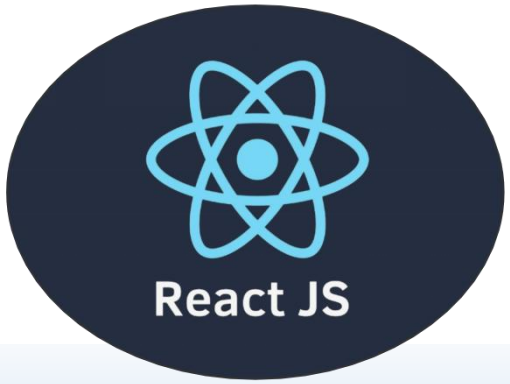
FPT Education
FPT UNIVERSITY



Introduction to Bootstrap Layout và JavaScript Components

Objectives

- Identify the purpose of using front-end UI frameworks in web design and development
- Set up and Configure project with Bootstrap
- Use the Bootstrap grid system to design responsive websites
- Recognize how the JavaScript components provide dynamic behavior
- Create various navigational elements using Tabs and Accordion
- Create modals, tooltips and popovers to reveal content in your website
- Build a carousel to show dynamic content on your website

FPT Education
FPT UNIVERSITY

Front-End Web UI Frameworks

What are front-end UI frameworks?

- Collection of ready-to-use HTML, CSS and JavaScript templates for UI components:
  - Typography, Forms, Buttons, Tables, Navigations, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel etc.

Popular front-end UI frameworks

1. Bootstrap
2. Semantic-UI
3. Foundation
4. Materialize
5. Material UI
6. Pure
7. Skeleton
8. UIKit
9. Milligram
10. Susy

Why Front-End Web UI Frameworks?

- Responsive web design
  - Mobile first
- Cross-browser compatibility
  - Dealing with quirks of browsers
- Increased productivity
  - Easy to get started
- Community support
  - Resources and web page templates

