

CS 338: Graphical User Interfaces

Lecture 2-1: Web Development I

Some materials adapted from
<http://developer.mozilla.org>
<http://w3schools.com>
<http://getbootstrap.com>

Dev Week!

- Almost all of classes will be devoted to some theory, some discussion, some implementation
- This week, though, is all about development
 - Quick run through a full stack
 - We'll flesh this out in future weeks

Working Example

- Let's say you're a fan of goats. A BIG fan.
- You'd like to build a web site for all those goat aficionados like yourself: **The Goat Pasture**
- Your target users:
 - People who want to learn about goats
 - People who want to adopt a goat
- This example will help us to demonstrate all the core principles of full-stack web development.
- We will start simple and keep adding layers...

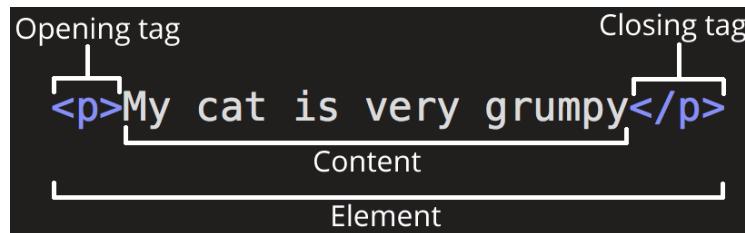
HTML

- HTML = HyperText Markup Language
 - Not a programming language — it's a markup language
- First introduced by Tim Berners-Lee in 1989-1991
- Considered an essential component of the beginnings of the World Wide Web
 - Gave the Web the concepts of “pages” and “hyperlinks”

HTML

- Elements

- A standard *element* has an opening and closing tag



- Some elements are “empty” (no opening/closing)

```

```

- An element can have one or more *attributes*

```
<p class="editor-note">My cat is very grumpy</p>
```

HTML

- Useful tags

<html>	The entire page
<header>	Header information
<body>	Main content
<h1>, <h2>, ...	Headers at level 1, 2, ...
<p>	Text paragraph
	Image
, 	List, list item
<table>, <tr>, <td>	Table, row, cell
<div>	Section

- And many more

HTML

- A basic web page document:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    <p>This is my page</p>
  </body>
</html>
```

HTML

- Key observation: Web page <-> hierarchy of objects

The Washington Post

Democracy Dies in Darkness

In the News Glitter bomb Yemeni mother Tear-gas photo Baby 'Adolf' Tucker Carlson Lin Wang Ocasio-Cortez Huntington Bank Cinder the bear Springsteen Man. U.

Introducing Post Reports
The Post's premier daily podcast. Unparalleled reporting, insight and analysis — for your ears.

Why President Trump is shutting down his charity
0:00 / 24:46
Add to Apple Podcasts, Google Podcasts, Stitcher

Veteran judge upends hopes of Trump allies as he spotlights Flynn's misdeeds

Supporters of former national security adviser Michael Flynn thought the independent-minded jurist would reveal overreach by special counsel Robert S. Mueller III and the FBI.

By Carol D. Leonnig and Rosalind S. Helderman •

- Flynn's sentencing delayed after judge says he might not avoid prison
- Flynn's transformation from storied officer to heated partisan

Trump backs off demand for \$5 billion for wall, but budget impasse remains ahead of shutdown

Ahsan Jadoon, 10, feeds chickens on the rooftop of his uncle's home in Rawalpindi. (Sarah Caron for The Post)

The U.S. had a 'chicken in every pot.'
Pakistan aims for chickens on every plot.

With Pakistan facing dire macroeconomic and fiscal crises, the backyard poultry project has been met with widespread derision. But

On Tonight's Podcast

- Suit alleges "illegal conduct" at charity.
- Policing where murders go unsolved.

Video

Joyce Koli/The Washington Post

Trump shuts down foundation amid allegations of illegal conduct

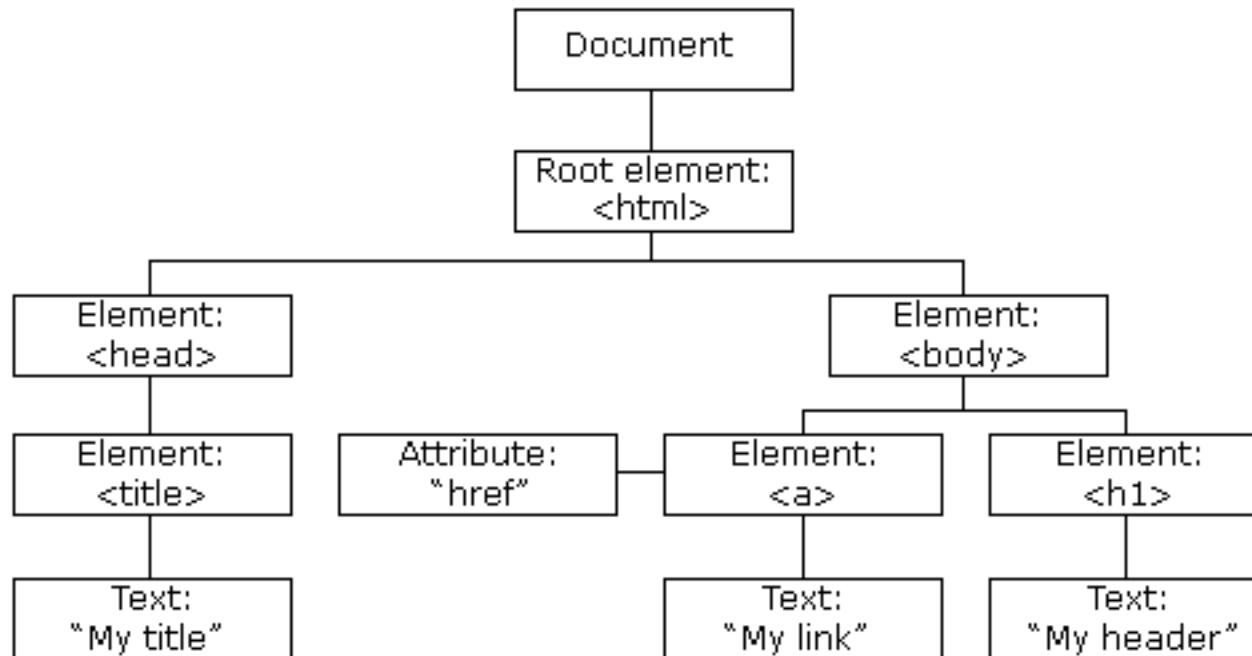
UP NEXT

How Martha McSally got a Senate seat despite losing the election

- Check out developer tools on Chrome, Safari, ...

HTML

- DOM = Document Object Model
 - The data structure that underlies the page, giving it both its structure and its content



Example

- Back to The Goat Pasture.
- Let's say we want the main sections to include:
 - “About Us” – tell our story
 - “Goat School” – a place for learning about goats
 - “Adopt a Goat” – an interactive area to browse goats up for adoption, select and read about them, etc.
- What might the home/splash page look like?

Example

The diagram illustrates a website layout with the following components:

- Header:** A dark green horizontal bar at the top containing navigation links: "Home", "About Us", "Goat School", "Adopt a Goat", and "Account".
- Main Title:** A large, bold, black font title "The Goat Pasture" centered above the main content area.
- Text Placeholder:** Below the title is a placeholder text "- Image Placeholder:** A large gray rectangular box below the slogan, labeled with placeholder text "- Action Buttons:** Two light blue rounded rectangular buttons at the bottom, labeled "[action 1](#)" and "[action 2](#)".

Example

Home About Us Goat School Adopt a Goat Account

The Goat Pasture

Love goats? Want a goat? We're here for you!



Learn about Goats Adopt a Goat

- How can we translate this to basic HTML?

