

# Web Programming

# Bootstrap

**Leander Jehl** | University of Stavanger

# CSS Frameworks



Write classes instead of CSS

# Fixed-width vs fluid layouts

- Fixed-width layout
  - Components inside a fixed-width wrapper have either percentage or fixed widths. Typically, grid systems.
- Fluid (or liquid) layout
  - Components have percentage widths (in % or em), thus adjust to the user's screen resolution



# Responsive design

- Tailoring layout specifically for the type of screen
  - E.g., three column layout for desktops, a two column layout for tablets, and a single column layout on smartphones
- Using a fluid grid and media queries in CSS

# **Front-end frameworks**

**(a.k.a. CSS boilerplates)**

- Provide a common structure
- Usually a mix of HTML, CSS, and JS code
- Typical components
  - Positioning blocks (often based on a grid system)
  - Typography style definitions for HTML elements
  - Solutions for browser (in)compatibility
  - Advanced components

# Grid systems

- 960gs  
<http://960.gs/>
- Golden grid  
<http://goldengridsystem.com/>
- Responsive grid  
<http://www.responsivegridsystem.com/>
- Gridiculo.us  
<http://gridiculo.us/>

# Complete frameworks

- Bootstrap  
<http://getbootstrap.com/>
- Foundation  
<http://foundation.zurb.com/>
- Gumby  
<http://gumbyframework.com/>
- YAML  
<http://www.yaml.de/>

# Pros and cons for using frameworks

## Advantages

- Speeds up processes
- Clean and tidy code
- Solutions to common CSS problems
- Browser compatibility
- Helpful in collaborative work

## Disadvantages

- (Often) mixes content and presentation
- Overhead
- Slower learning curve
- You don't learn to do it yourself



Write classes instead of CSS

**960gs**

# **960 Grid System**

**<http://960.gs/>**

- Provide commonly used dimensions based on a width of 960 pixels
  - Why 960?
- 12 or 16 columns
  - can be used separately or in tandem
  - each column has 10px of margin on the left and right

# Drupal, 12col grid

Get Started   Community & Support   Documentation   Download & Extend   Marketplace   About

## Drupal

Come for the software, stay for the community

Drupal is an open source content management platform used to power millions of websites and software applications. It's built, used, and supported by an active and diverse community of people from around the world.

[Drupal Homepage](#) [Login / Register](#)

**Why Choose Drupal?**

Drupal can be used to build everything from personal blogs to sophisticated enterprise applications. It features thousands of add-on modules and design themes that enable you to build any kind of site you can imagine. Drupal is free software that is flexible, robust, and constantly being improved by a community of hundreds of thousands of passionate users. Come join us!

[Get Started with Drupal](#)

**Sites Made with Drupal**

 The Economist  
Log in   Register  
Home   World   Business & Finance  
The Economist.com data migration to Drupal

Drupal is used by some of the biggest sites on the Web, like The Economist, Examiner.com and The White House. Read more Drupal success stories.

**Develop with Drupal**

Drupal is extensible, powerful, scalable, and flexible.

**Current activity:**  
6,988 modules  
862 themes  
3,740 active developers  
[Modules and Themes](#)

**This week:**  
31 new modules  
1,355 code commits  
5,410 issue comments  
[Explore Drupal modules and themes](#)

**Get Drupal Security Announcements**  
Keep your site secure.

 708,149 people in 230 countries speaking 182 languages power Drupal.

 DrupalCon Chicago  
March 7-10 2011

News   Docs Updates   Forum Posts   Commits

Support Drupal by voting in the 2010 Open Source Awards  
October 15, 2010

Packt Publishing, the company behind many well-known Drupal books, is also responsible for the Open Source Awards, an annual online event held to distinguish excellence among open Read more

beta.drupal.org is ready to QA!  
Distributed Authentication (the drupal.module) on drupal.org will be turned off November 1st, 2010

# Drupal, 12col grid

The screenshot shows the Drupal.org homepage utilizing a 12-column grid system. The layout is structured as follows:

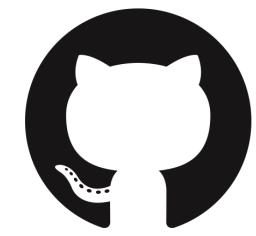
- Top Bar:** A dark purple header bar with the word "Drupal" in white. To its right are links for "Get Started", "Community & Support", "Documentation", "Download & Extend", "Marketplace", and "About".
- Search Bar:** A search input field with a magnifying glass icon and a "Search" button.
- Refine Your Search:** A sidebar with links to "Module", "Theme", "Documentation", "Forum & Issues", and "Group".
- Main Content Area:** Contains three main columns.
  - Why Choose Drupal?**: A section with text about Drupal's features and a "Get Started with Drupal" button.
  - Sites Made with Drupal**: A section featuring a thumbnail of "The Economist" website and a link to "The Economist.com data migration to Drupal".
  - Develop with Drupal**: A section with text about Drupal's extensibility and developer statistics, along with links to "Drupal API", "Download Drupal", "Security Info", "Handbook", "Modules and Themes", "Explore Drupal", and "modules and themes".
- Footer:** A dark footer bar with a blue "D" logo, the text "708,149 people in 230 countries speaking 182 languages power Drupal.", and a "Support Drupal by voting in the 2010 Open Source Awards" section.
- Bottom Footer:** A light blue footer bar with links for "News", "Docs Updates", "Forum Posts", and "Commits".

# Usage

- Each column has 10 pixels of margin on the left and right, which create 20 pixel wide gutters between columns

```
<div class="container_12"> → 12 or 16 columns  
  <div class="grid_*"> → * is the column width (1, 2, ... 12)  
    [...]  
  </div>  
  <div class="grid_*">  
    [...]  
  </div>  
  <div class="clear"></div> → separator between rows  
  <div class="grid_*">  
    [...]  
  </div>  
  [...]  
</div>
```

# Exercise #1



[https://github.com/dat310-2024/info/tree/master/  
exercises/bootstrap](https://github.com/dat310-2024/info/tree/master/exercises/bootstrap)

# Solution

⌚ solutions/bootstrap/exercise1.html



```
<div class="container_12">  
    <div class="grid_6">  
        <div class="blue"></div>  
    </div>  
    <div class="grid_6">  
        <div class="blue"></div>  
    </div>  
    <div class="clear"></div>  
    <div class="grid_4">  
        <div class="yellow"></div>  
    </div>  
    <div class="grid_4">  
        <div class="yellow"></div>  
    </div>  
    <div class="grid_4">  
        <div class="yellow"></div>  
    </div>
```

# **Bootstrap**

# **Bootstrap**

**<http://getbootstrap.com/>**

- Powerful front-end framework
- Developed by Twitter, open sourced in 2011
- Responsive, mobile-first
- Common HTML elements
- Custom CSS classes
- Custom HTML and CSS components
- Newest version 5.3
- Uses Flexbox

# Getting started

- Download or include it from CDN
- Include CSS for styling and layout

```
<!DOCTYPE html>           HTML5 doctype is required!
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/
bootstrap.min.css" integrity="sha384-
giJF6kkoqNQ00vy+HMDP7az0uL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1">
</head>
```

Bootstrap is mobile-first. To ensure proper rendering and touch zooming, include:

# Getting started

- Download or include it from CDN
- Include CSS for styling and layout

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/
bootstrap.min.css" integrity="sha384-
giJF6kkoqNQ00vy+HMDP7az0uL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1">
</head>
```

# Getting started

- Download or include it from CDN
- Include minified JavaScript for components

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/
bootstrap.min.css" integrity="sha384-
giJF6kkoqNQ00vy+HMDP7az0uL0xtbfIcaT9wjKHr8RbDVddVHyTfAAAsrekwKmP1">
  <!-- Latest compiled JavaScript -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/
bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/
t7LECLbyPA2x65Kgf800JFdroafW"></script>
</head>
```

# Getting started

- Download or include it from CDN
- Include minified JavaScript for components
  - JS only required for certain components:  
<https://getbootstrap.com/docs/5.0/getting-started/introduction/#components>

