Red Academy – Web Development

Day 2

10/04/2018

Common Form Elements

* All forms on the web use the <form> element

E.g.

<form action=”submit.php” method=”get”>

<!—The form input goes here🡪

/form>

* The action attribute defines the location (url) where the forms collected data should be sent
* The method attribute defines whih http method to use to send the data (can be get or post. Get means receive data. Post means send data).

Input Element

* We can use the <input> element to define which kind of data we’re expecting (e.g. date, range, numeric, email, etc.)

<input type =”text” name=”username” size =”12” maxlength=”12 />

* Size controls the size of the box itself. Maxlength limits the number of characters that someone can submit

Type = Radio – have the same name across all the radio buttons (it’s a group name). This makes it mutually exclusive. If they have different names, then a user can select more than 1 button.

* The value of the radio button determines what value will get sent through to the server. The id is used largely to tie it to javascript conditions. The name you give it after the end of the element will be what is shown to the user.

Type = password – don’t define the value because we want the user to define it by what they input. Also this type will make the input characters appear as dots

* <input type =”password” name=”password” size “8” minlength=”6” max length=”8” />
* Best to use https (SSL/TLS certificates) to send requests securely

Type = textarea – good for things like blog posts.

<form <textarea name =”,essage” cp;s =”100” rows=”4”> enter your message… </textarea> /form>

Type = Select element is a special form element. This allows us to have specific options for someone to select (e.g. Select your computer. Then option values = “Mac” or “PC”

Select Name =”os>

<option value=”Mac”>Mac</option>

<option value=”PC>

Select=”Selected>PC Mac</option>

The selected bit defines the default value

Label element – best to use on ever input types. This keeps things ordered and helps machinereaders (for blind ppl).

<label>

First name: <input type=”text” name=”first \_name” />

</label>

* You can also use the for attribute to match the id of the input it’s labelling
* <label for=”terms> I agree to the terms</label
* <input type=”text name=”terms id= terms />

Button Element

* Button type=”submit”>Submit</button
* This puts a button there that will distinguish that once it’s clicked all the values in the input box will get wrapped up and sent to our server
* On it’s own, this will usually send someone to another page. We usually don’t want this, so we typically work it with some javascript so that it keeps ppl on the same page but still sends us the data
* Button and input are similar but input is selft closing whilte the button can contain content

