and links on web pages...

HTML is written in text files and end with the extension .html eg: index.html

Web browsers are typically used to display HTML such as safari, Chrome & I.E

Recipe Web Page

Before we make a web page we have to figure out what the content will be. Then we mark up the content with elements, tags and attributes.. Once we create a markup then we style it with CSS and finally use some JavaScript to add to actions or functionality.

DOCUMENT STRUCTURE

- DOCTYPE
- HTML
- TITLE
- BODY

```
<!DOCTYPE html>
<html>
  <head>
    <title>Lake Tahoe</title>
  </head>
  <body>
    <header>
      <span>Journey through the Sierra Nevada
Mountains</span>
      <h1>Lake Tahoe, California</h1>
    </header>
    <p>
      Lake Tahoe is one of the most breathtaking attractions
located in California. It's home to a number of ski resorts,
summer outdoor recreation, and tourist attractions. Snow and
skiing are a significant part of the area's reputation.
    </p>
    <a href="#">Find out more</a>
  </body>
</html>
```

HTML SYNTAX

When writing HTML content we use something called tags..
There are many types of tags /elements such as h1 - h6, p, span etc..

<h1> Zip Code Wilmington</h1>

HTML Tags / Elements and Attributes

<h1> to <h6>... For headings

 ...for paragraphs

href=" " ...attributes to define tags

 ..... used to add more description to a tag

 ....used for grouping and styling inline elements

 ....used to target more than one element

<a> ....Anchor tags to create links

 .....Used to group certain elements in your markup like a container

<link> ...used to links things usually a stylesheet

  ....Used to create unordered and ordered list

See w3schools and let's practice these tags and elements

What is CSS?

CSS Stands for Cascading Style Sheet.

We use CSS to make our HTML markup presentable. {add some swag}

By adding colors to fonts/text, background colors

Adding spaces between elements, selecting multiple elements

Laying out and arranging images and videos on our websites and more....

CSS SYNTAX

The Syntax / Rule
We select elements with tags, ID's and classes...

h1 {color : yellow;}
Selector{declaration block}

PropertyName: Value

Let's start zip coding...

Download the starter project via - my github and download the starter files

https://github.com/MikailaAkeredolu/teachingfrontend

Next let's download sublime text. Open up the index.html file with sublime.

See that we have a doctype declared and we already have a structure with several tags
Let's practise some HTML by going to w3schools...

Finally let's create our own HTML file, write some markup and view it with google chrome. We will be using these tags as we create our webpage and style with css

Inline Styles

There are three ways to add CSS to your HTML
Inline styles , Internal and External styles

Let's add a background color to our body and h1 element.
Not a good practice so let's use an internal then external stylesheet.

Inline styles.....

<body style=" background-color: orange"; >

<h1 style=" color:white";>

Internal Styles

To add an internal style you use the style tag as seen below

```
<style>

p{ font-size : 20px; font-weight: bold;}  h1{font-size:90px;}

</style>
```

External Styles

This is the recommended way to add css. We will need to create a stylesheet and call our style page something like style.css.

Next we link the external stylesheet to our html file with a link element within the <head> tag. We will include what we call attributes...they help create relationships and define our tags.. So we will add a rel attribute and a type attribute and an href attribute.

The href points to the location of our css file...which is in the css folder

```
<link rel="stylesheet" type="text/css" href="css/style.css">
```

Using our external CSS style page

Lets style the existing html file with styles we will write in our external css file.
Add the following styles below to your style.css. Save it and refresh your browser.

```
body{text-align: center;}
h1{font-size: 72px;}
header{background-color: orange;}
p{font-size: 20px;}
```

Using Div Tags ... open a and add content in a <p> tag

Remember that we use div tags to separate our content. Think of them as containers but they have no semantic meaning. Which means that search engines ignore them for SEO. Let's add a starting tag to our existing content

```
<div>
 <p> Lake Tahoe is one of the most breathtaking attractions located in California. It's home to a number of ski resorts, summer outdoor recreation, and tourist attractions. Snow and skiing are a significant part of the area's reputation.</p>
```

Next...Add more HTML ...tags <h2>   + closing....

<h2>Check out all the Wildlife</h2>
As spawning season approaches the fish acquire a humpback and protuberant jaw. After spawning they die and their carcasses provide a feast for gatherings of mink, bears, and Bald eagles.
See the Wildlife

  <---- Here we close our first tag!