FPT Education
FPT UNIVERSITY

Introduction to Bootstrap

Bootstrap Overview

- Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web

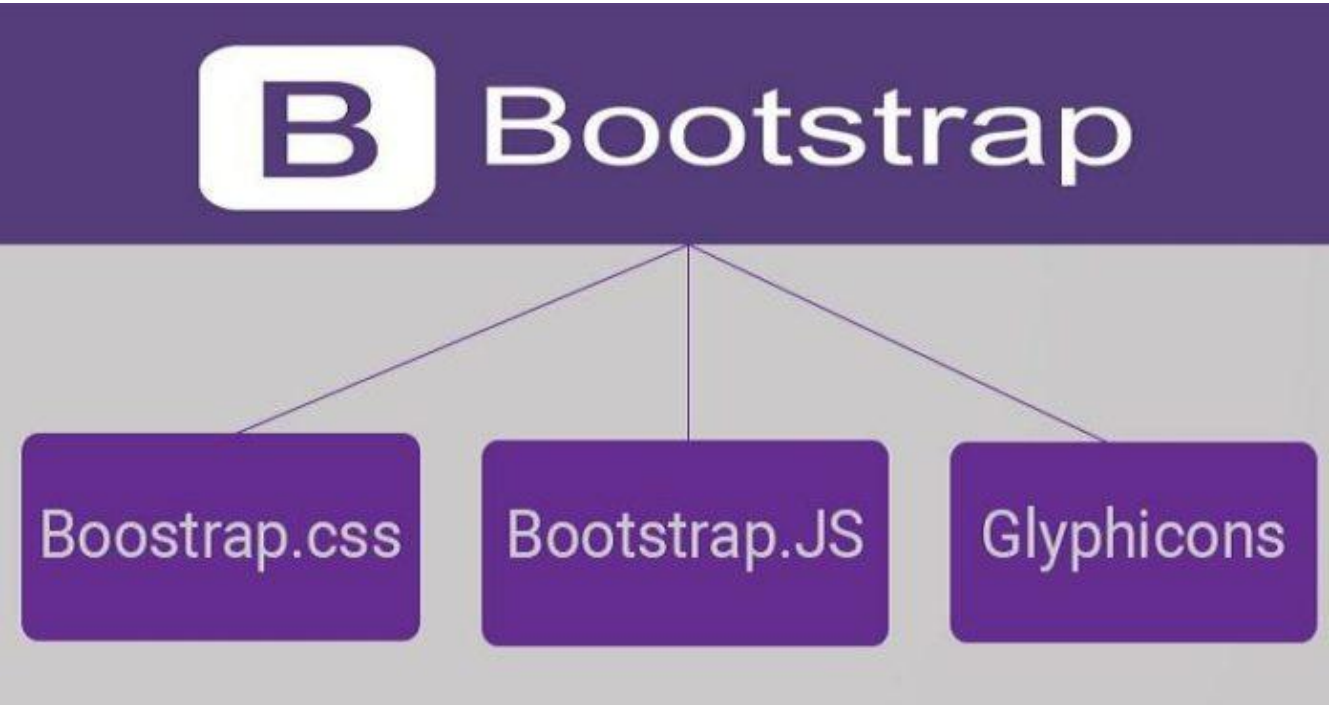
From the Bootstrap webpage

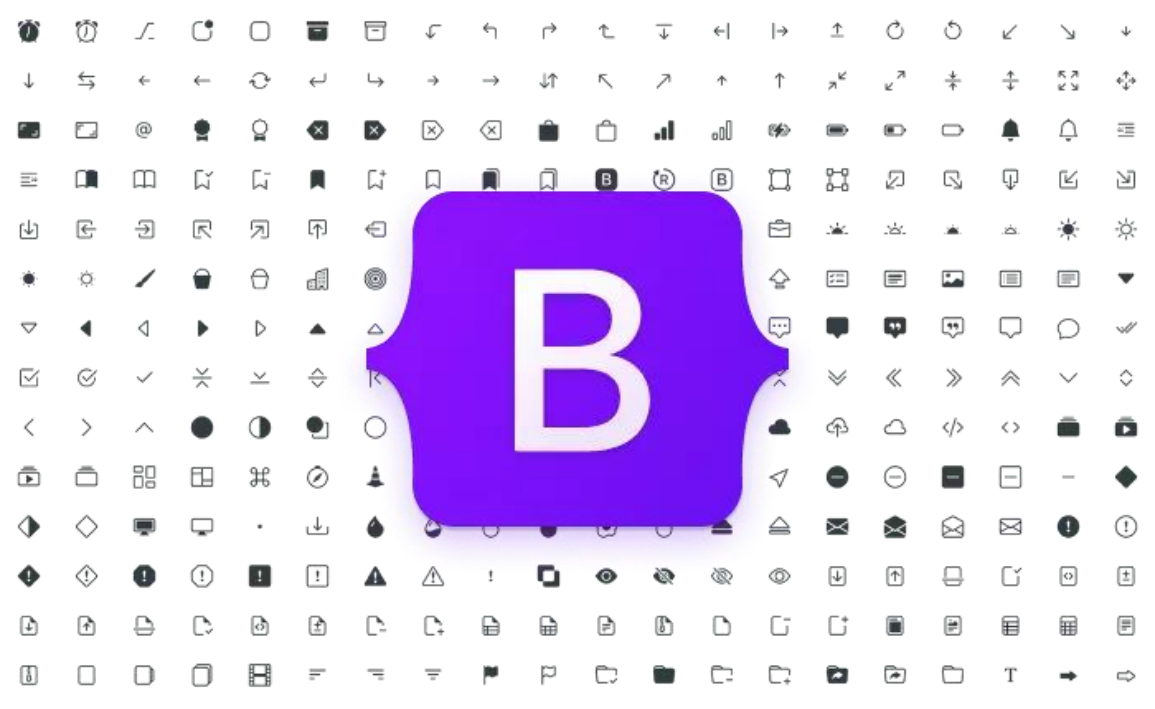
- Front-end framework for faster and easier web development
- Includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Easily create responsive designs with mobile first approach

Bootstrap History

- First released in 2011
  - Mark Otto and Jacob Thornton
- Current Production Version 5.3
- The first comprehensive framework
  - Gained popularity very quickly

Bootstrap files





Viewport

- <meta name="viewport" content="width=device-width, initial-scale=1">
- The *viewport* meta tag:
  - Ensures that the screen width is set to the device width and the content is rendered with this width in mind
  - Designing the websites to be responsive to the size of the viewport
    - Bootstrap grid system