INTRO to CSS

What is CSS – Cascading Style Sheets

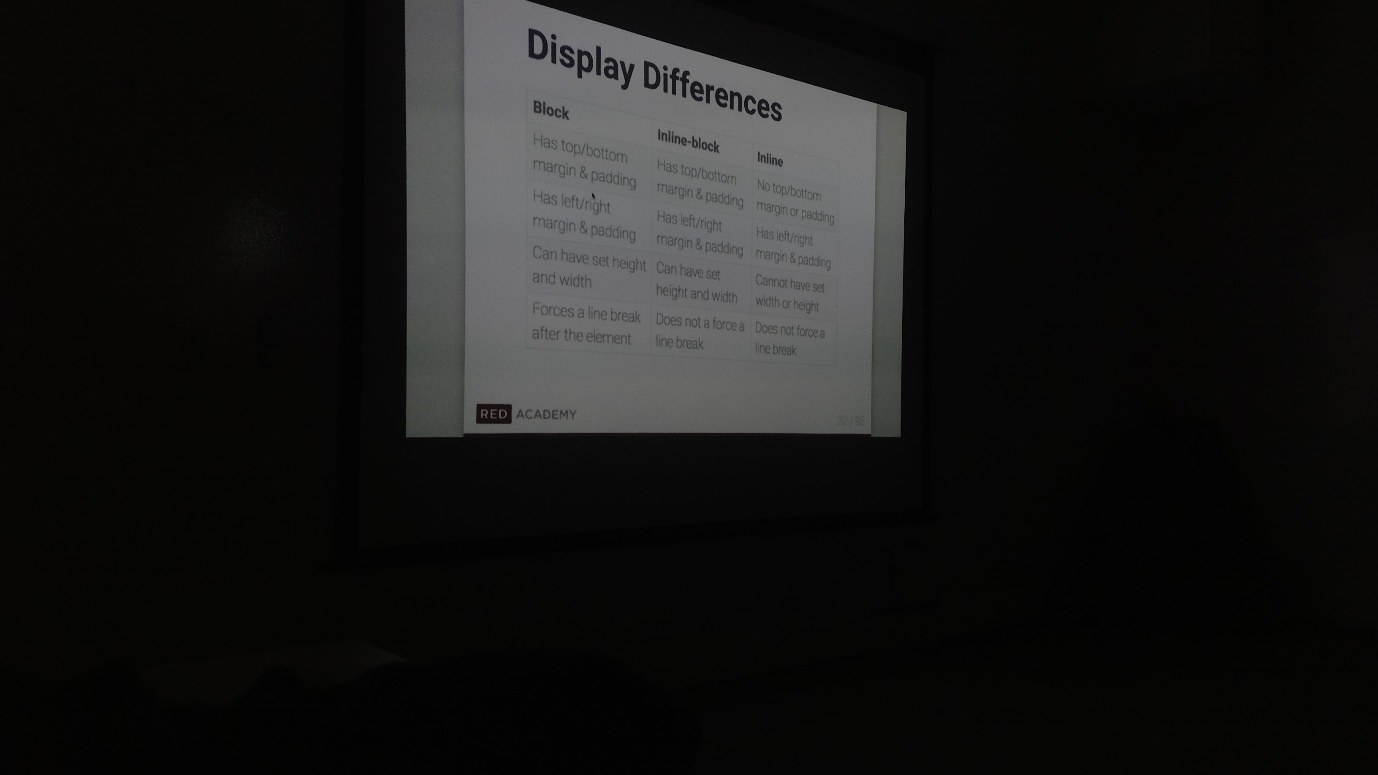
* Describes how and HTML element will be rendered on screen (what makes it pretty)
* HTML – Content Layer
* CSS – Presentation Layer
* 3 different ways you can apply it
  + Inline (not advisable anymore)
    - P { background -color: red;}
      * P = Selector
      * { = Declaraction Start
      * Background-color = property
      * Red = value
  + Internal Stylesheet
  + External Stylesheet (preferred method)
    - Uses link element to link - <link rel=stylesheet” href=”style.css”>
    - Then in your stylesheet you have all the information relating to your styling
    - Advantages to using this method are…
      * Makes it easy to link the same stylesheet to multiple pages across website
      * Much easier to make global changes to your css that are immediately picked up throughout the website
      * Your websites content is clearly separated from your websites style
* Can change everything like color, font family, font style, line height, etc.
* Always put the link to your stylesheet in the head portion of the HTML script

Boxes and Blocks

* Every HTML element is wrapped into an invisible box
* The box model gives additional properties that we can adjust for each HTML too. Including
  + Margin
  + Border
  + Padding
  + Width
  + Height
* Good demo here - <https://codepen.io/guyroutledge/pen/hgpez>
* We can specify if we want to just apply margin or border, etc. to top, bottom, left or right (eg border-top:1px;). There’s also the shorthand that if you just give one number, it applies to all. If you give two, first number is for top and bottom, second is for left and right. Three numbers then first one is top, second is left and right and third is bottom. 4 properties applies it to top, then right, then bottom then left
* We can also specify if we want the border or margin, etc. to be colored, solid or dotted, dashed, etc (e.g. border: 1px dotted red;)

Block vs Inline elements

* HTML elements will either be Block or inline elements
* Block elements will always appear to start a new line
* Som blok elements include <img>, <a>, <strong>
* We can override an elements default display behaviour by using the display property in our CSS.