Add more HTML content...including unordered links

```
<div>
<h3>From Tents to Resorts</h3>
 <p>Lake Tahoe is full of wonderful places to stay. You have the ability to sleep in the outdoors in a tent, or relax like a king at a five star resort. Here are our top three </p>
<ul>
 <li><a href="#">Lake Tahoe Resort Hotel</a></li>
<li><a href="#">South Lake Tahoe Resorts</a></li>
<li><a href="#">Tahoe Ski Resort Lodging</a></li>
</ul>
```

Add more HTML including Ordered links...

```
<h3>Pack Accordingly</h3>
<p>One of most important things when it comes to traveling through the great outdoors is packing accordingly. Here are a few tips:</p>
<ol>
        <li>Bring layers of clothing</li>
        <li>Pack sunscreen</li>
        <li>Carry extra water just in case</li>
        <li>Pack light</li>
 </ol>

</div>  <--------Don't forget to close your div tag
```

Add a footer Section <footer>.....</footer>

The footer is the bottom section of your site where you add things like copyright etc.

```
<footer>
<p>All rights reserved to the state of <a href="#">Delaware</a>.</p>
<a href="#">Back to top &raquo;</a>
</footer>
```

CSS Selectors in detail

- * Universal selector allows you add universal styles to your web page

Let’s comment out the css rule in the body element then add a margin, padding and color: red. Save and let’s take a look

This removed all margins and paddings and changed all text color to red...

Type selectors p{ }

Let's comment out the body, h1, header and p tag css rules...
Next let's add a new css rule for the p tag

P{ color: white; background-color: blue; }

If we want to target the header we use the header element and change it to orange.

header{background-color: orange;}

Next let's remove the P tag rule /*P{ color: white; background-color: blue; }*/
Let's then select the h1 tag and set it its properties to font-size: 90px; color: white;

Next Target the h1, h2 and h3 tags using hexadecimal colors

Now let's declare the following...

h1 - font-size: 90px; and color: white;

h2 - font-size: 53px;

h3 - font-size: 20px; color: #48525c; (See hexadecimal colors - RGB)

Body - color: #878787; margin: 0; (no more white space with margin 0..)

```
footer {background-color: lightblue;}
p { color: slategrey;font-size: 18px;}
```

HTML ID's and CSS ID Selectors, Anchor and a Border

Let's target elements in a unique way based on its id attribute.
Let's declare an ID selector that will match an tag we add the id to...

HTML -

CSS - #primary content{border: 3px solid red;} Border adds a box to your content

Anchor - Add the #top id attribute - <header id="#top">

Then add an href #top attribute to the footer Back to top »

More on ID's

Add an ID to the footer called main-footer - <footer id="main-footer">
Let's target the main footer ID with CSS and add the properties below

padding-top: 60px;

padding-bottom: 60px;

border-bottom: solid 10px orange;

Class Selectors

Class selectors are like Id selectors except that you can target multiple elements that you give the class attribute - Let's change our the primary-content id to a class selector
.primary-content{border: 3px solid red;} (Now we use a dot instead of the hashtag)
Add the same class of primary content to the div for From tents to resorts
Then change the div's class to secondary-content
Then for the header give it a main-header class <header id="#top" class="main-header">
Change the header selector to match the .main-header class - . main-header
Next remove the red border style from .primary-content class and let's center align all the text in that div with text-align: center
Next target .secondary-class and add border-top declaration - border-top: 2px solid gray;

Using CSS Rules more than once... t-border

We are going to create a CSS rule that can be used more than once

Comment out the .border-top in the secondary-content and add a new CSS rule

.t-border{ border-top:2px solid gray;}

Then we need to add the t-border rule to the primary and secondary contents

Also let's add it to our footer <footer id="main-footer" class="t-border">

Descendant Selectors

Let’s target the span inside the main-header.