```
<!DOCTYPE html>
<html>

<head>
    <title>The Goat Pasture - Home</title>
</head>

<body>

    <div>
        <a href="/">Home</a>
        <a href="/about">About Us</a>
        <a href="/school">Goat School</a>
        <a href="/adopt">Adopt a Goat</a>
    </div>

    <div>
        <h1>The Goat Pasture</h1>
        <p>Love goats? Want a goat? We're here for you!</p>
        <p></p>
        <p>
            <a href="/school">Learn about Goats</a>
            <a href="/adopt">Adopt a Goat</a>
        </p>
    </div>

</body>

</html>
```

```
<!DOCTYPE html>
<html>

<head>
    <title>The Goat Pasture</title>
</head>

<body>

    <div>
        <a href="/">Home</a>
        <a href="/">About Us</a>
        <a href="/">Goat School</a>
        <a href="/">Adopt a Goat</a>
    </div>

    <div>
        <h1>The Goat Pasture</h1>
        <p>Love goats? Want a goat? We're here for you!</p>
        <p>
            <a href="#">Learn about Goats</a>
            <a href="#">Adopt a Goat</a>
        </p>
    </div>

</body>

</html>
```

[Home](#) [About Us](#) [Goat School](#) [Adopt a Goat](#)

The Goat Pasture

Love goats? Want a goat? We're here for you!



[Learn about Goats](#) [Adopt a Goat](#)

Example

- The basic structure has no styling
- We can add styles with the “style” attribute
 - More spacing for menus?
`margin-right: 1rem;`
 - Color the menu bar?
`background: #406C39;`
 - Color the menu text?
`color: #fff;`
 - Center the main section?
`text-align: center;`

[Home](#) [About Us](#) [Goat School](#) [Adopt a Goat](#)

The Goat Pasture

Love goats? Want a goat? We're here for you!



[Learn about Goats](#) [Adopt a Goat](#)

```
<!DOCTYPE html>
<html>

<head>
    <title>The Goat Pasture - Home</title>
</head>

<body>

    <div style="background: #406C39;">
        <a href="/" style="color: #fff; margin-right: 1rem;">Home</a>
        <a href="/about">About Us</a>
        <a href="/school">Goat School</a>
        <a href="/adopt">Adopt a Goat</a>
    </div>

    <div style="text-align: center;">
        <h1>The Goat Pasture</h1>
        <p>Love goats? Want a goat? We're here for you!</p>
        <p></p>
        <p>
            <a href="/school">Learn about Goats</a>
            <a href="/adopt">Adopt a Goat</a>
        </p>
    </div>

</body>

</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      Home
    </title>
  </head>

  <body>
    <div style="text-align: center;">
      <a href="#">Home</a>
      <a href="#">About</a>
      <a href="#">Contact</a>
      <a href="#">Logout</a>
    </div>

    <div style="text-align: center;">
      <h1>Welcome</h1>
      <p>This is a simple static website</p>
      <p>Created by [Your Name]</p>
      <p>[Your Website URL]</p>
    </div>

  </body>
</html>
```



[Learn about Goats](#) [Adopt a Goat](#)

HTML

- We could style the entire page this way if we wanted, adding a “style” attribute to every element
- But this would come with some major drawbacks:
 - Every element is independent of the others;
If you change one, there’s a high chance of inconsistency
 - Same style specified many times
 - Bad: Longer HTML text
 - Worse: Difficult for designers/programmers to stay consistent within a single page or between pages
- There’s gotta be a better way...

CSS

- CSS = Cascading Style Sheets
- Developed by Håkon Wium Lie in 1994,
based on earlier developments in markup languages
- My two cents:
 - Very easy to understand the basics
 - Very hard to master the (sometimes crazy) details
 - We'll try to give you some tips & tricks that cover
90% of what you'll need for most web sites

CSS

- CSS specification is typically in a separate **.css** file
- Specify the styles for a tag:

.css