- https://getbootstrap.com/docs/5.3/layout/grid/

Using BootStrap by CDN

1. Create a new index.html file in your project root.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

2. Include Bootstrap's CSS and JS

- CDN links: Place the <link> tag in the <head> for our CSS, and the <script> tag for our JavaScript bundle
  - <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
  - <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"></script>
- Including Popper for positioning dropdowns, poppers, and tooltips
  - <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"></script>
  - <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.min.js"></script>

https://getbootstrap.com/docs/5.3/getting-started/introduction/

Using BootStrap by Terminal

1. Create a folder to hold the files.
2. Initialize a new npm project:

   Run the command "**npm init -y**" to initialize.
3. Install Bootstrap package.

```
npm install bootstrap@5.3.1
```

4. Create an HTML file: index.html
5. Inside the index.html file, add the following code to include the Bootstrap CSS and JavaScript files

```
index.html > html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.min.css">
</head>
<body>

    <script src="node_modules/jquery/dist/jquery.slim.min.js"></script>
    <script src="node_modules/popper.js/dist/umd/popper.min.js"></script>
    <script src="node_modules/bootstrap/dist/js/bootstrap.min.js"></script>

</body>

</html>
```

6. Start a local development server to view your project:

   npx lite-server

```
package.json > dependencies
  "description" : "",
  "main": "index.js",
  Debug
  "scripts": {
    "start": "npm run lite",
    "test": "echo \"Error: no test specified\" && exit 1",
    "lite": "lite-server"
  },
```

FPT Education
FPT UNIVERSITY

Exercise 5: Getting Started with Bootstrap

Bootstrap Grid

- Designed to be:
  - Responsive
  - Mobile first
  - Fluid

CSS Flexbox Layout

- Simpler and flexible layout options in CSS
- Can easily handle dynamic/unknown size of content containers
- Direction-agnostic layout

Why Flexbox for Bootstrap?

- Easy vertical alignment of content within a parent element
- Easy reordering of content across devices and screen resolutions with the help of media queries
- Easy CSS-only equal height columns for your grid-based layouts

FPT Education
FPT UNIVERSITY

Bootstrap Container

```
<div class="container-sm">100% wide until small breakpoint</div>
<div class="container-md">100% wide until medium breakpoint</div>
<div class="container-lg">100% wide until large breakpoint</div>
<div class="container-xl">100% wide until extra large breakpoint</div>
<div class="container-xxl">100% wide until extra extra large breakpoint</div>
```

- Bootstrap comes with three different containers:
  - **.container**, which sets a max-width at each responsive breakpoint
  - **.container-{breakpoint}**, which is width: 100% until the specified breakpoint
  - **.container-fluid**, which is width: 100% at all breakpoints

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
.container	100%	540px	720px	960px	1140px	1320px
.container-sm	100%	540px	720px	960px	1140px	1320px
.container-md	100%	100%	720px	960px	1140px	1320px
.container-lg	100%	100%	100%	960px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