.main-header span{ color: white; font-size: 26px;}

Let’s target another descendant selector the li element in the ol
ol li{background-color: tomato; color: white; margin-bottom: 5px;}

Finally give the .main-header span its own class for more flexibility

.main-header span{ color: white; font-size: 26px;} = .title{ color: white; font-size: 26px;}

Pseudo Classes - link, visited, hover, active, focus

They are used to target elements based on user interactions.
For starters there is a link history pseudo class to help specify the color of a link when clicked or hovered. Let's add a pseudo code to target unvisited links. This will only affect links with href attributes. Eg: <a href="#more">
Let's add some styling for unvisited links with - a:link{color:lightblue;}
Next let's target all visited links and make them lightblue a:visited{color :orange;}
Next add a hover pseudo class rule a:hover{color : forestgreen;}
Next let's add an active pseudo class a:active{color : yellow;}
Last let's add a focus pseudo class a:focus{color : white; background-color: orange}

CSS Value Units - px, ems and rems

Absolute Units : There are 6 different types but we will use Px (pixel) to demo
Let's add a width: 960px to the .main-header. The size is fixed to what you set
Percentage and em Units are relative to a parent element's length.
Add width of 60% to .primary and secondary content with one rule.
.primary-content, .secondary-content{ width: 60%;} then remove the width: 960px from the .main-header.
Relative units use the em unit. Default font sizes are 16px which is equal to 1 em
Let's add 1 em to the body for font-size then change the title's font-size: 26px; to em's
To calculate we do 26px / 16px = 1.625em result = .title{font-size: 1.625em;}
Now let's change the h1 from px to em 90px / 16px = 5.625em

Rem - Root element - refers to the HTML element

Not affected by the parent element. It's relative to the default element. Let's change the h1 font-size: 5.625rem; to rem and the title font-size to rem

Colors Hexadecimal, RGB

Hexadecimal notation is a combo of Red Green and Blue values
They are represented by 0-9 and A-F. Eg: #FF0033. FF is Red, 00 is Green and 33 is blue
Also if all three values are the same then you can do #F03 as shorthand syntax.
We can try this out by changing the main header to #FF0033 then #F03
Let's change our orange colors to an hex value #ffa949. Use it for the following:
.main-header - #ffa949; .main-footer - #ffa949;
ol li - color: #878787; and comment out the background-color and the a:visited{}
The RGB method uses a functional notation. Let's change our h3 color
H3 {color: rgb(255,169,73);} to red green and blue Note: (255 = white and 0 = black)
Change the a:hover to accept an opacity a:hover{color: rgba(255,169,73, .4);}

Text Styles

Text-Align: Lets us control the horizontal alignment of text eg: center
Text-transform: Changes the case of text to Uppercase. Lowercase or Capitalized
Text-Decoration: Sets the line decoration of elements like removing underline links
Font-Weight: Sets how thick the characters are displayed.
Lets align the text in the header by adding the .main-header class next to:
.primary-content, .main-header{text-align: center;}
Add text align center to #main-footer and make the background color lightblue
Next let's transform our h1 to uppercase h1{text-transform: uppercase;}
Remove underline links with text-decoration a:link{text-decoration: none;} also add an underline to the h2{text-decoration: underline;} use font-weight: normal; on h1 and h2

Font Properties and Google Fonts

Font-family defines a typeface for our font

Font-style allows italic or oblique faces to be selected within a font family.

Let's add to the body element a few font stacks incase the user does not have Helvetica

body{font-family: Helvetica, Arial, sans-serif ;}

In the footer let's use italic footer p {font-style:italic;}

Let's go to google fonts and use a font for our h1.

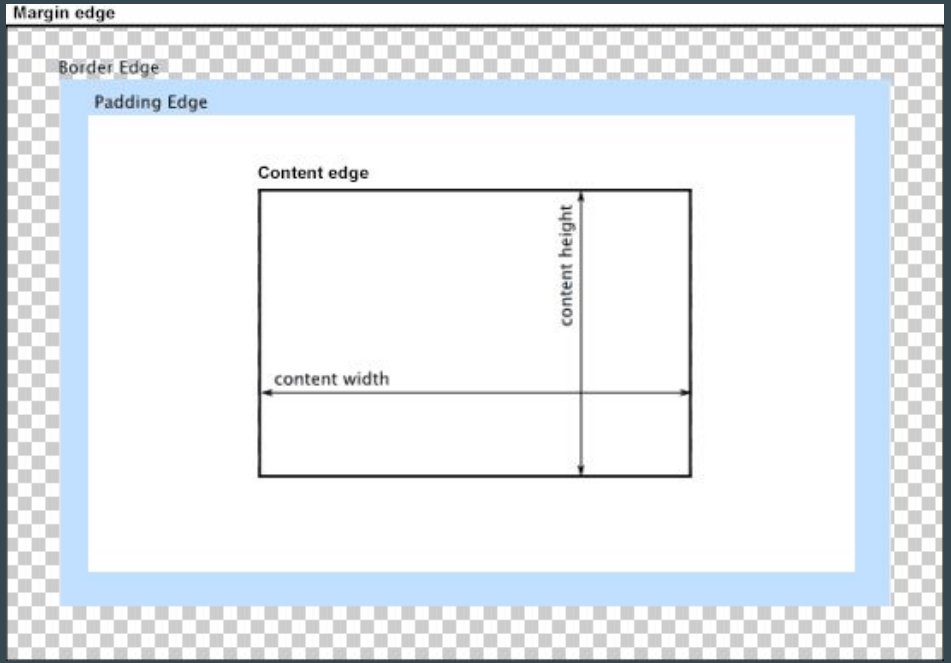
Pick a font and add it to your html <head> then add the css to your css file