```
h1 {  
    color: red;  
}
```

- Specify the styles for a **class**:

.css

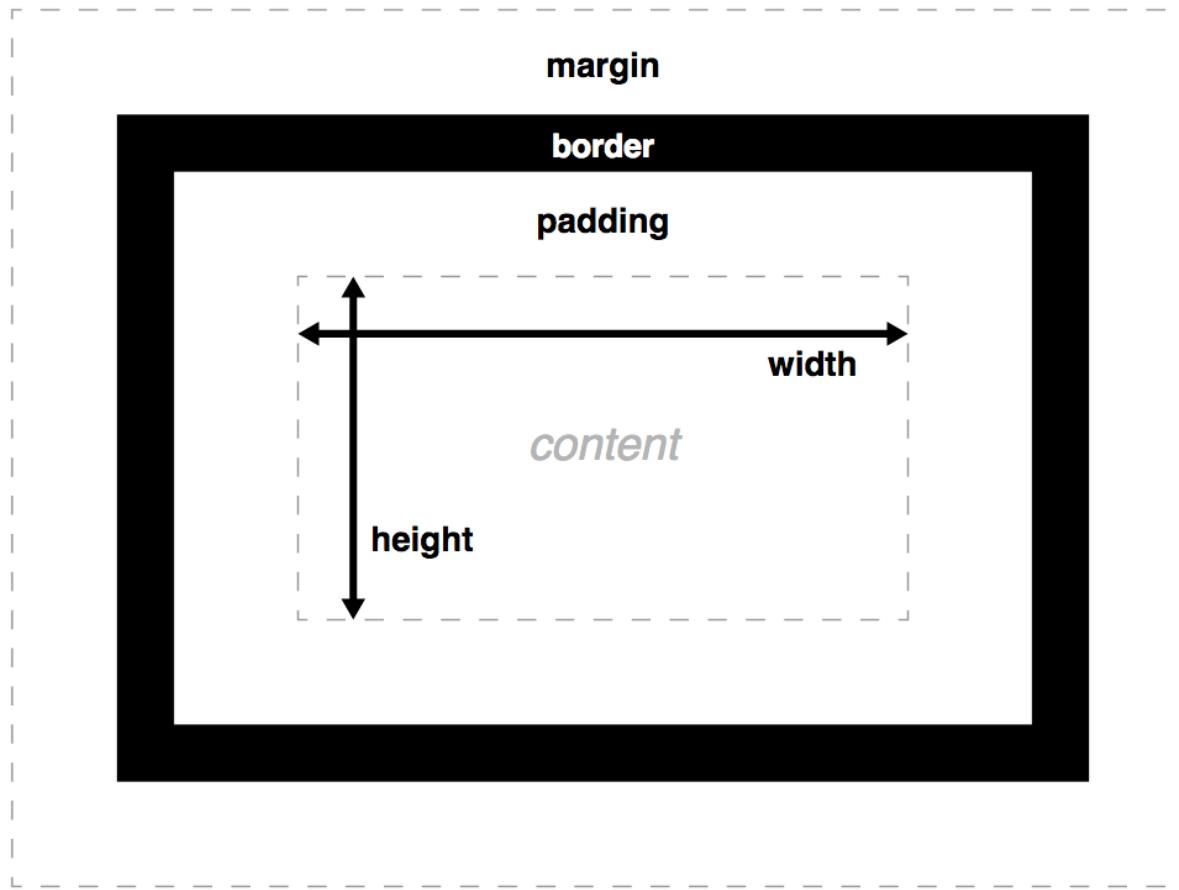
```
.my-class {  
    color: red;  
}
```

.html

```
<h1 class="my-class"> ... </h1>
```

CSS

- Margin vs. padding (+ border and content)



■ Units

– Absolute

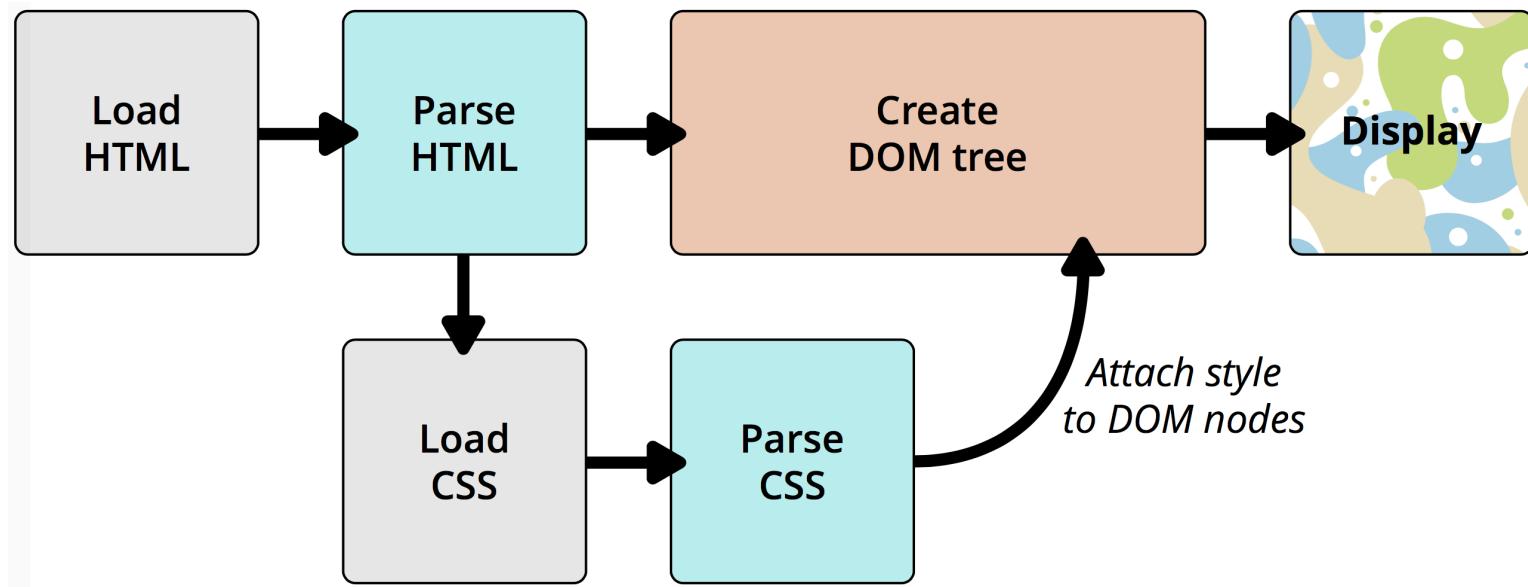
cm	centimeters
mm	millimeters
in	inches (1in = 96px)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

– Relative

em	Relative to the font-size of the element (2em means 2 times the size of the current font) (affected by parent!)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

CSS

- The loading/parsing/rendering process



Example

- Let's move the styles we saw earlier to **style.css**