# Containers

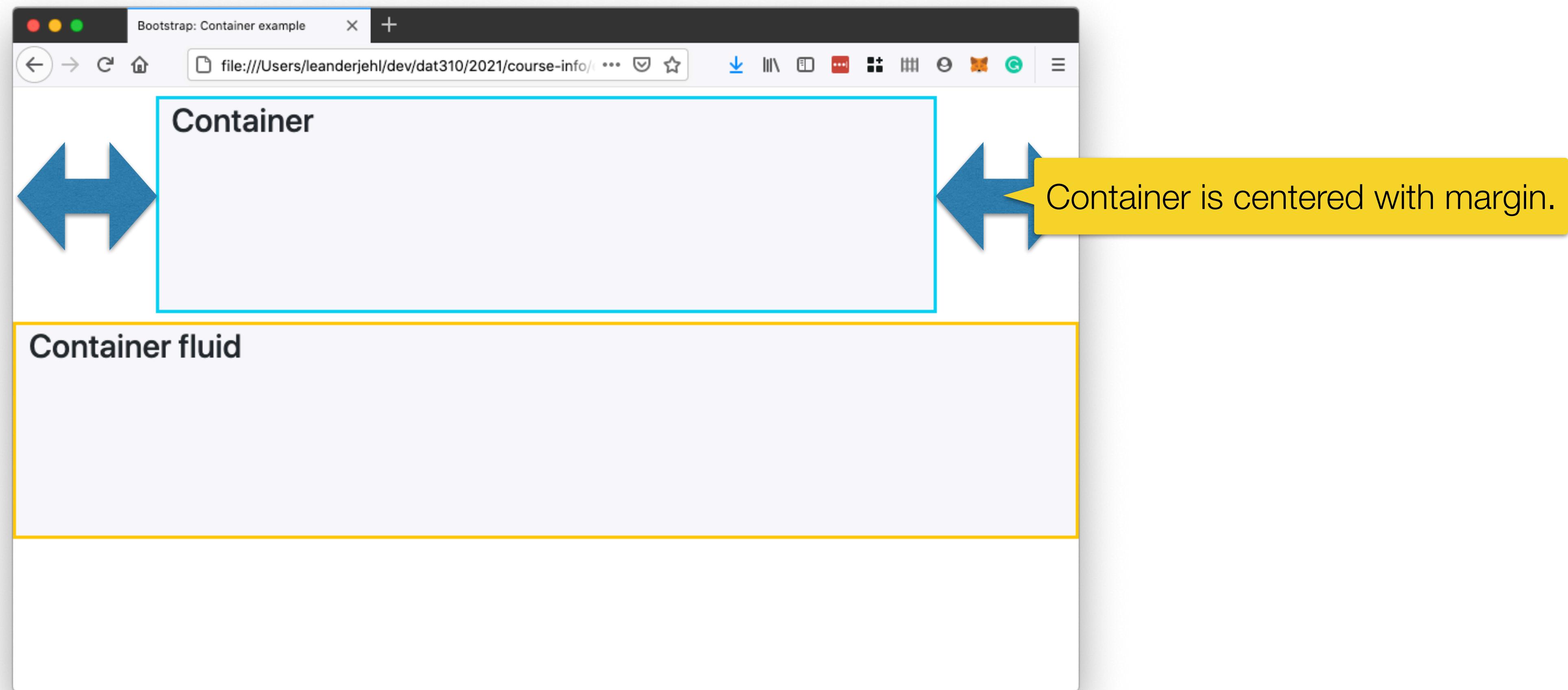
- Two container classes to wrap site contents

```
<div class="container">  
  
</div>
```

- **container** provides a responsive fixed width container
- **container-fluid** provides a full width container, spanning the entire width of the viewport
- Do not nest containers!

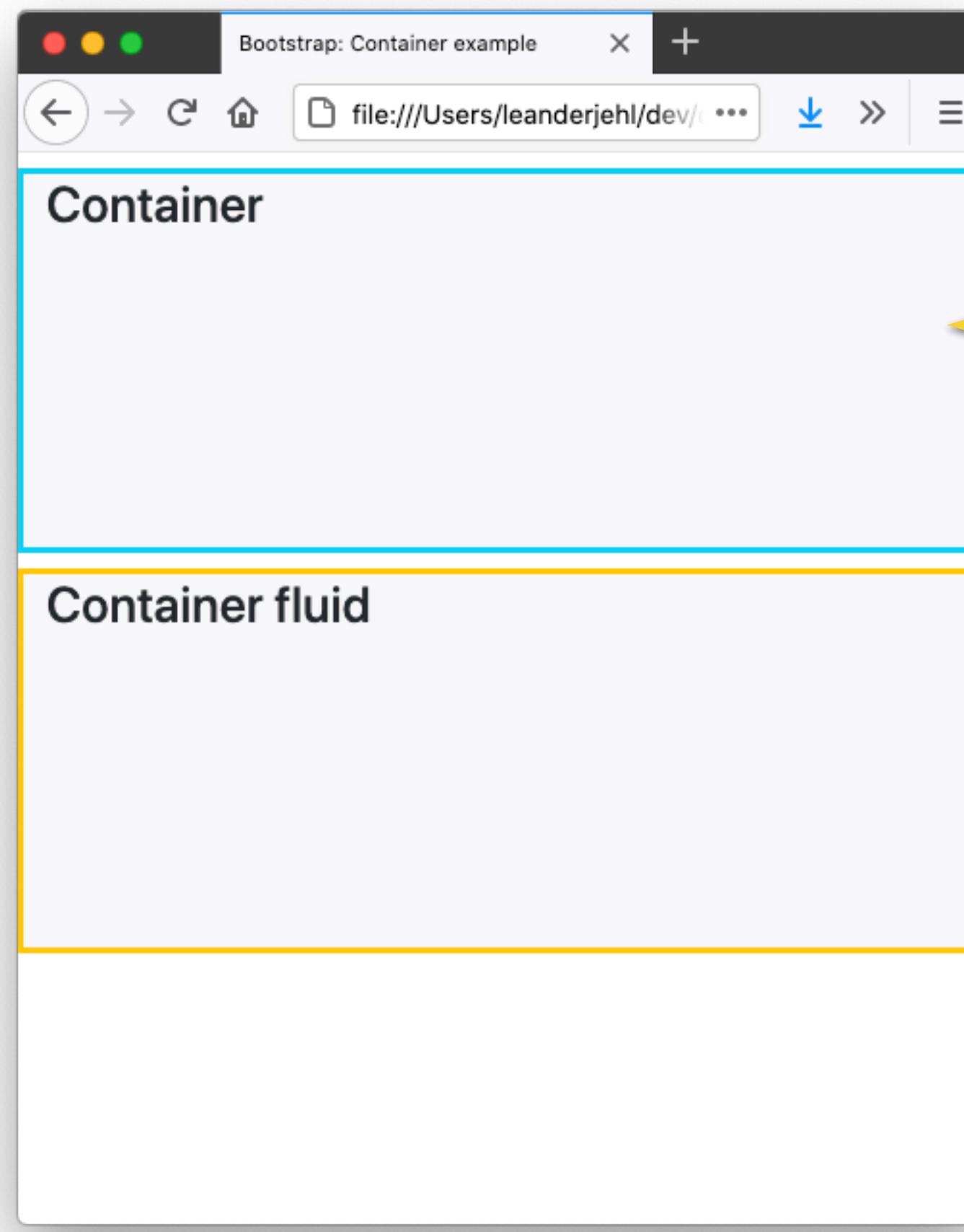
# Example

examples/bootstrap/container.html



# Example

examples/bootstrap/container.html



On small screen (mobile),  
container has no margin

# Rows and columns

- **row** class on the element containing multiple elements next to each other.
- **col** class on the elements.
- **col-\*** class on the elements with \* = **1, 2, ..., 12**
  - Specify width on 12 column grid
  - **col-6** column takes 50% width
  - **col-4** column takes 33% width
  - **col-3** column takes 25% width

# Example

examples/bootstrap/columns.html

The screenshot shows a web browser window with four examples of Bootstrap column usage:

- 12 Columns**: A row containing 12 "col-1" columns.
- Combined Columns (total 11)**: A row containing 5 columns with the following spans: col-2, col-2, col-3, col-3, and col-1. The total width is 11 columns, with 1 missing column left empty.
- Combined columns (total 13)**: A row containing 5 columns with the following spans: col-2, col-2, col-3, col-3, and col-1. The total width is 13 columns, with 1 missing column left empty. A note below states: "Columns more than 12 are wrapped."
- Variable columns**: A row containing 6 columns with the following spans: col, col, col-3, col-3, col, and col-2. The first two "col" classes grow to fill the row.

Missing columns are left empty.

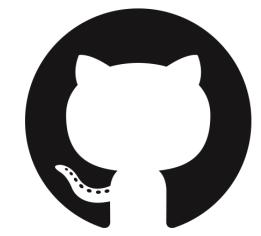
Columns more than 12 are wrapped.

Columns without number, `col` grow to fill the row.