Eg: <link href="https://fonts.googleapis.com/css?family=Pacifico" rel="stylesheet">

font-family: 'Pacifico', cursive;

CSS BOX Model

Think of each element as a box.
We have box level elements
and inline elements.
Add a *{border: solid 2px;} to see
* Content -- inside
*Padding: surrounds the content
* Border - outline to the box
*margin - space outside the box



Experiment with Padding and Borders

Lets add a before the h2 with a class of wildlife then close it before the see the wildlife link. Now style it ...
.wildlife{color: white; background-color: #434a52; padding-top: 100px; padding-right: 120px, padding-bottom: 100px; padding-left: 120px} For padding think (TRBL)
Add a border with of 10px and border-style of solid; = border: 10px solid #ffa949;
Let's change it to border-top: 10px solid #ffa949;

Experiment with Margins and more styling

Add a margin to our wildlife class margin: 105px 0 60px 0; (T,R,B,L)

Let's convert some pixels to ems as a review..

The h2 is font-size of 53px / 16 = 3.3125em

And the h3 font-size of 20px / 16 = 1.25em

Then add a margin-bottom to our h2 of .5em and h3 of 1.7em

Also let's remove the underline from the h2 by changing the h2's text decoration: none

Let's center the primary and secondary content with margin auto -

The browser centers it for us with margin auto.

.primary-content, .secondary-content{ width: 80%; margin: auto; }

Display Values - Block and Inline

<a> tags are displayed inline by default while , <h1> etc.. are block level elements
Let's add a class called "callout" to find out more & See the wildlife <a>links.

```
<a class="callout" href="#">Find out more</a>
<a class="callout" href="#">See the Wildlife</a>
```

Let's style the .callout class with the following properties

```
font-size: 1.25em;
border-bottom: 3px solid orange;
padding: 0 9px 3px 9px;
margin-top: 20px;
display: inline-block;
```

(Your margins will not work without inline-block)

Adding Images with alternate text and Box Sizing

Let's add two images to our site with the <img> tag. Add the first image under the div of secondary content <img src="img/resort.jpg" alt="Resort"> then wrap the next h3 content (Pack Accordingly)in a and add the image as the first element inside the <img src="img/mtn-landscape.jpg" alt="Mountain Landscape">.....
For our .primary,.secondary{} styles
Add a width: 75%; padding-left: 50px, padding-right: 50px;
Let's add a universal box-sizing property. Box sizing alters the default css box model to so that we can calculate width and height of elements without the browser adding to it *{box-sizing: border-box} then add a height: 850px and padding-top: 170px to our .main-header. Finally add a max-width: 900px to our primary,.secondary{} (keep fixed)

Style the <img> Element and Background Images

Let's make our images responsive with a max-width property Img{max-width: 100%}
Let's create some separation between the image and content- margin-bottom: 20px;
Now let's add a background image to our main header with some properties...
.main-header{background-image: url('../img/mountains.jpg');} (../ go one dir backwards)
Background-size: cover; -- shows the full size of the image
background-repeat : no-repeat; -- makes the image Not repeat
Background-position: center; - centers the image
Let's practice adding one more image to the .wildlife div. Add the bear image and style it with the following - background-image: url(../img/bear.jpg); background-size: cover; background-position: center; background-repeat: no-repeat;