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

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

FPT Education
FPT UNIVERSITY

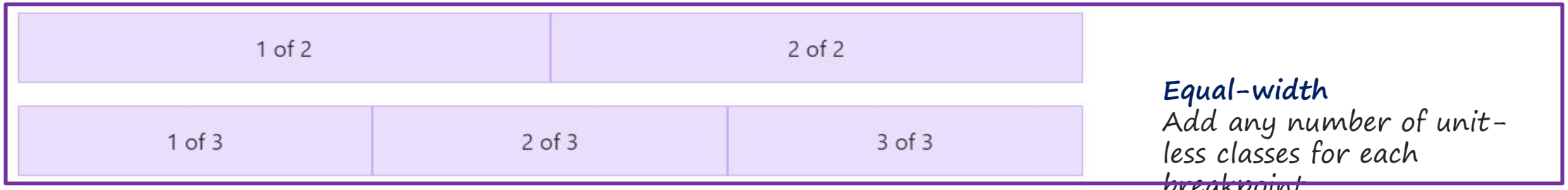
Boostrap Grid

Bootstrap Grid

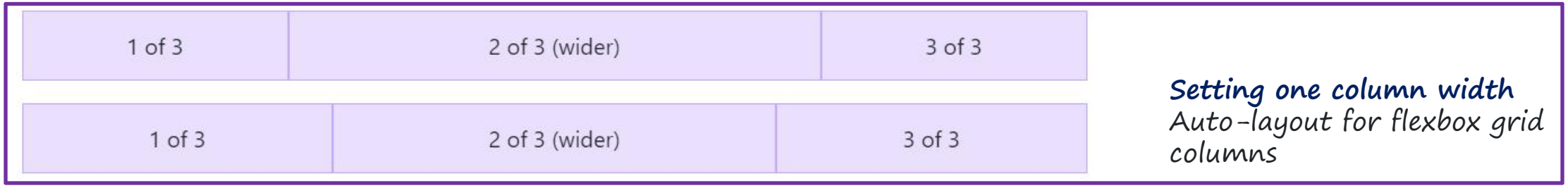
- Bootstrap's grid system can adapt across all six default breakpoints, and any breakpoints you customize. The six default grid tiers are as follows:
  - Extra small (xs)
  - Small (sm)
  - Medium (md)
  - Large (lg)
  - Extra large (xl)
  - Extra extra large (xxl)

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container `max-width`	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`	`.col-xxl-`

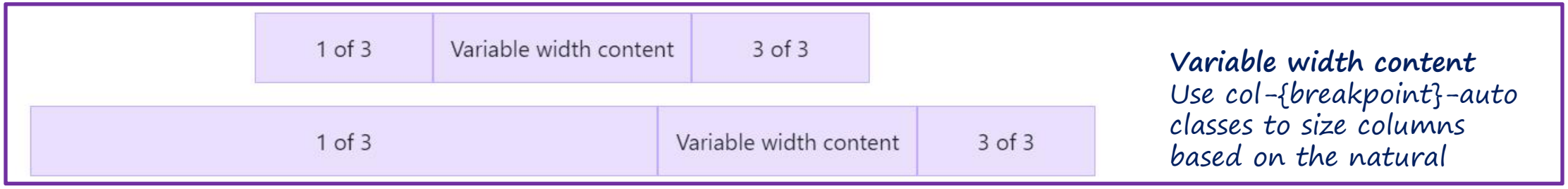
Auto-layout columns



Equal-width
Add any number of unit-less classes for each breakpoint



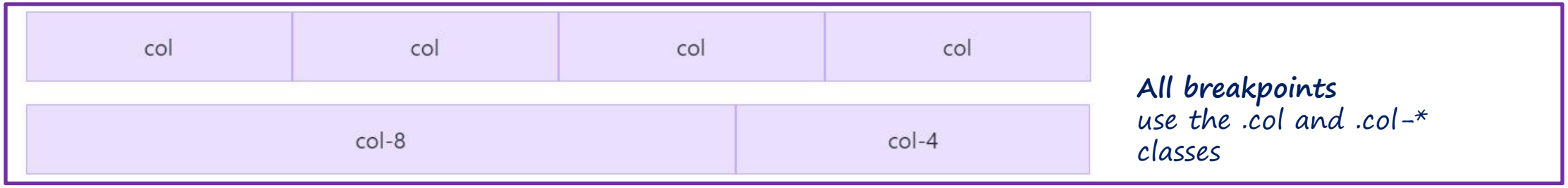
Setting one column width
Auto-layout for flexbox grid columns



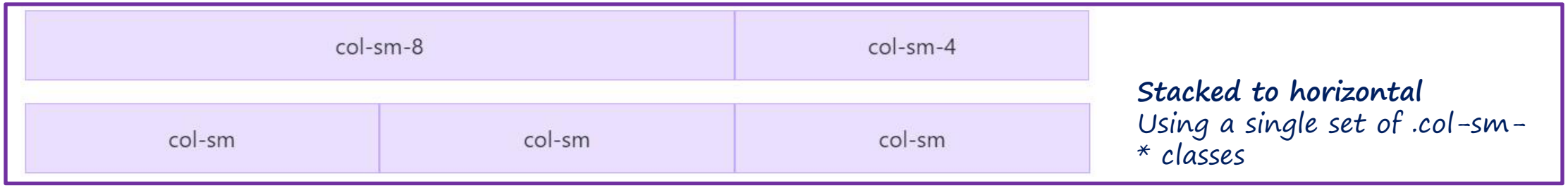
Variable width content
Use col-{breakpoint}-auto classes to size columns based on the natural