The C in CSS

* The idea of cascading order is the key to how CSS works on HTML elements

Specificty

* How specificity applied to the cascade:

1. Where the specificity is equal the rule that comes last applies
2. Where specificity isn’t equal, the more specific rule applies

* When there’s two conflicting rules and neither are specific, the last one will be taken into account
* Where one is more specific than the other, it will override the other rule

Inheritance

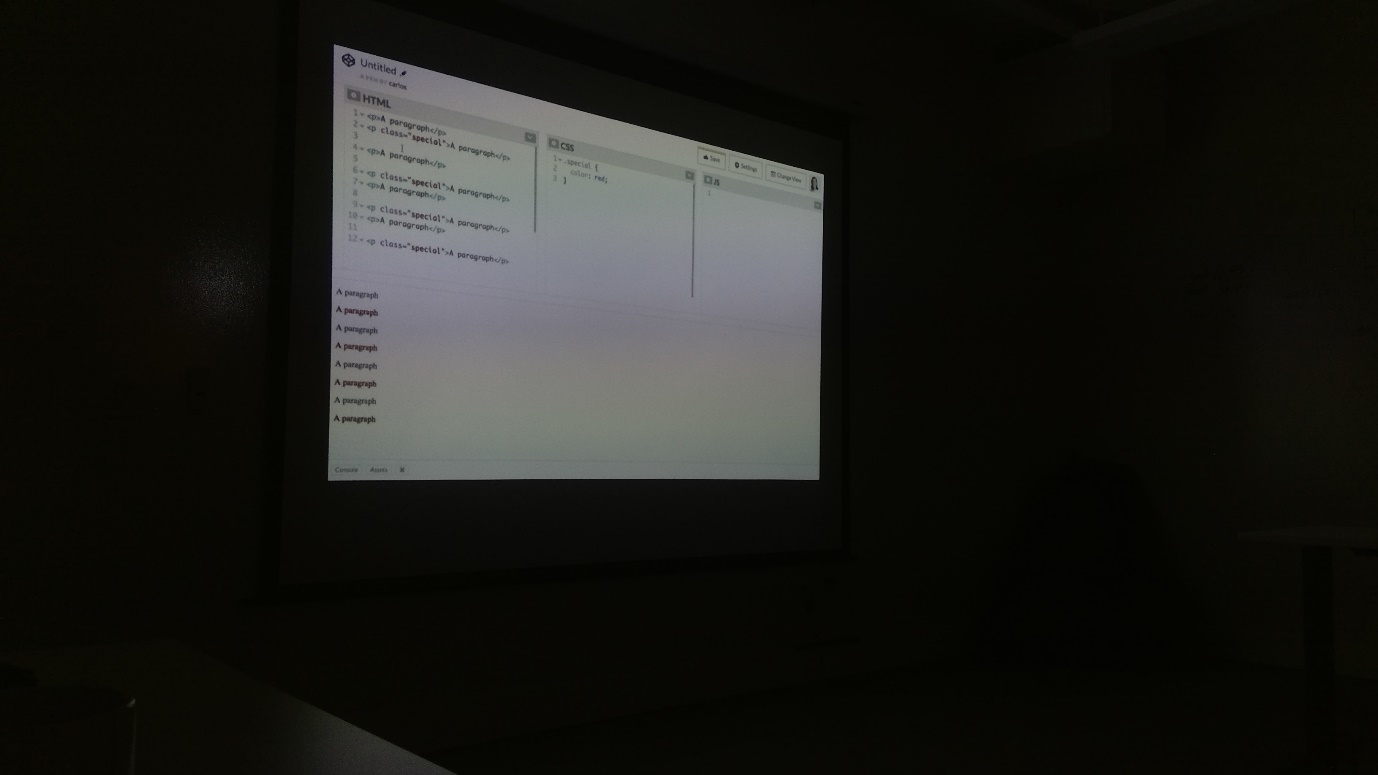
* Anything that we apply to a parent element will also apply to children elements

Class and ID

* What are they?
  + Every HTML element can have a class and ID attributes added to it like this

<h1 class=”site-title” id=”masthead”>My Awesome Websit</h1>

* Classes and IDS make it much easier to target specific CSS (and JS) to elements in our HTML
* Whats thedifference?
  + Think about an actual school. A student can be in multiple classes but eah student will have a unique name
  + Similarly if an HTML element ha an ID, that ID must be unique to the element
  + But a class can be applied to as many elements as you like



* In your css
  + To target a class, use a dot plus the class name

.site-title{ font-weight:bold;}

* + To target an ID, use a # plust the ID name

#masthead {font-weight;Bold;}

* You can specify multiple classes for an element by separating with a space
* And if you wanted to target
* You can target elements (.element), class (.class) and id’s (#id) using css. You can do these separately or together (either .element.class#id or .element#id etc.)
* If you separate the first and subsequent .element #id) it will only target the id’s which are in that element.
* In general, you’ll want to stick to classes wherever possible.
* IDs are considered to be more specific then classes in you CSS, which makes the ID based styles you apply to your website harder to override later on.