Floats

We will float the two divs side by side in a row with floats. Let's give each div a class
Under the add then
Add a class tips to the second div before the <img src="img/mtn-landscape.jpg" alt="Mountain Landscape">
Add a width for both divs .resorts, .tips{width: 46.5%}
Add .tips{ float: right; } then .resorts{ float: left; }
We need to clear our floats to prevent collapse with a clearfix so add the group class to the secondary-content div (parent) -
Add these -.secondary-content{ padding-top: 80px; padding-bottom: 70}
Add a Float Clearfix - .group:after{ content: " "; display: table; clear: both }

List Styles, Text Shadow, Box Shadow & Border Radius

Create new rules for our url with the list-style-type property.

ul { list-style-type: square; padding-left: 0; margin-left: 0 }

ol { list-style-type: decimal-leading-zero; padding-left: 0; margin-left: 0 }

Next we will add some text shadow properties to our <h1> tag

H1{ text-shadow: 5px 8px 0 #222; } /* horizontal, vertical, blur, color */ (blur 10px)

We can add a box shadow to our .wildlife div container.

.wildlife{ box-shadow: 15px 15px 10px #222; } /* horizontal, vertical, blur, color */

Border Radius is used to apply rounded corners to divs and images...

Let's add border radius to our .wildlife and see.. border-radius: 10px;

Media Queries

Media queries are used to make your website responsive.@media(){ } targets all screens
Let's add that to our stylesheet and add an expression that checks for the conditions.
We will start with max-width to check for the max width of browsers viewport
@media(max-width: 960px){ } - when browser is at or below 960px. We will change the background-color of the body to royalblue and the color of the text to white.

```
@media(max-width: 960px){
body{background-color: royalblue;}
p{color: white;}
}
```

Next try to add a max-width of 480px and set background color to darkred

Combining Media Queries and adjusting layout

Next we will look at how to combine media queries. Create a new media query/seagreen
@media(min-width: 481px) and (max-width: 700px){ body{background-color: yellow;}}

As you can see our layouts are breaking so let's adjust them with media queries.

@media(max-width: 1024px). Let’s target the .primary and .secondary-content classes

```
@media(max-width: 1024px)
{
.primary-content, .secondary-content{ width: 90%; }
}
```

More media queries please....having fun yet?

Create a media query to adjust our layout at max-width of 768px.

```
@media(max-width: 768px){ .primary-content, .secondary-content{width: 100%; padding: 20px; border: none;} ....let's add the next ones here ....}

.main-header{ max-height: 380px; padding-top: 10px; padding-left: 25px; padding-right: 25px; padding-bottom: 0px; }
.title{display: none;}
h1{font-size: 3rem;text-shadow: 3px 4px 0 #222;}
h2{font-size: 1.5em;text-decoration: none;font-weight: normal;margin-bottom: .5em;}
p{font-size: .9em;}
```

Media queries continued

Add the following to the same 768px breakpoint that we were working on....

```
ul{font-size: .9em; list-style-type: none; padding-left: 0; margin-left: 0;}
ol{font-size: .9em; list-style-type: none; padding-left: 0;margin-left: 0;}
.secondary-content a{color: white;}
.secondary-content ol li { color: white;}
```

Let's modify the .wildlife div to make everything look good on smaller devices too

```
.wildlife{padding: 50px 60px 50px 60px; box-shadow: none; margin: 55px 0 30px 0; }
```

Go back to the @media(min-width: 481px) and (max-width: 700px) style and add..

```
wildlife{padding: 100px 80px 100px 80px; box-shadow: none; margin: 55px 0 30px 0; }
```

Now refresh and take a look and notice the margin and paddings make a difference.

Let’s adjust our column layout for our images

In the @media(max-width: 768px) breakpoint.. Let’s readjust the images .resorts, .tips{float: none; width: 100%;} Now watch the images become more visible Finally let’s adjust the padding for the #main-footer{padding: 20px 0;}

If we view our webpage now on a mobile device none of these changes will appear. We need to add a meta tag with viewport to make all devices can view our changes <meta name=”viewport” content=”width=device-width”>

Next Animations....