https://getbootstrap.com/docs/5.3/layout/grid/#auto-layout-columns

Responsive classes



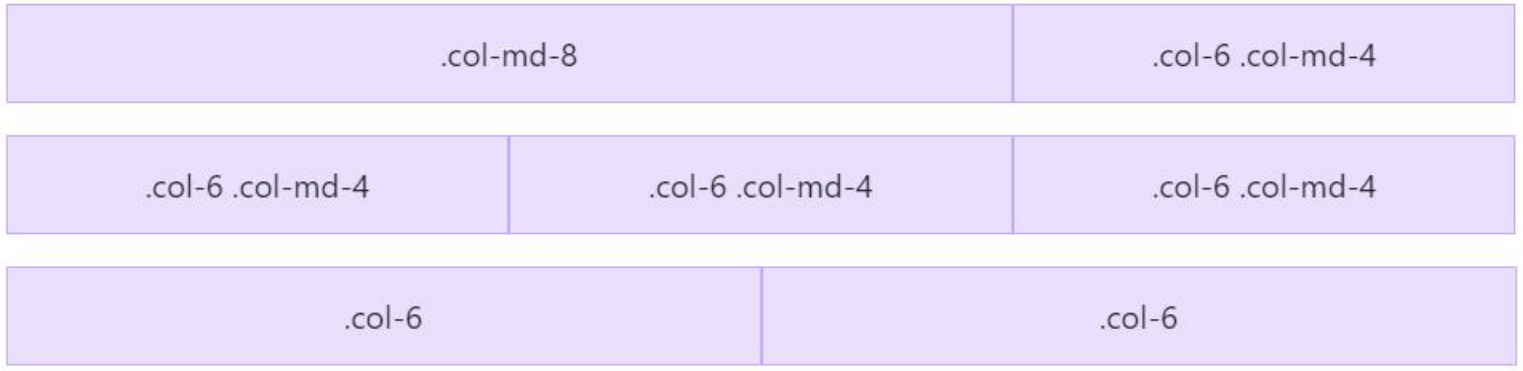
All breakpoints
use the .col and .col-* classes



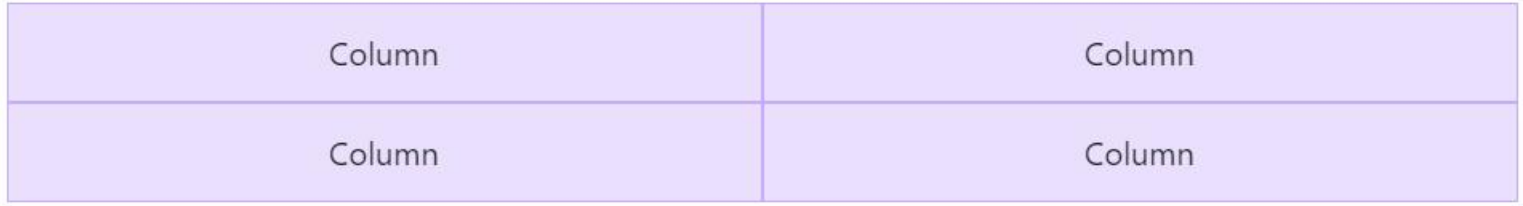
Stacked to horizontal
Using a single set of .col-sm-* classes

https://getbootstrap.com/docs/5.3/layout/grid/#responsive-classes

Responsive classes – cont'd



Mix and match
Use a combination of different classes for each tier as needed

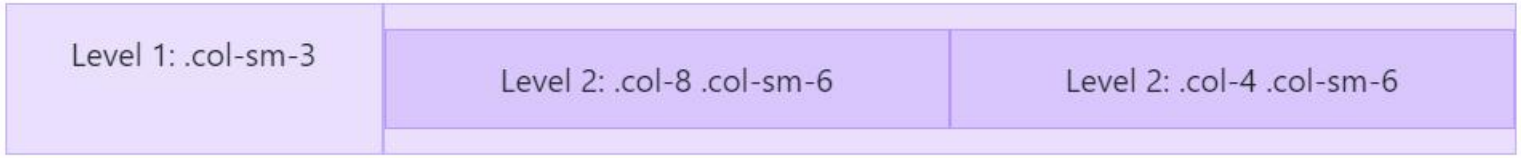


Row columns
Use the responsive .row-cols-*

https://getbootstrap.com/docs/5.3/layout/grid/#responsive-classes

Nesting column

- To nest content with the default grid, add a new .row and set of .col-sm-* columns within an existing .col-sm-* column.