# Rows and columns

- Columns build on flexbox and have the flexbox possibilites:
  - **align-items-center** on row vertically centeres items
  - **justify-content-**\* horizontal spread of items
    - **center** centered items
    - **between** equal space between items
    - **evenly** equal space between and around items

# Exercise #2a



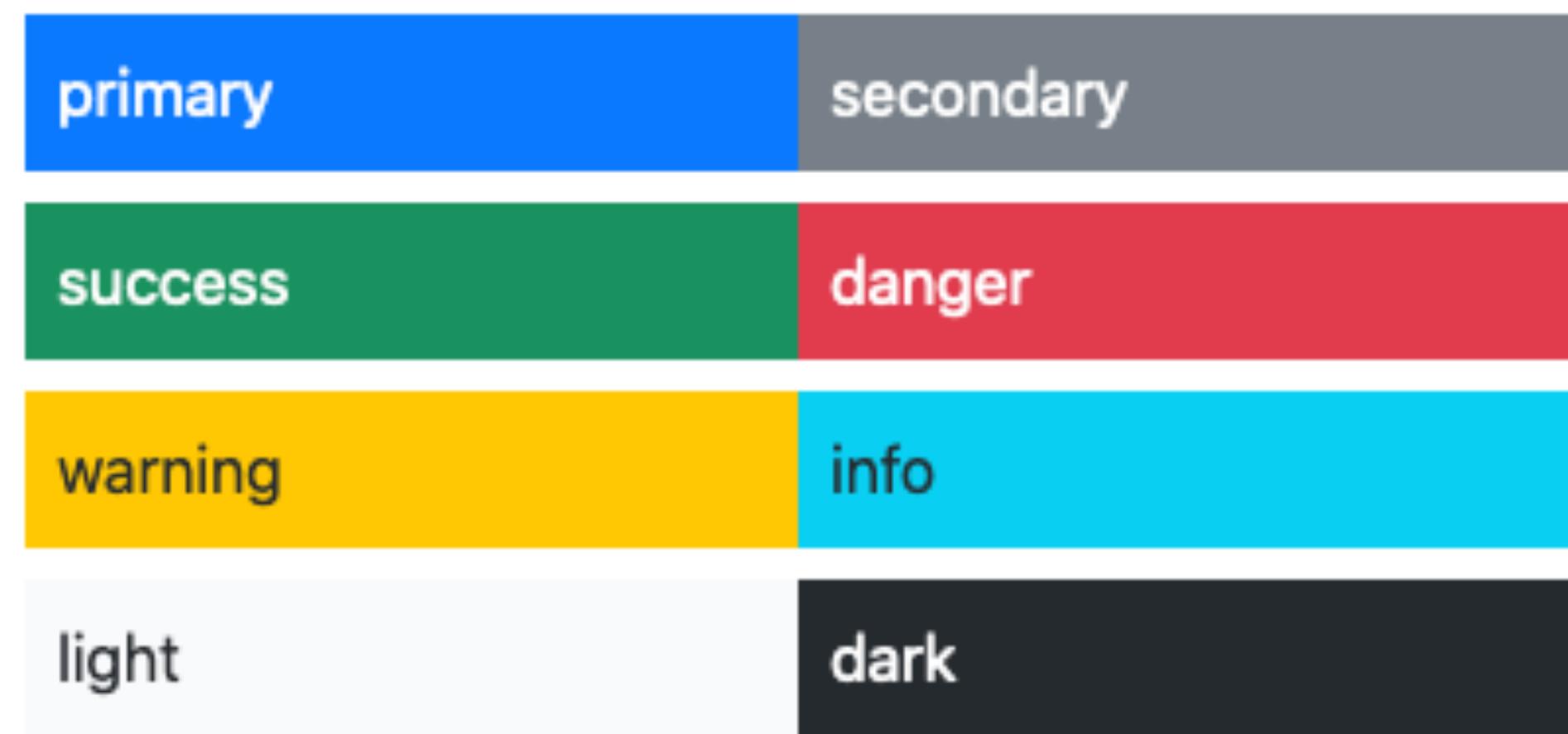
[https://github.com/dat310-2024/info/tree/master/  
exercises/bootstrap](https://github.com/dat310-2024/info/tree/master/exercises/bootstrap)

# Colors

- 8 Theme colors are predefined but configurable

primary, secondary, success,  
\* → danger, warning, info,  
light, dark

## Color Themes



Class	Effect
<b>bg-*</b>	background color
<b>text-*</b>	Color
<b>border-*</b>	Border color
<b>table-*</b>	Table background
<b>btn-*</b>	Button color

# Breakpoints

- Adjust columns (and other classes based on screen width)
- Responsive, i.e., columns rearrange automatically depending on the screen size

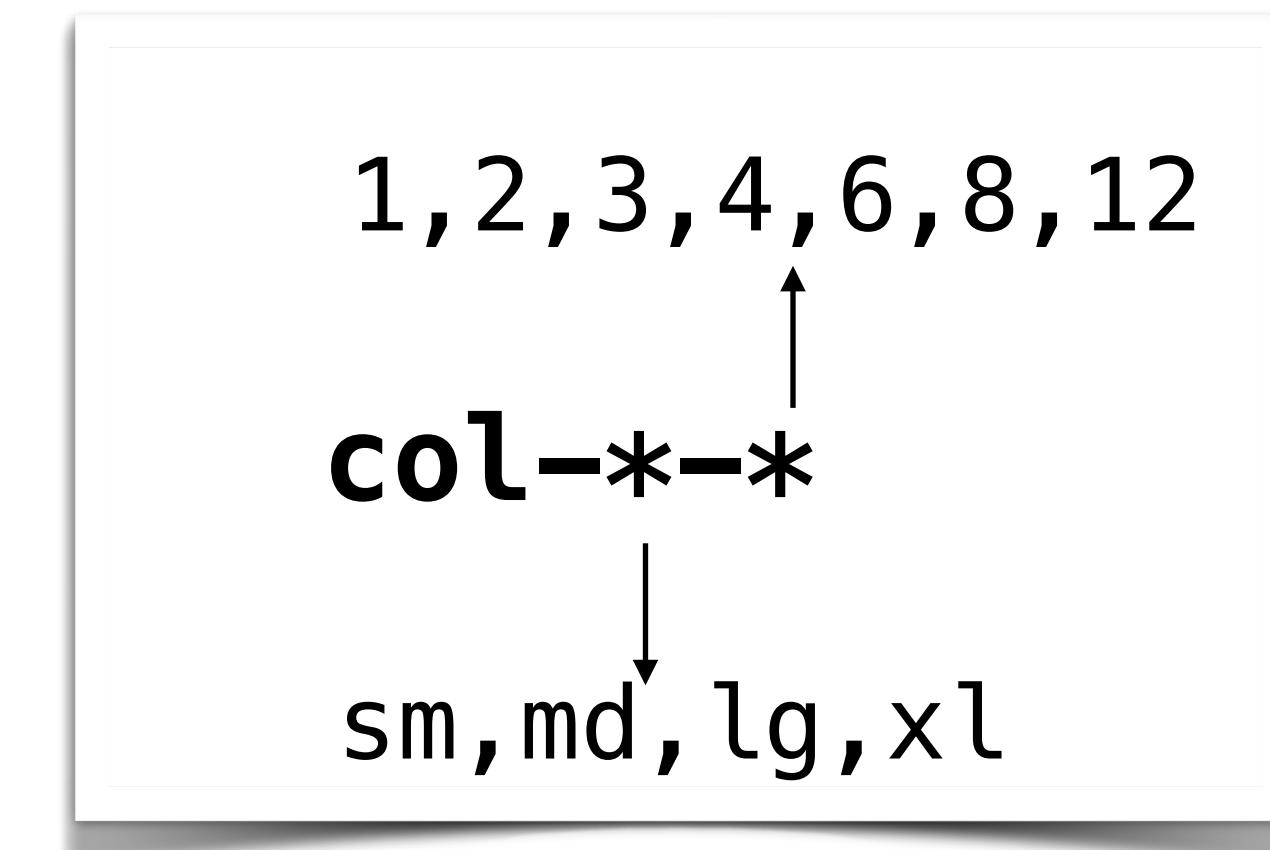
Breakpoint	Class infix	Dimensions
X-Small	None	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