CSS3 Animations with Transitions and Transforms

Transitions allow you to change a css property over a period of time or duration.
In order to transition a property, you need a start value and an end value.
The transition-duration property is required property .It sets the length of time of the transition. Eg: transition-duration: .4s; The 4 transition properties available are:
Transition-property: The property or set of properties to transition.
Transition-duration: How long the transition should last
Transition-timing-function: speed Transition-delay: when the transition starts.

```
H1{ transition-property: color, background-color, border; transition-duration: 2s;
transition-delay: .4s; transition-timing-function: ease-out;}
H1:hover{color: orange; border: dashed 3px orange;}
```

CSS Transforms

The Transforms property is has several functions such as rotate.
Let's rotate our images and see. Add a new rule of img{transforms: 25deg}
Transforms can be animated on hover so let's make our images spin 360 degrees
Add the transition property to your <img> then use the transform on hover
Add img {transition-property: transform} then add img:hover{transform: rotate(1turn)}
Next let's add a border transition to it as well upon hover. Make it 5px solid black
Finally let's combine all we have learnt by transitioning and transforming the see the wildlife's image on the next slide...

Combining CSS transition and transform

Add the following to your .wildlife div

```
transition-property: background-image,transform;
transition-duration: 2s;
transition-delay: .5s;
transition-timing-function: ease-in;
```

Then on hover add the following. We will scale the image and transform it

```
.wildlife:hover{
background-image: url(../img/mountains.jpg);
transform: scale(1.1);
}
```

CSS Translate - used to move elements

You can move elements with the translate function of css transform. You can move elements on the x or y axis. Let's move our .wildlife div down to see. Add the following

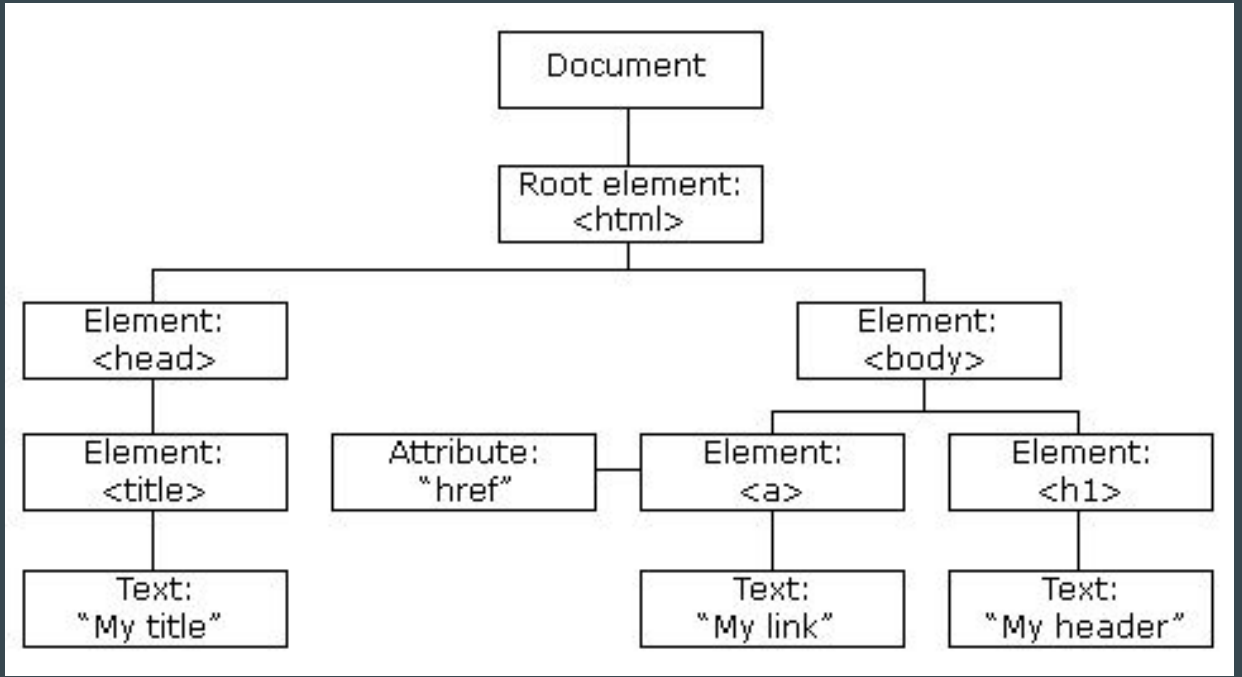
```
.wildlife:hover{
background-image: url(../img/mountains.jpg);
/*transform: scale(1.1);*/
transform: translateY(100px);
}
```

JavaScript

Javascript is a programming language that enables you to create dynamic content, control multimedia and more. It can be used on the client (frontend code that runs on the user's computer) and server side (backend) of your application to send user's info to the frontend or store user info in a database.
An example is using node.js on the server and mongoDB as your database.
We will briefly go over JavaScript so that we can use it in our webpage to affect the frontend. We will go over the DOM (document object model) as seen here
https://www.w3schools.com/js/js_htmldom.asp
We use the DOM to change, add or delete HTML elements

Document Object Model - DOM

DOM = Family Tree
Document = window/box
<html> = root inside box
<elements> = parents
Elements have children
They also have siblings.