```
.menu-bar {  
    background: #406C39;  
}  
  
.menu-item {  
    margin-right: 1rem;  
    color: #fff;  
}  
  
.splash {  
    text-align: center;  
}
```

- And then change the HTML accordingly...

```
<!DOCTYPE html>
<html>

<head>
    <title>The Goat Pasture - Home</title>
    <link rel="stylesheet" href="style3.css">
</head>

<body>

    <div class="menu-bar">
        <a href="/" class="menu-item">Home</a>
        <a href="/about" class="menu-item">About Us</a>
        <a href="/school" class="menu-item">Goat School</a>
        <a href="/adopt" class="menu-item">Adopt a Goat</a>
    </div>

    <div class="splash">
        <h1>The Goat Pasture</h1>
        <p>Love goats? Want a goat? We're here for you!</p>
        <p></p>
        <p>
            <a href="/school">Learn about Goats</a>
            <a href="/adopt">Adopt a Goat</a>
        </p>
    </div>

</body>

</html>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>
```

```
        <link>
```

```
</head>
```

```
<body>
```

```
    <div>
```

```
        <a>
```

```
        <a>
```

```
        <a>
```

```
        <a>
```

```
    </div>
```

```
    <div>
```

```
        <h1>
```

```
        <p>
```

```
        <p>
```

```
        <p>
```

```
    </
```

```
    </div>
```

```
</body>
```

```
</html>
```

Home About Us Goat School Adopt a Goat

The Goat Pasture

Love goats? Want a goat? We're here for you!



[Learn about Goats](#) [Adopt a Goat](#)

Example

- Let's improve this...
 - Eliminate margin around entire page
 - Change the font
 - Fix the menu bar
 - Soften the color of the motto text
 - Make bottom links look like buttons

[Home](#) [About Us](#) [Goat School](#) [Adopt a Goat](#)

The Goat Pasture

Love goats? Want a goat? We're here for you!



[Learn about Goats](#) [Adopt a Goat](#)

Example

- Eliminate page margin, change page font

```
body {  
    margin: 0;  
    font-family: "Avenir";  
}
```

Home About Us Goat School Adopt a Goat

The Goat Pasture

Love goats? Want a goat? We're here for you!



[Learn about Goats](#) [Adopt a Goat](#)

Example

- Fix the menu bar (and menu items)

```
.menu-bar {  
    padding: 1rem;  
    background: #406C39;  
}  
  
.menu-item {  
    margin-left: 1rem;  
    color: #fff;  
    text-decoration: none;  
}
```

Home About Us Goat School Adopt a Goat

The Goat Pasture

Love goats? Want a goat? We're here for you!



[Learn about Goats](#) [Adopt a Goat](#)

Example

- Make bottom links look (and act) like buttons

```
.splash a {  
    margin: .5rem;  
    padding: .5rem 1rem;  
  
    background: #475;  
    border: 1px solid #253;  
    border-radius: .5rem;  
  
    color: #fff;  
    text-decoration: none;  
}  
  
.splash a:hover {  
    background: #697;  
}
```

Home About Us Goat School Adopt a Goat

The Goat Pasture

Love goats? Want a goat? We're here for you!



Learn about Goats

Adopt a Goat

Example

- Make buttons

```
.splash a {  
    margin: .5rem 1rem;  
    padding: .5rem 1rem;  
  
    background: #475;  
    border: 1px solid #253;  
    border-radius: .5rem;  
  
    color: #fff;  
    text-decoration: none;  
}  
  
.splash a:hover {  
    background: #697;  
}
```

Only applies to “a” tags
that are descendants of
“splash” class elements

like buttons

Goat School Adopt a Goat

The Goat Pasture

Love goats? Want a goat? We're here for you!



Learn about Goats

Adopt a Goat

Example

- Make bottom links look (and act) like buttons