# Breakpoints

- **container-md** change to fluid container for screen < medium

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

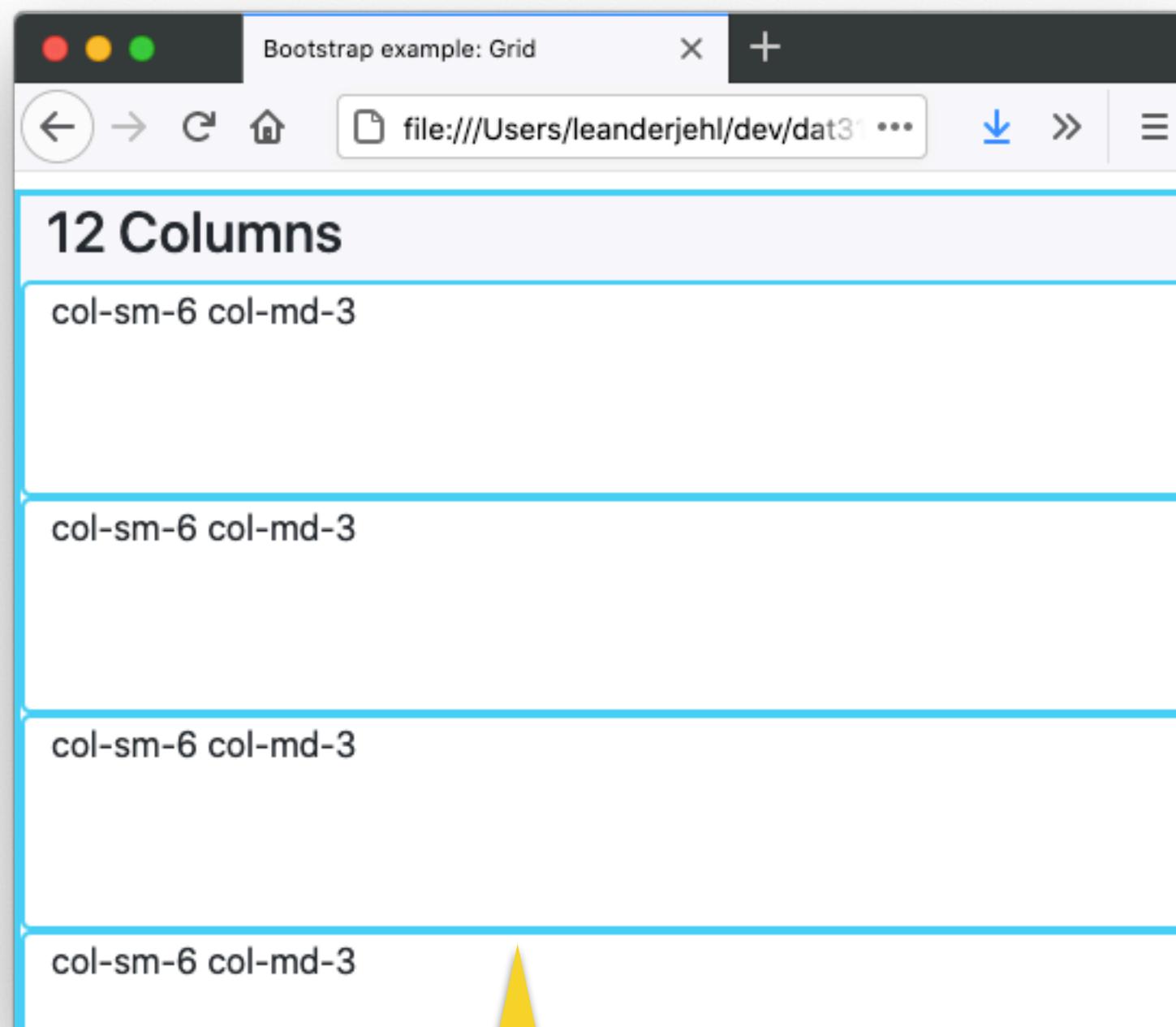
- **col-sm-6** takes 6 columns on **sm** and larger  
-> full width on **xs**.



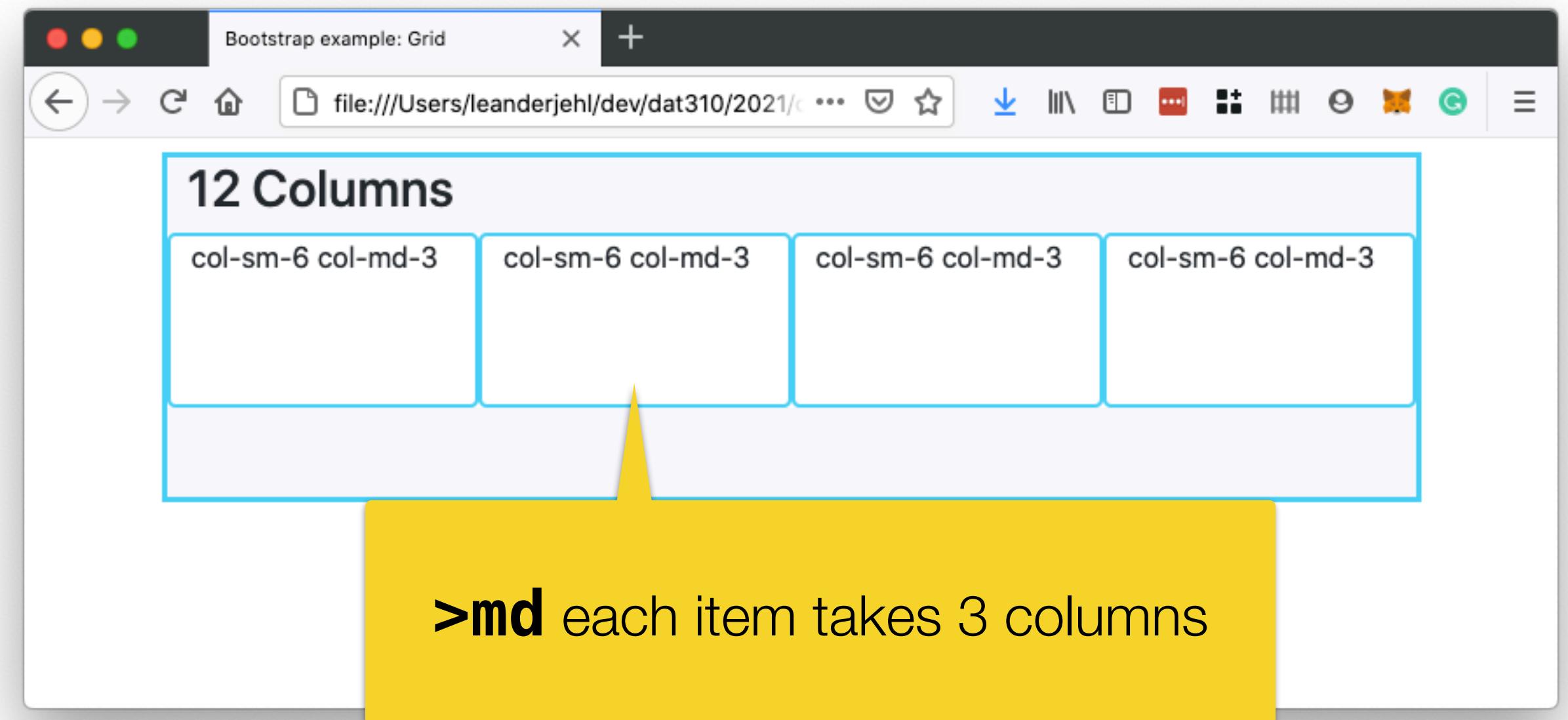
# Example

examples/bootstrap/grid.html

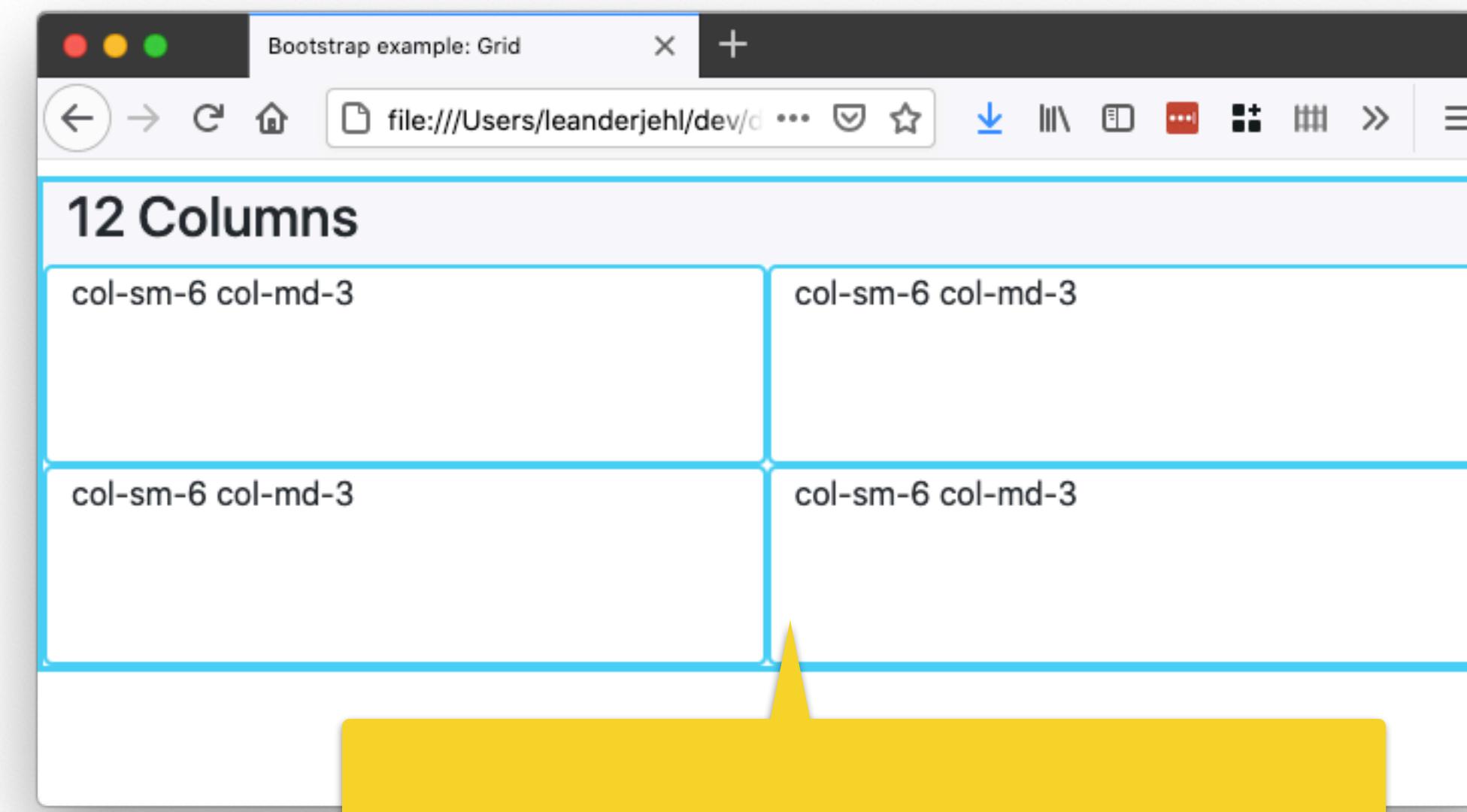
```
<div class="col-md-3 col-sm-6">  
  ...  
</div>
```