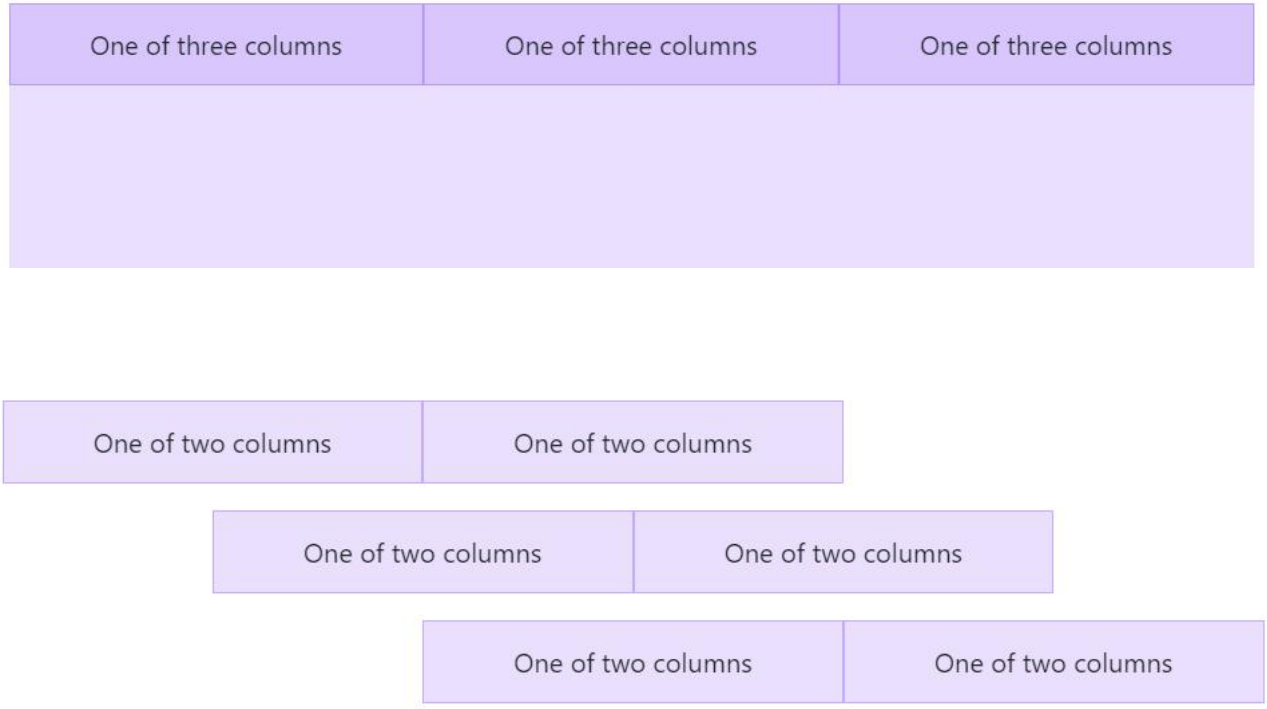



```
<div class="container text-center">
  <div class="row">
    <div class="col-sm-3">
      Level 1: .col-sm-3
    </div>
    <div class="col-sm-9">
      <div class="row">
        <div class="col-8 col-sm-6">
          Level 2: .col-8 .col-sm-6
        </div>
        <div class="col-4 col-sm-6">
          Level 2: .col-4 .col-sm-6
        </div>
      </div>
    </div>
  </div>
</div>
```