```
.splash a {  
    margin: .5rem;  
    padding: .5rem 1rem;  
  
    background: #475;  
    border: 1px solid #253;  
    border-radius: .5rem;  
  
    color: #fff;  
    text-decoration:  
}  
  
.splash a:hover {  
    background: #697;  
}
```



The screenshot shows a website layout. At the top is a green navigation bar with white text containing links for "Home", "About Us", "Goat School", and "Adopt a Goat". Below the navigation bar is a section with the title "The Goat Pasture" in large bold letters, followed by the text "Love goats? Want a goat? We're here for you!". A large image of a goat's head and upper body is centered below this text. At the bottom of the page are two green buttons with white text: "Learn about Goats" and "Adopt a Goat". A light blue callout bubble with a black outline points from the left towards the "Learn about Goats" button, containing the explanatory text: "Applies to these “a” tags when the user is hovering over the link".

CSS

- We could do all the styling we needed with a combination of CSS and HTML
- However:
 - We'd be doing everything from scratch.
 - More importantly, it would be difficult to keep a consistent look & feel across an entire web site.
 - (We'll discuss this issue in detail later in the course.)
- Fortunately, there are packages that make styling much easier — and come with other goodies (involving dynamic interactions etc.)
- Let's explore what we'll be using in this course...

Bootstrap

- Front-end CSS/HTML/JavaScript framework
- Developed by Mark Otto and Jacob Thornton at **Twitter** to use internally (originally “Blueprint”)
- Released as an open-source project in 2011
- Perhaps the most popular framework out there, though as usual, there are other options
- Most importantly for us, we use Bootstrap as an example of what such frameworks provide

Bootstrap

- One big useful feature: Loads of styled classes and re-styled tags
 - Bootstrap overwrites the styles for built-in tags like `<h1>`, `<p>`, etc.
 - It also provides new classes that can be added to elements to give them a Bootstrap look
 - These classes provide a consistent look & feel across the interface and across pages
- Example: Instead of the “splash” section we created earlier, we can use Bootstrap’s “jumbotron” and related classes...

```
<!DOCTYPE html>
<html>

<head>
    <title>The Goat Pasture - Home</title>
    <link href="../course/lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="style5.css">
</head>

<body>

    <div class="menu-bar">
        <a href="/" class="menu-item">Home</a>
        <a href="/about" class="menu-item">About Us</a>
        <a href="/school" class="menu-item">Goat School</a>
        <a href="/adopt" class="menu-item">Adopt a Goat</a>
    </div>

    <section class="jumbotron text-center">
        <h1 class="jumbotron-heading">The Goat Pasture</h1>
        <p class="lead text-muted">Love goats? Want a goat? We're here for you!</p>
        <p></p>
        <p>
            <a href="/school" class="btn btn-secondary my-2 mr-2">Learn about Goats</a>
            <a href="/adopt" class="btn btn-secondary my-2">Adopt a Goat</a>
        </p>
    </section>

</body>

</html>
```

Example

- Our homemade version
- Bootstrap jumbotron

Home About Us Goat School Adopt a Goat

The Goat Pasture

Love goats? Want a goat? We're here for you!



Learn about Goats Adopt a Goat

Home About Us Goat School Adopt a Goat

The Goat Pasture

Love goats? Want a goat? We're here for you!



Learn about Goats Adopt a Goat

Bootstrap

- We could emulate Bootstrap's styling with just CSS, but it's much easier to just inherit and use the styles
- At the same time, it's often the case that you'd like to change the out-of-the-box styles
 - In this case, you can simply override Bootstrap
 - Let's say we want a white background for the jumbotron instead of the gray default:

```
.jumbotron {  
    background: white;  
}
```

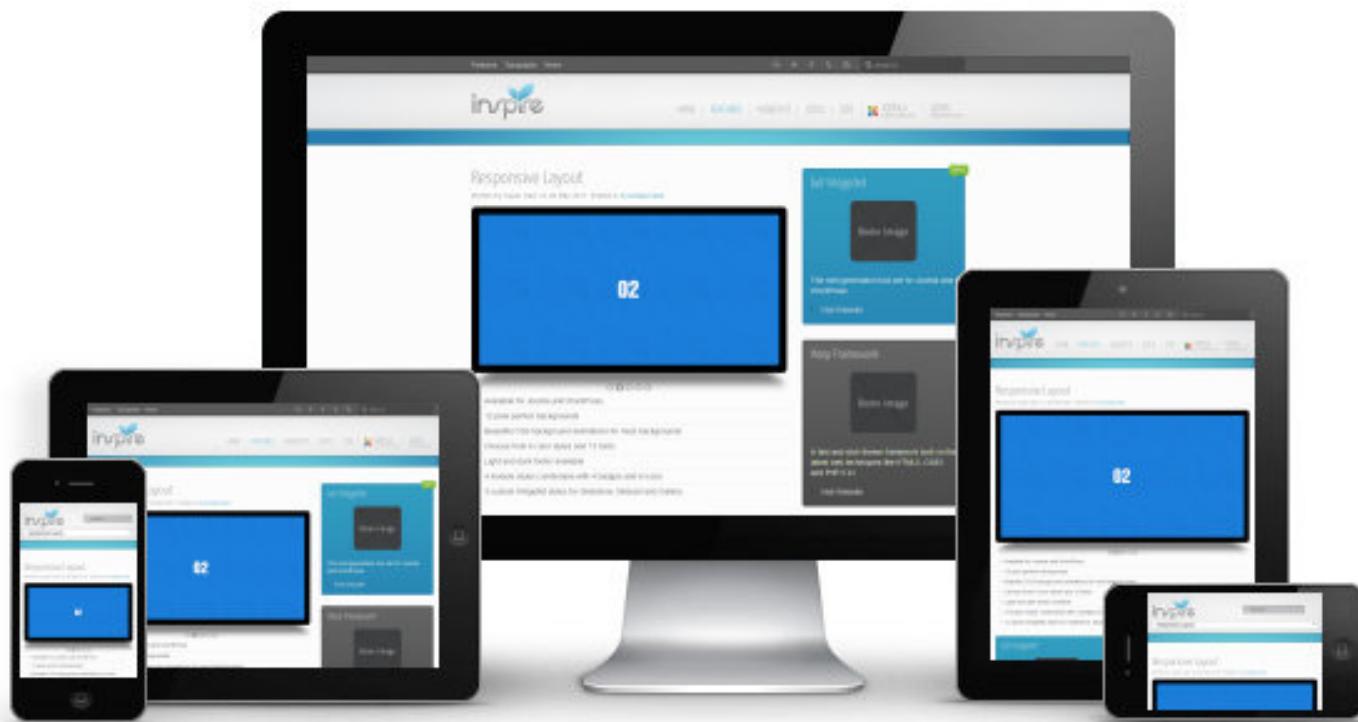
- Is it always this easy? Not at all. More on this later.

Layout

- Layout is notoriously difficult in HTML
- In the old days, sites often used tables (`<table>`) to lay out content
- CSS “position” and “float” also played a role
- Today, there are advances to make this easier
 - E.g., CSS “flexbox” layout methods
- Even with these advances, in some ways, layout is harder today than 15-20 years ago
 - Why?

Layout

■ Responsive layout



from <https://www.perfect-web.co/blog/45-responsive-layout-for-yoo-inspire-joomla-template>

Bootstrap

- We'll focus on the Bootstrap approach to layout, since it's well known, powerful, and responsive
- Bootstrap includes...
 - Wrapping containers to help at a high level
 - A grid system to help place components
 - A flexible media object to help with layout of repetitive small components (e.g., comments, items)
 - Responsive utility classes to change things for different screen sizes

Bootstrap

■ Containers

- Content is generally placed into a “container”
- A default container is centered and adjusts its width responsively to the current screen size

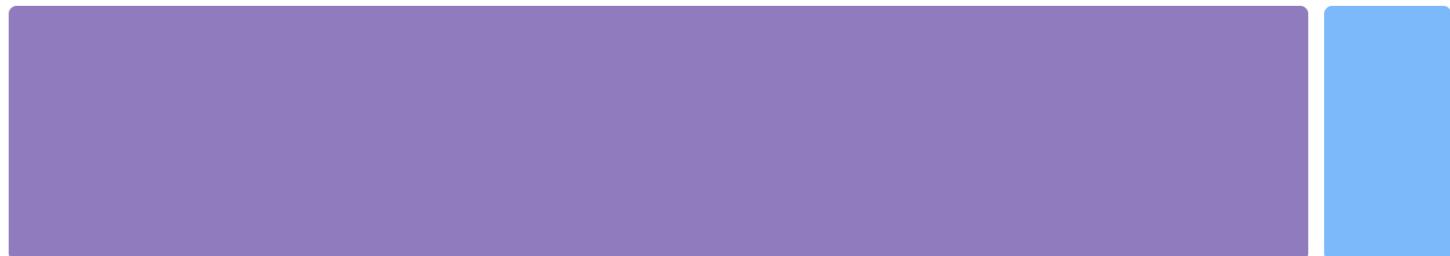