**xs** each item takes all columns



**>md** each item takes 3 columns

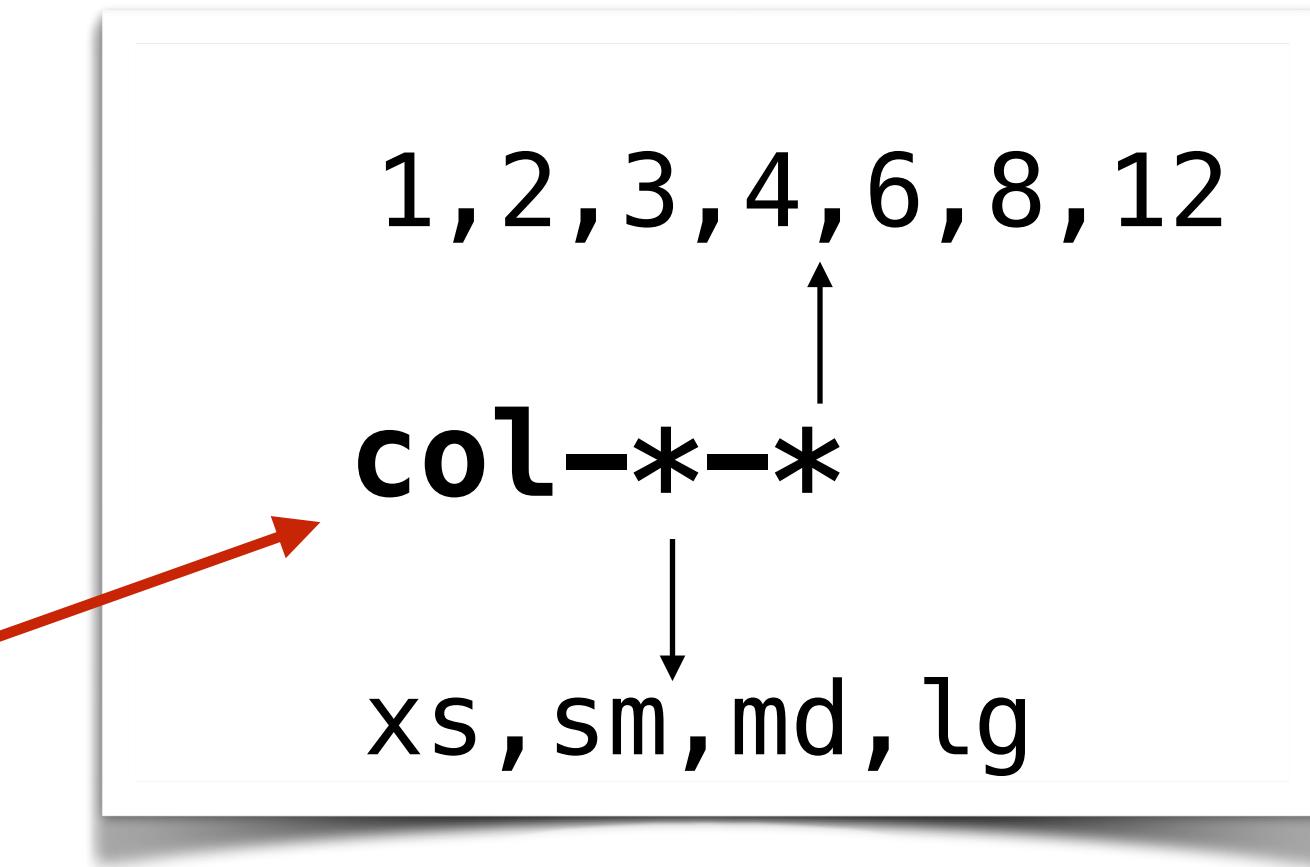


**sm** each item takes 6 columns

# Grid system (2)

- Basic structure

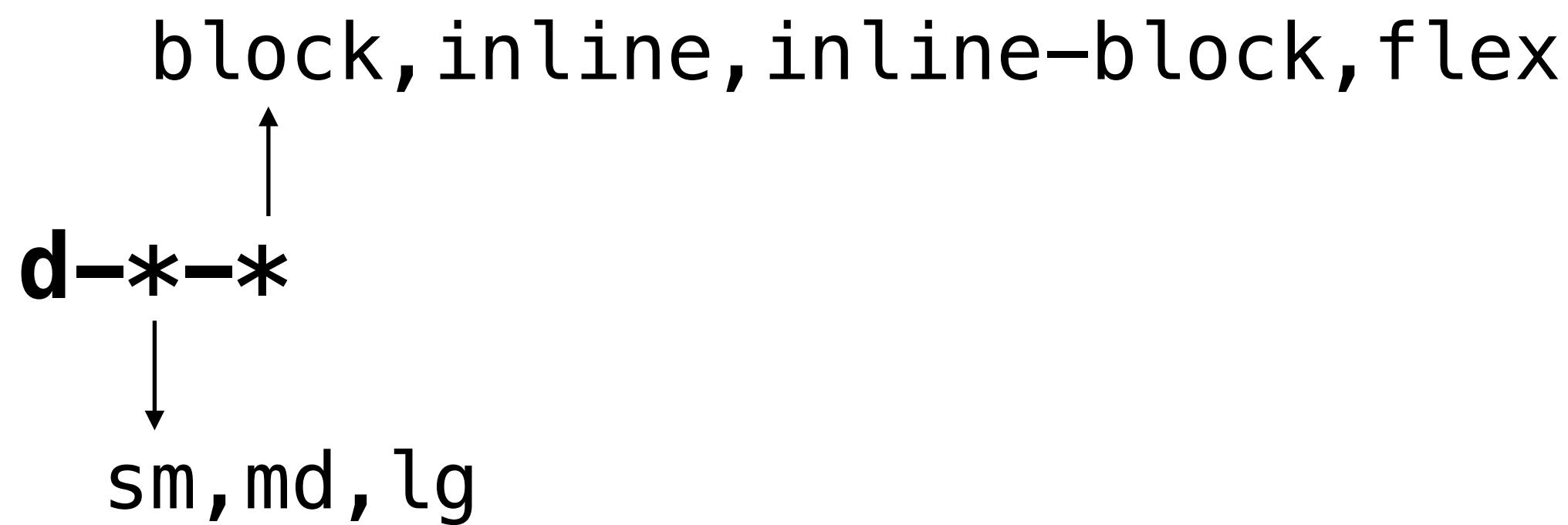
```
<div class="row">
  <div class="col-*-*">...</div>
  <div class="...">...</div>
  <div class="...">...</div>
</div>
<div class="row">
  ...
</div>
```



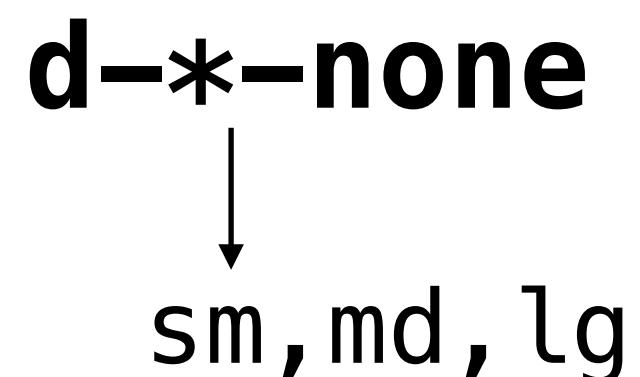
- If more than 12 columns are placed within a single row, they will wrap into a new line

# Responsive classes

- Change display property based on device
- Visible only on certain devices



- Hidden on certain devices (visible on all other)

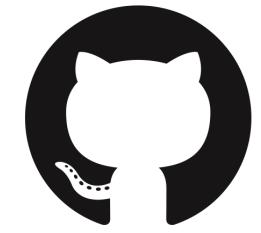


# Example (responsive classes)

# examples/bootstrap/grid\_rc.html



# **Exercise #2b**

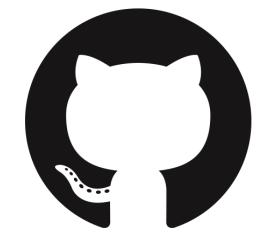


[https://github.com/dat310-2024/info/tree/master/  
exercises/bootstrap](https://github.com/dat310-2024/info/tree/master/exercises/bootstrap)