Script Tag

To start writing JavaScript the first thing we need to do is add an external js file before the closing body of our html eg: - <script type="text/javascript" src="script.js"></script> *** Right after the closing footer tag </footer>*** then we need a js file. Now that we have our js file linked to our html. Let's add the following to our js file... console.log('It works'). In your browser click view > developer > javascript console We should see a print out in the console saying it works!

JavaScript Variables and data types

Variable are like labelled boxes, they hold things (data types)
Data types in Javascript consist of strings, numbers, arrays, booleans, objects etc..
Strings are just text inside quotes like this “zipcodewilmington” or “1978”
Let’s say we want a variable that stores a name. We can declare it like this var name;
then we can store a name inside like this var name = “zipcodewilmington”
We can also store a number data type in a variable like var number = 19.
Arrays are a type of list data type to hold things like a grocery list or list of favorites
Eg: var groceryList = [“apples”, “milk”, “bread”, “oranges”];
Objects are variables too but they can store different data types with keys and: “values”
For example: var car = {type: “Honda”, “model”:”CRV”, “year”: 2017”, “color”:”black”};

functions

Functions are like cooking recipes. For example if you had a bakeCookiesFunction. It goes through all the steps in the function to bake the cookie but if you wanted to bake a specific type of cookie then you pass the function a parameter like (chocolate chip). Once the function runs and sees chocolate chips then it knows to use the chocolate chip to bake the specific type of cookie you want which is chocolate chip.

```
function bakeCookie(chocolateChip){
Mix flour + add sugar + add chocolate chip + put in oven;
return chocolate chip cookies;
}
```

To run the function you simply call it (invoke it) - bakeCookie(chocolateChip);

Variables, Objects and Functions

The DOM itself is an object. Objects have things that define who/what they are and things they can do. Properties gives the object an identity like a (Person - name, age)
Methods are the actions that can be performed on objects such as action : function(){}

```
var celebrity = {
name : ”jayZ”,
isFromBrooklyn : true,
age : 47,
genre : [ “Hip Hop”, ”rap”, ”poetry”],
action :  function(){return “this.name” + “ is  ” + “this.age” + “ years old” };
}
console.log(celebrity.name);       or     console.log(celebrity.action());
```

Working with the DOM

Now that we have a basic understanding of objects , properties and methods . We will leverage the DOM Object and its methods to manipulate our website. In your script.js file let's create a variable and use the dom to manipulate our main-footer area.

```
var footer = document . getElementById ('main-footer') ;
footer. addEventListener ('mouseover', function() {
footer. style.backgroundColor = "black";
});
```

This changes the background color of the footer area once we mouse (hover) over it (Event). By invoking(calling the anonymous (no name) function)

Callback - A function that's passed to another function as an argument and used later.

More DOM methods - GetElementsByClassName

Another way to manipulate the DOM is by using the DOM class method - getElementsByClassName. You can select all classes at ones and target specific ones by their index. Eg: Let's create a duplicate of the with class="title" like this Second Journey through the Mountains of code . Next type

```
var theTitle = document.getElementsByClassName('title');
theTitle[1].addEventListener('click', function(){
theTitle[1].style.color = "purple";
});
```

Note that we are able to target a specific title by using its class name and index number

Triggering events from other elements

Finally, we will take a look at combining all we learned by triggering an event one one element from another element. Eg: We click a button and some other element is affected. Let's create a button element below our with class of title
<button id="button">The button</button> then add a class to our <ol> tag like this
 <ol class="orderedList">
In our script.js file let's target all the ordered list and make them disappear. (magic)

```
var button = document.getElementById('button');
var orderedList = document.getElementsByClassName('orderedList');
button.addEventListener('click', function(){
orderedList[0].style.display = "none";  });
```

Thanks for your time and for coming out!

To learn more about software development opportunities visit zipcodewilmington.com

Final code:

https://github.com/MikailaAkeredolu/finalcodelecture