```
<div class="container">  
  <!-- Content here -->  
</div>
```

Bootstrap

■ Containers

- Content is generally placed into a “container”
- A fluid container spans the entire width of the window



```
<div class="container-fluid">  
  ...  
</div>
```

Bootstrap

- Grid system
 - Inside a container, you create **rows**, and then create **columns** within each row
 - Bootstrap assumes **12 columns** in each row, which you can fill as you'd like
 - But this number 12 is important, since the numbers on the classes represent part of these 12
 - The HTML tags for these elements is typically <div> (though it doesn't need to be)

Bootstrap

- Grid system
 - 4 equal-width columns
 - 2 spec-width columns: [8/12] , [4/12]



```
<div class="row">
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
</div>
<div class="row">
  <div class="col-8">col-8</div>
  <div class="col-4">col-4</div>
</div>
```

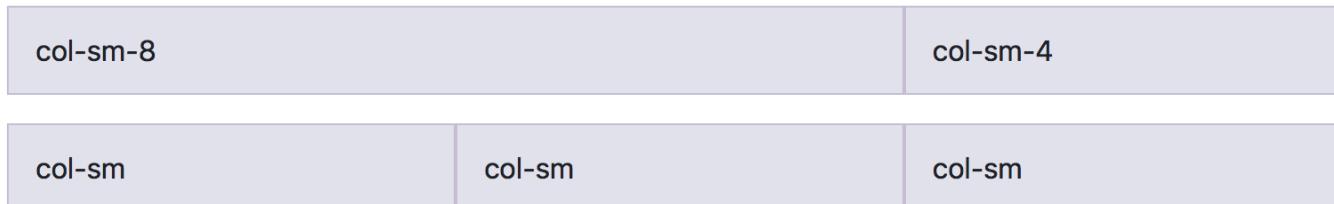
Bootstrap

- **Breakpoints define where the layout changes**
 - Each is associated with a class prefix (sm, md, lg, xl)
 - When using a class with this prefix, the class will be active for any size \geq that prefix
 - E.g., 4 x “col-md” will be $\frac{1}{4}$ -width columns at size \geq Medium, but will stack vertically at smaller sizes

	Extra small $<576\text{px}$	Small $\geq 576\text{px}$	Medium $\geq 768\text{px}$	Large $\geq 992\text{px}$	Extra large $\geq 1200\text{px}$
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-

Bootstrap

- Grid system
 - Columns using the “sm” prefix will stack at smaller sizes



```
<div class="row">
  <div class="col-sm-8">col-sm-8</div>
  <div class="col-sm-4">col-sm-4</div>
</div>
<div class="row">
  <div class="col-sm">col-sm</div>
  <div class="col-sm">col-sm</div>
  <div class="col-sm">col-sm</div>
</div>
```

- Classes can be mixed together as well...