# Spacing

- Set **margin** and **padding** with predefined classes
  - sizes: 0, 1, 2, 3, 4, or 5 (and auto for margin)
    - p-0 → padding: 0;
    - m-1 → margin: 0.25rem;
  - sides: **t** top, **b** bottom, **s** left (start), **e** right (end),  
**x** left&right, **y** top&bottom
    - py-2 → padding-left: 0.5rem; padding-right: 0.5rem;
    - mt-3 → margin-top: 1rem;
  - add breakpoints
    - p-sm-1 → padding on small and larger screens
    - my-md-2 → margin 2 on medium and larger screens
- Use gutter **gy-2** to add horizontal spacing between columns

# **Exercise #2c**



[https://github.com/dat310-2023/info/tree/master/  
exercises/bootstrap](https://github.com/dat310-2023/info/tree/master/exercises/bootstrap)

# Style classes for HTML elements

- Basic text
- Tables
- Images
- Buttons
- Forms
- Helpers
- See <https://getbootstrap.com/docs/5.0>

# Text/Typography

- Elements are styled differently from browser defaults
  - Headings
  - Marks
  - Blockquotes
  - ...
- Contextual classes (for colors and backgrounds)
- See <https://getbootstrap.com/docs/5.0/content/typography/>

# Example (text)

examples/bootstrap/text.html

**Browser default**

## h1 heading secondary text

## h2 heading secondary text

## h3 heading secondary text

## h4 heading secondary text

## h5 heading secondary text

## h6 heading secondary text

### Highlight Text

Use the mark element to **highlight** text.

### Blockquotes

Theblockquoteelementisusedtopresentcontentfromanothersource:

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

From WWF's website

### Code Snippets

Example: Bootstrap text

## h1 heading secondary text

## h2 heading secondary text

## h3 heading secondary text

## h4 heading secondary text

## h5 heading secondary text

## h6 heading secondary text

### Highlight Text

Use the mark element to **highlight** text.

### Blockquotes

Theblockquoteelementisusedtopresentcontentfromanothersource:

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

From WWF's website

### Code Snippets

Inline snippets of code should be embedded in the code element:

The following HTML elements: `span`, `section`, and `div` defines a section in a document.

### Contextual Colors

Use the contextual classes to provide "meaning through colors":

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

### Contextual Backgrounds

Use the contextual background classes to provide "meaning through colors":

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

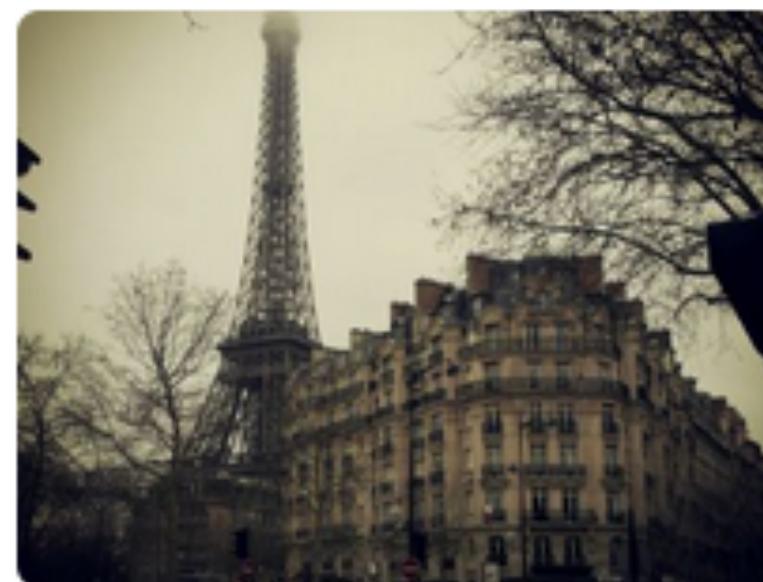
# Tables

- **.table** – basic table
- **.table-striped** – adds zebra-stripes
- **.table-bordered** – adds borders (on all sides of the table/cells)
- **.table-hover** – enables a hover state on table rows
- **.table-sm** – makes a table more compact
- For colors use **.table-active**, **.table-primary**, ... or **border-active**, **border-primary**, ...
- See <https://getbootstrap.com/docs/5.0/content/tables/>

# Images

- **.rounded** – adds rounded corners to image or other elements
- **.rounded-circle** – shapes the image to a circle
- **.img-thumbnail** – shapes the image to a thumbnail
- **.img-fluid** – automatically scale img to parent element
- See <https://getbootstrap.com/docs/5.0/content/images/>

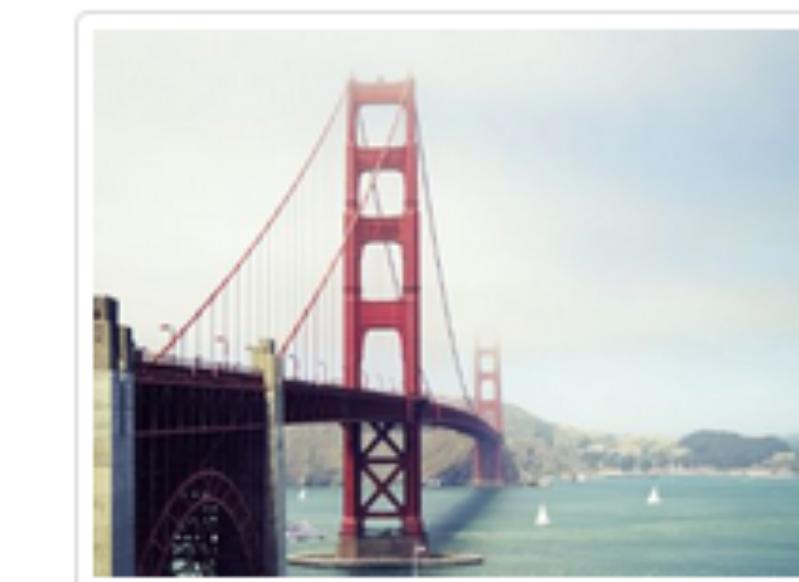
Rounded Corners:



Circle:

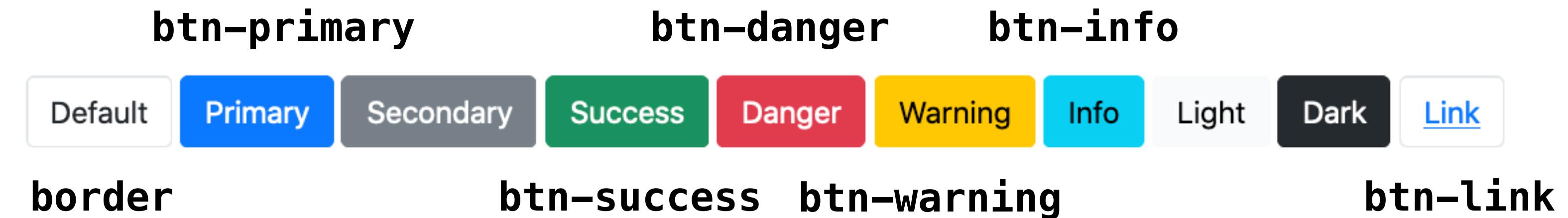


Thumbnail:

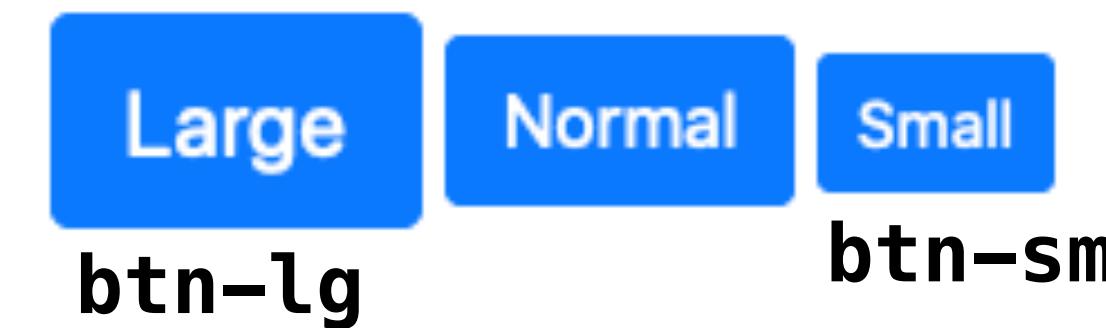


# Buttons

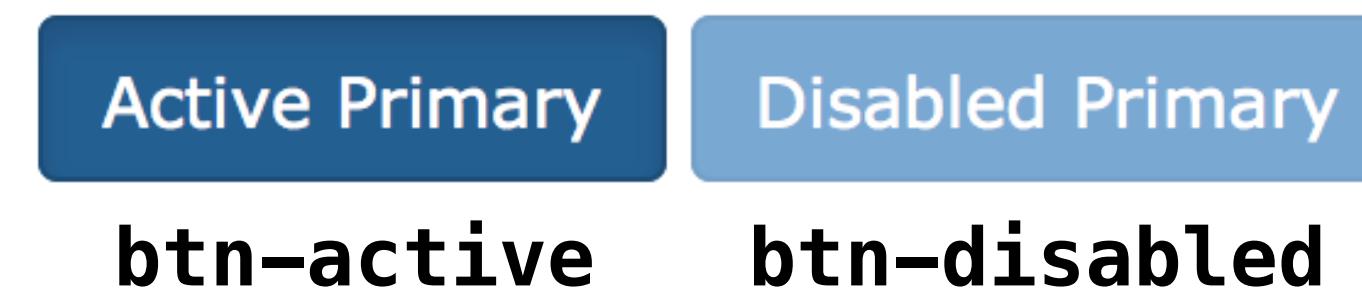
- Styles



- Sizes



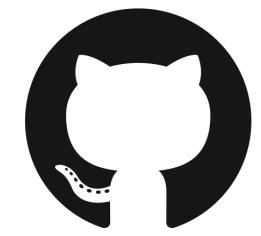
- States



- Button classes can be used on <a>, <button>, and <input> elements. E.g.,

```
<input type="submit" class="btn btn-info btn-md" value="Submit Button">
```