NAMING TIPS

* Classes and ID names must start with a letter or an underscore character
* Choose semantic class and ID names
* Remember that ID names must be unqiue to the element

Attribute Selector

* We can always target form inputs by their id or class, but we can also target entire types of form elements using the attribute selector in CSS.

Pseudo- Classes

* Are special built in keywords you can add to elements to describe some dynamic state of the element
* You can use the :link and :visited pseudo-classes for links and :hover, :active and :focus pseudo-classes for a broader array of elements
* Pseudo- classes also allow us to select very specific child elements inside of a parent

POSITION

* Working with position
* We can take even more control over our layouts using the CSS position property. Ther position property can take a specific array of values
  + Static (default)
  + Relative
  + Absolute
  + Fixed
* Relative Positioning
  + Moves and element realative to the amount that you specify e.g.
    - Position: relative;
    - Left: 299px;
  + On it’s own, having position: relative won’t do anything. Specifying Left: 299px will make it go 299px to the left
  + The movement of that element does not impact the position of the elements around it (if you’re using the realative positioning)
  + We can move using px, %, rem (realative to whats on the body. E.g. 2rem is twice as big as what’s specified on the body font size),
  + Also of note, children are relative to their parents, not to the entire page
* Absolute Positioning
  + Like realative positioning, but the rest of the elements no longer think that the effected element is there (e,g. if you move it to the right, the ones around it will take up the space that it was in)
* Fixed Positioning
  + Will fix an element so that it stays in that spot no matter where you scroll (e.g. a message us or a banner. It will fix this position throughout the whole page)
* Z-index
  + With elements that have relative,absolute or fixed position set we can apply a property called z-index.
  + Lets us specify the stacking order of non-static overlapping elements
  + Elements with a higher z-index will appear on top
  + Everything has a default z-index of 0. 1 will put it in front, -1 will put it behind
* Color on the Web
  + 4 different options
    - Color names
    - HEX codes
    - RGB values
      * Can specify opacity (transparent vs solid) using the 4th value. E.g. rgb(234,67,67,0.5); will make it half solid/transparent
    - HSL values
      * Like RGB, but hue, saturation and likeness. Rarely ever used
  + Background Properties
    - CSS gives us a number of tools for controlling the backgrounds of our HTML elements
    - We can specify
      * Background color
      * Background image (using a URL)
      * And if we specify a background-image URL we can also set these properties for it
        + Background repeat-Whether to repeat and image across the X or Y axis
        + Background-attachment – Whether the img should scroll with the browser or remaing fixed in one place
        + Background-position-for non repeated images, specify if it should be anchored to left, center or right and the top center or bottom
      * Background shorthand
        + Can specify multiple properties at once

Background:#29932 url(“images/fda.png”) no repeat center top;

You don#t need to specify. CSS will just figure it out because those vales are all mutually exclusive

* + - * Background size
        + Can do things like
        + contain, this will make it so that the size will be contained within the background that it’s in.
        + Cover will make sure that the entire element is covered by the background.
        + Can also specify size (e.g. 30% or 200 px)
      * Can have multiple backgrounds by separating images with a comma
        + Background:url(gfaho.png), url(fewon.png);

The first image you stipulate will be on top

CSS Resets, Developer Tools & Resources

* What is a CSS Reset and why do we need one?
  + Browsers can try to override some of your css settings. Doing a reset allows you to override those attempts.
  + There’s a few popular ones – Eric Meyer Reset or Normalize.css
  + Should just go to those sites and either link to the website in your css sheet or copy and paste the contents into your stylesheet
* Dev Tools
  + <http://discover-devtools.codeschool.com/>
  + <https://jigsaw.w3.org/css-validator/>
  + Mozilla Developer Network
  + Stack Overflow
  + CSS Tricks
  + Smashing Magazine