```
.col-12 .col-md-8
```

```
.col-6 .col-md-4
```

```
.col-6 .col-md-4
```

```
.col-6 .col-md-4
```

```
.col-6 .col-md-4
```

```
.col-6
```

```
.col-6
```

```
<!-- Stack the columns on mobile by making one full-width and the other half-width -->
<div class="row">
  <div class="col-12 col-md-8">.col-12 .col-md-8</div>
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
</div>

<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
<div class="row">
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
</div>

<!-- Columns are always 50% wide, on mobile and desktop -->
<div class="row">
  <div class="col-6">.col-6</div>
  <div class="col-6">.col-6</div>
</div>
```

Bootstrap

- Grid system
 - To read more about the grid system and simultaneously see the responsive layout in action, check out:
<https://getbootstrap.com/docs/4.1/layout/grid/>
 - (Try resizing your browser window to see the effects on the different layout examples.)

Bootstrap

- Helpful spacing classes
 - Bootstrap includes a few classes that help to space things out the way you want
 - Specifically, they represent a quick-and-easy way to specify **margin** and **padding**
 - See next slide...

The classes are named using the format `{property}{sides}-{size}` for `xs` and `{property}{sides}-{breakpoint}-{size}` for `sm`, `md`, `lg`, and `xl`.

Where `property` is one of:

- `m` - for classes that set `margin`
- `p` - for classes that set `padding`

Where `sides` is one of:

- `t` - for classes that set `margin-top` or `padding-top`
- `b` - for classes that set `margin-bottom` or `padding-bottom`
- `l` - for classes that set `margin-left` or `padding-left`
- `r` - for classes that set `margin-right` or `padding-right`
- `x` - for classes that set both `*-left` and `*-right`
- `y` - for classes that set both `*-top` and `*-bottom`
- blank - for classes that set a `margin` or `padding` on all 4 sides of the element

Where `size` is one of:

- `0` - for classes that eliminate the `margin` or `padding` by setting it to `0`
- `1` - (by default) for classes that set the `margin` or `padding` to `$spacer * .25`
- `2` - (by default) for classes that set the `margin` or `padding` to `$spacer * .5`
- `3` - (by default) for classes that set the `margin` or `padding` to `$spacer`
- `4` - (by default) for classes that set the `margin` or `padding` to `$spacer * 1.5`
- `5` - (by default) for classes that set the `margin` or `padding` to `$spacer * 3`
- `auto` - for classes that set the `margin` to `auto`

The classes are named using the format `{property}{sides}-{size}` for `xs` and `{property}{sides}-{breakpoint}-{size}` for `sm`, `md`, `lg`, and `xl`.

Where `property` is one of:

- `m` - for classes that set `margin`
- `p` - for classes that set `padding`

Where `sides` is one of:

```
<section class="jumbotron text-center">
  <!-- other stuff -->
  <p>
    <a href="/school" class="btn btn-secondary my-2 mr-2">Learn about Goats</a>
    <a href="/adopt" class="btn btn-secondary my-2">Adopt a Goat</a>
  </p>
</section>
```

- `y` - for classes that set both `*-top` and `*-bottom`
- blank - for classes that set a `margin` or `padding` on all 4 sides of the element

Where `size` is one of:

- `0` - for classes that eliminate the `margin` or `padding` by setting it to `0`
- `1` - (by default) for classes that set the `margin` or `padding` to `$spacer * .25`
- `2` - (by default) for classes that set the `margin` or `padding` to `$spacer * .5`
- `3` - (by default) for classes that set the `margin` or `padding` to `$spacer`
- `4` - (by default) for classes that set the `margin` or `padding` to `$spacer * 1.5`
- `5` - (by default) for classes that set the `margin` or `padding` to `$spacer * 3`
- `auto` - for classes that set the `margin` to `auto`

Browser Tools

- When you’re doing web development, it’s super helpful to have developer tools at your disposal
- The major browsers (Chrome, Safari, Firefox) all include a variety of tools
- Given what we’ve covered just now, two functions are particularly useful:
 - 1. Point-and-click exploration of the DOM
 - 2. Figuring out “computed” CSS styles

Browser Tools

- Point-and-click exploration of the DOM
 - E.g., in Chrome

The screenshot shows a web page titled "The Goat Pasture" featuring a large image of a goat. The browser's developer tools are open, specifically the Elements tab, which displays the HTML structure of the page. A blue arrow points from the text "Love goats? Want a goat? We're here for you!" in the page content down to its corresponding element in the DOM tree. Another blue arrow points from the top right corner of the browser window to the title bar of the developer tools.

Home About Us Goat School Adopt a Goat

The Goat Pasture

Love goats? Want a goat? We're here for you!



Learn about Goats Adopt a Goat