# Exercises #3



[https://github.com/dat310-2024/info/tree/master/  
exercises/bootstrap](https://github.com/dat310-2024/info/tree/master/exercises/bootstrap)

# Forms

- Individual form controls
- Inline vs. horizontal forms
- States (focused, disabled, readonly)
- Help text
- Validation states
- See <https://getbootstrap.com/docs/5.0/forms/overview/>

# Forms

```
<form>
  <label for="email" class="form-label">Email address</label>
  <input type="email" class="form-control" id="email">
  <div class="form-text">We'll never share your email.</div>
  <label for="password" class="form-label">Password</label>
  <input type="password" class="form-control" id="password">
  <div class="form-check">
    <input type="checkbox" class="form-check-input" id="check">
    <label class="form-check-label" for="check">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Email address

We'll never share your email with anyone else.

Password

Check me out

**Submit**

# Forms

Email address

We'll never share your email with anyone else.

Password

Check me out

**Submit**

```
<form>
  <label for="email" class="form-label">Email address</label>
  <input type="email" class="form-control" id="email">
  <div class="form-text">We'll never share your email.</div>
  <label for="password" class="form-label">Password</label>
  <input type="password" class="form-control" id="password">
  <div class="form-check">
    <input type="checkbox" class="form-control" checked="checked"/>
    <label class="form-check-label">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

All textual `<input>`, `<textarea>`, and `<select>` elements with **form-control** are set to **width: 100%**. Their `<label>`s with **form-label** also get **width: 100%**.

# Forms

```
<form>
  <label for="email" class="form-label">Email address</label>
  <input type="email" class="form-control" id="email">
  <div class="form-text">We'll never share your email.</div>
  <label for="password" class="form-label">Password</label>
  <input type="password" class="form-control" id="password"> Text explaining an input is formatted with form-text class.
  <div class="form-check">
    <input type="checkbox" class="form-check-input" id="check">
    <label class="form-check-label" for="check">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Email address

We'll never share your email with anyone else.

Password

Check me out

**Submit**

# Forms

```
<form>
  <label for="email" class="form-label">Email address</label>
  <input type="email" class="form-control" id="email">
  <div class="form-text">We'll never share your email.</div>
  <label for="password" class="form-label">Password</label>
  <input type="password" class="form-control" id="password">
  <div class="form-check">
    <input type="checkbox" class="form-check-input" id="check">
    <label class="form-check-label" for="check">Check me out!</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Email address

We'll never share your email with anyone else.

Password

Check me out!

**Submit**

Checkboxes and radio buttons should be put into a separate **<div>**.

# Forms

Email address

We'll never share your email with anyone else.

Password

Check me out

**Submit**

```
<form>
  <label for="email" class="form-label">Email address</label>
  <input type="email" class="form-control" id="email">
  <div class="form-text">We'll never share your email.</div>
  <label for="password" class="form-label">Password</label>
  <input type="password" class="form-control" id="password">
  <div class="form-check">
    <input type="checkbox" class="form-check-input" id="check">
    <label class="form-check-label" for="check">Check me out</label>
  </div>
  <button type="submit">Submit</button>
</form>
```

Add classes **form-check**, **form-check-input** and **form-check-label** to **<div>**, **<input>**, and **<label>** for checkbox and radio buttons

# nicer Forms

```
<form>
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1">
    <div class="form-text">We'll never share your email with anyone else.</div>
  </div>
  <div class="mb-3">
    <label for="password" class="form-label">Password</label>
    <input type="password" class="form-control" id="password">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Email address

We'll never share your email with anyone else.

Password

Check me out

**Submit**

Wrap label and input in a `<div>`, and add **bottom margin**.

# Horizontal Forms

```
<form>
  <div class="mb-3 row">
    <label for="exampleInputEmail1" class="form-label col-3">Email address</label>
    <div class="col-9">
      <input type="email" class="form-control" id="exampleInputEmail1">
      <div class="form-text">We'll never share your email with anyone else.</div>
    </div>
  </div>          Use the grid (row and col) to make horizontal forms
  <div class="mb-3">
    <label for="password" class="form-label">Password</label>
    <input type="password" class="form-control" id="password">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Email address

We'll never share your email with anyone else.

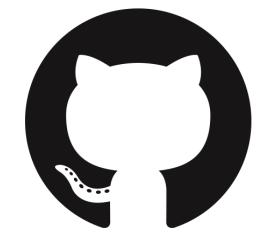
Password

Check me out

Submit

Use the grid (**row** and **col**) to make horizontal forms

# **Exercises #4-#5**



[https://github.com/dat310-2024/info/tree/master/  
exercises/bootstrap](https://github.com/dat310-2024/info/tree/master/exercises/bootstrap)

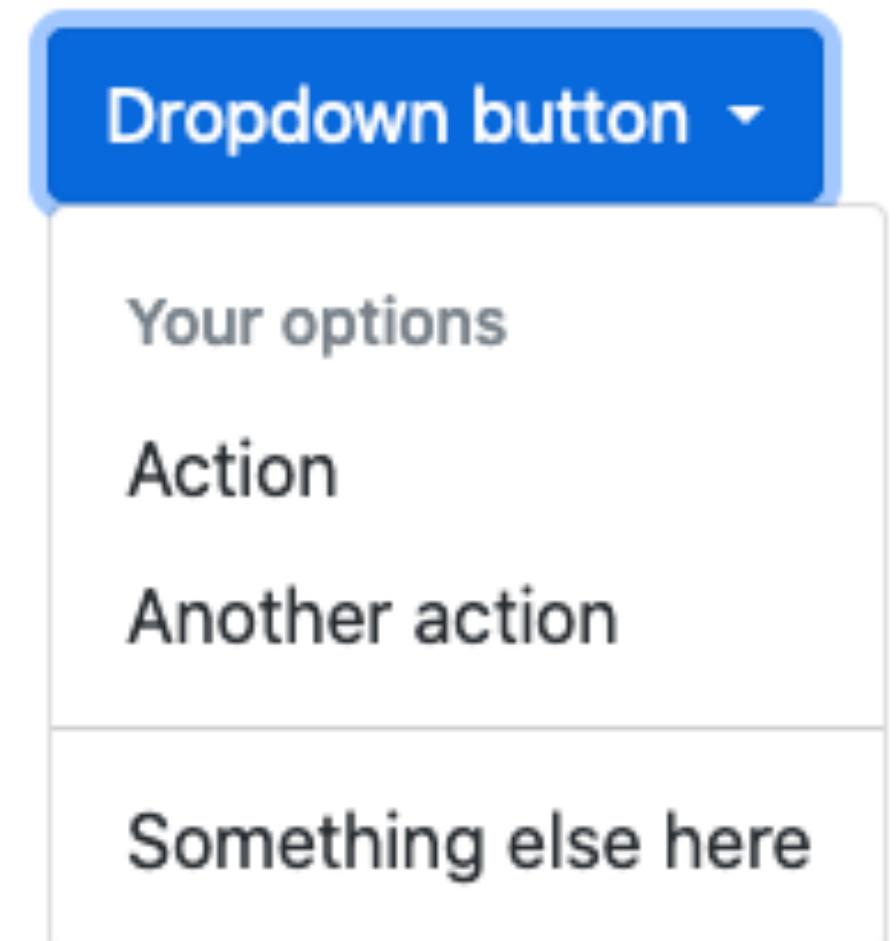
# **Bootstrap Components**

**Need to include JavaScript**

# Dropdowns

- Toggleable, contextual menu for displaying lists of links
  - Made interactive with JavaScript

```
<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button"
    data-bs-toggle="dropdown">
    Dropdown button
  </button>
  <ul class="dropdown-menu">
    <li><h6 class="dropdown-header">Your options</h6></li>
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
  </ul>
</div>
```



# Button groups

- Group a series of buttons together on a single line

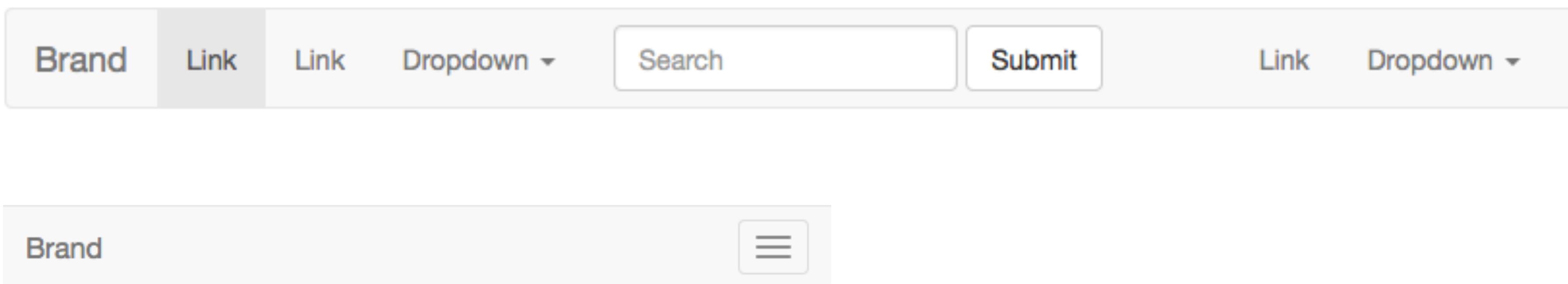
```
<div class="btn-group" role="group" aria-label="Basic example">
  <button type="button" class="btn btn-primary">Left</button>
  <button type="button" class="btn btn-primary">Middle</button>
  <button type="button" class="btn btn-primary">Right</button>
</div>
```



Left    Middle    Right

# Navbar

- Navigation header for the application or site
  - Collapsed in mobile views and become horizontal as the viewport increases
  - Requires JS



# Badges

- Labels in rounded boxes
  - Size is adjusted automatically

```
<h3>Example heading <span class="badge bg-secondary">New</span></h3>
```



```
<a href="#">Inbox <span class="badge bg-secondary">42</span></a>
```



```
<button class="btn btn-primary" type="button">  
  Messages <span class="badge bg-light text-dark">4</span>  
</button>
```



# Badges rounded

- Labels in rounded boxes
  - Size is adjusted automatically

```
<h3>Example heading <span class="badge bg-secondary rounded-pill">New</span></h3>
```

Example Heading 

```
<a href="#">Inbox <span class="badge bg-secondary rounded-pill">42</span></a>
```

[Inbox](#) 

```
<button class="btn btn-primary" type="button">  
  Messages <span class="badge bg-light text-dark rounded-pill">4</span>  
</button>
```

 Messages 4

# Alerts

- Contextual feedback messages for typical user actions

```
<div class="alert alert-warning" role="alert">
  <strong>Warning!</strong> Check the <a href="https://getbootstrap.com/docs/5.0/components/alerts" class="alert-link">documentation</a>.
</div>
```

**Warning:** Check the [documentation](#).

# Dismissible alerts

- Make any alert "closeable"
  - Needs JS

```
<div class="alert alert-warning alert-dismissible" role="alert">
  <button type="button" class="btn-close" data-dismiss="alert"><span>&times;</span></button>
  <strong>Holy guacamole!</strong> Check out all the options!</div>
```

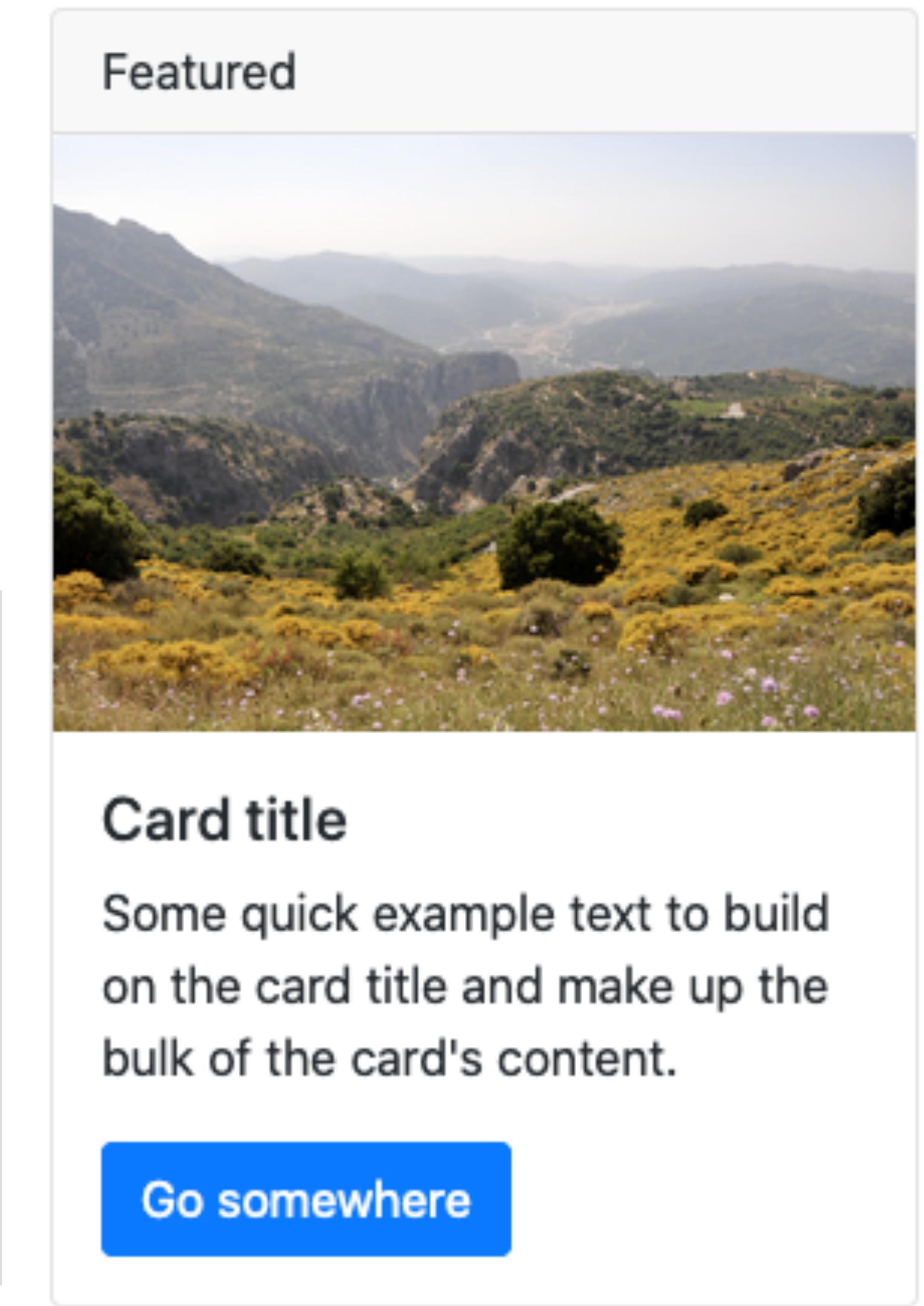
Holy guacamole! Check out all the options!



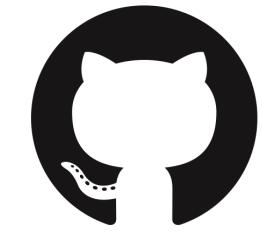
# Cards

- For "boxing" content

```
<div class="card" style="width: 18rem;">
  <div class="card-header">
    Featured
  </div>
  <img src=..." class="card-img-top" alt="...">
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on
make up the bulk of the card's content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```



# **Exercise #6-#7**



[https://github.com/dat310-2024/info/tree/master/  
exercises/bootstrap](https://github.com/dat310-2024/info/tree/master/exercises/bootstrap)

# Example

examples/bootstrap/datatables.html

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008/11/28	\$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09	\$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12	\$86,000
Bradley Greer	Software Engineer	London	41	2012/10/13	\$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07	\$206,850
Brielle Williamson	Integration Specialist	New York	61	2012/12/02	\$372,000
Bruno Nash	Software Engineer	London	38	2011/05/03	\$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011/12/12	\$106,450
Cara Stevens	Sales Assistant	New York	46	2011/12/06	\$145,600
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	2012/03/29	\$433,060

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008/11/28	\$162,700

Showing 1 to 10 of 57 entries

Previous [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) Next

# Server-side processing

- Needed if working with large databases
- All paging, searching, ordering operations are handed off to the server (where an SQL engine performs them)
- AJAX is used to get the required data from the server
- Details: <https://datatables.net/manual/server-side>

# References

- Bootstrap
  - <http://getbootstrap.com>
  - <http://www.w3schools.com/bootstrap>
- Lists of CSS boilerplates and frameworks
  - <http://mashable.com/2013/04/26/css-boilerplates-frameworks/>
  - <http://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html>