```
<!doctype html>
<html>
  <head>...</head>
  ...<body> == $0
    ><div class="menu-bar">...</div>
    ><section class="jumbotron text-center">
      <h1 class="jumbotron-heading">The Goat
      Pasture</h1>
      <p class="lead text-muted">Love goats? Want
      a goat? We're here for you!</p>
      ><p>...</p>
      ><p>...</p>
    </section>
  </body>
</html> body
Styles Computed Event Listeners >
Filter :hov .cls +
element.style { }
body {
  margin: 0;
  font-family: "Avenir";
}
body {
  margin: 0;
  font-family: -apple-
  system, BlinkMacSystemFont, "Segoe
  UI", Roboto, "Helvetica Neue", Arial, sans-
  serif, "Apple Color Emoji", "Segoe UI
  Emoji", "Segoe UI Symbol", "Noto Color Emoji";
  font-size: 1rem;
  font-weight: 400;
  line-height: 1.5;
  color: #212529;
  text-align: left;
}
: Console What's New ×
Highlights from the Chrome 71 update
Hover over a Live Expression to highlight a
DOM node
Hover over a result that evaluates to a node to
highlight that node in the viewport
```

Browser Tools

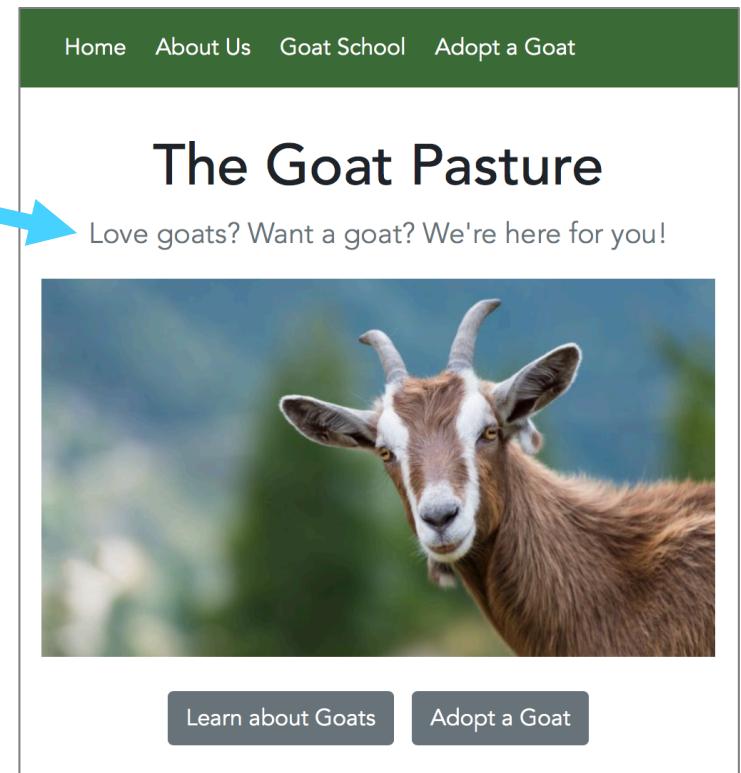
- Figuring out “computed” CSS styles

- Let's say you want to change the color of the jumbotron motto

```
<p class="lead text-muted">  
    Love goats? Want a goat? We're  
    here for you!  
</p>
```

- We can just change the color of the .lead class... right?

```
.lead {  
    color: blue;  
}
```



Browser Tools

- Figuring out “computed” CSS styles

- No – .lead color is overridden by .text-muted color (because of !important)
 - Now we can make .lead !important, or ...
- This sort of problem happens a lot, and browser tools help to find & fix issues

The screenshot shows the 'Computed' tab of a browser's developer tools. It displays the final 'computed' style for an element, which is a combination of styles from multiple sources. The 'element.style' section shows a single rule. Below it, the 'p.lead.text-muted' section shows the individual source rules:

```
element.style {  
}  
.lead {  
    color: #00f;  
}  
.text-muted {  
    color: #6c757d !important;  
}  
.lead {  
    font-size: 1.25rem;  
    font-weight: 300;  
}
```

The 'color' property for the '.lead' class is highlighted in red, indicating it is being overridden by the '!important' declaration in the '.text-muted' class.

Coming up...

- That's a VERY quick overview of
 - HTML
 - CSS
 - Bootstrap (the CSS part)
- Next, we will do a quick overview of
 - JavaScript

Dev Week Continued

- So far, we saw:
 - HTML
 - CSS
 - Bootstrap (the CSS part)
- This gives us a foundation for structure and style
- But with only these components, the web page is completely static — just a unchanging page
- We continue our focus on:
 - JavaScript

JavaScript

- Dates back to 1995 (part of Netscape)
- Adopted by all major browsers today
- Interpreted
 - no compilation of code
- Weakly typed
 - code doesn't specify types of variables, arguments, etc.
- First-class functions
 - treats functions as values
 - store them, pass them as arguments, etc.

JavaScript

- Language tidbits
 - Many things look the same as Java
 - Operators
 - + , * , && , || , == , ...
 - Conditionals and loops
 - if , switch , for , while , ...

JavaScript

- Language tidbits
 - Some things are a little different
 - Variable declaration

```
var a = 2;
```

- Arrays and Objects (associative arrays)

```
var myArray = ["hello", "there"];
alert (myArray[0]);
```

```
var myAssoc = new Array();
myAssoc["name"] = "John";
alert (myAssoc["name"]);
```

JavaScript

- Language tidbits
 - A straightforward recursive function

```
function factorial(n) {  
    if (n == 0) return 1;  
    return n * factorial(n - 1);  
}
```

JavaScript

- Language tidbits
 - A function with *closure*

```
function incrementCreator() {  
    var count = 0;  
    return function () {  
        return ++count;  
    };  
}  
var inc = incrementCreator();  
alert(inc()); /  
alert(inc()); /  
alert(inc()); /
```

[from <http://en.wikipedia.org/wiki/JavaScript>]

JavaScript

- Adding JavaScript to a web page

- Loading a JS file in the header:

```
<script type="text/javascript" src="URL"></script>
```

- Embedding JS directly into the page

```
<html>
  <head ... ></head>
  <body>

    <script>
      document.write('Hello World!');
    </script>

  </body>
</html>
```

JavaScript

■ Handling events

- Event handling is normally tied to a DOM object
 - The object generates the event (e.g., when clicked); the handler receives and then handles the event
- Embedding the handler in HTML

```
<button id="mybutton" onclick="doSomething()>
    My Button
</button>
```

- Specifying the handler in JavaScript

```
button.onclick = "doSomething();"
OR
button.onclick = doSomething;
OR
button.onclick = function() { ... };
```

JavaScript

- Handling events
 - What work is done inside the handler?
 - Retrieve and/or compute data, if needed
 - We will deal with this in the future when we discuss AJAX
 - Find DOM elements to update

```
var el = document.getElementById("myElement");
```

- Change some property of these elements

```
el.style.color = "blue";  
el.style.fontSize = "20pt";
```

That's it for now!

- That's our very quick tour of a main structure for web development
- There are many, many, many more pieces to each of these components — we've only touched the surface
- We'll continue to flesh out more pieces this term, in the context of larger issues
 - E.g., Jquery, Flask, AJAX, database access, graphics, possibly user authentication and cookies