FPT Education
FPT UNIVERSITY

Bootstrap Column

Flexbox Alignment



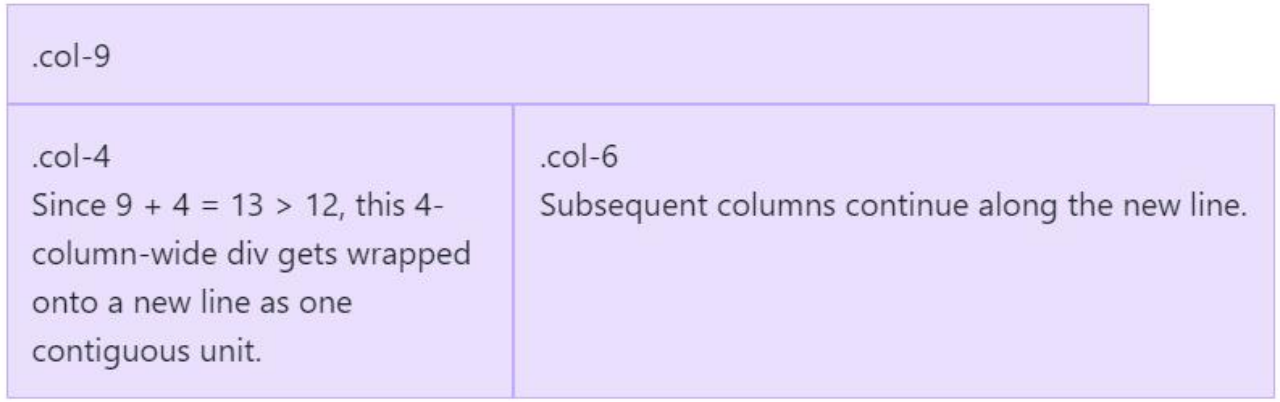
Vertical alignment

Change the vertical alignment with any of the responsive align-items- classes*

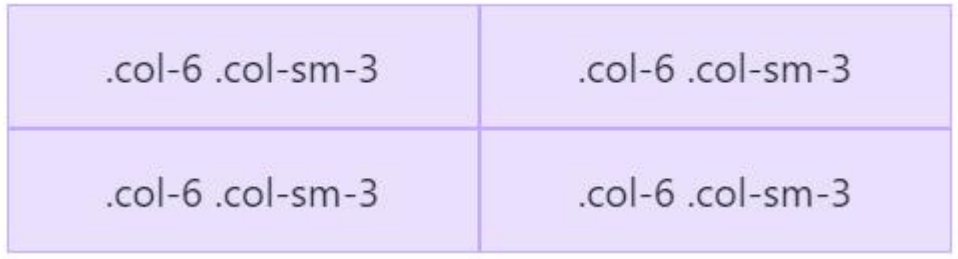
Horizontal alignment

Change the horizontal alignment with any of the responsive justify-content-* classes.

Flexbox Alignment – cont'd



Column wrapping
If more than 12 columns are placed within a single row, each group of extra columns will, as one unit, wrap onto a new line



Column breaks
Breaking columns to a new line in flexbox requires a small hack: add an element with width: 100% wherever you want to wrap your columns to a new line

Reordering

First in DOM, no order applied	Third in DOM, with an order of 1	Second in DOM, with a larger order

```
<div class="container text-center">
  <div class="row">
    <div class="col order-last">
      First in DOM, ordered last
    </div>
    <div class="col">
      Second in DOM, unordered
    </div>
    <div class="col order-first">
      Third in DOM, ordered first
    </div>
  </div>
</div>
```

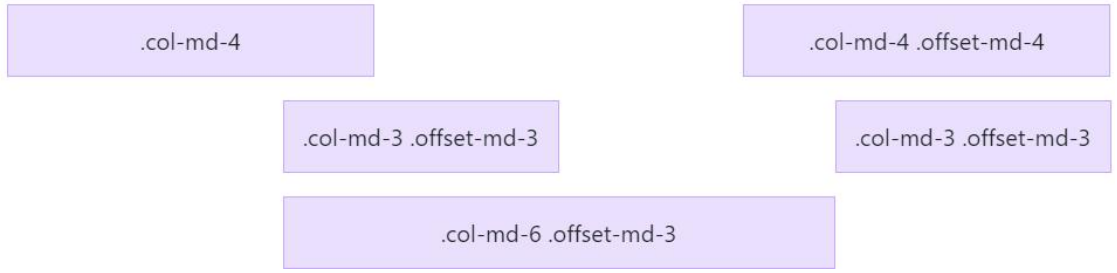
Order classes

- Use .order- classes for controlling the visual order of your content. These classes are responsive, so you can set the order by breakpoint (e.g., .order-1.order-md-2).

```
<div class="container text-center">
  <div class="row">
    <div class="col">
      First in DOM, no order applied
    </div>
    <div class="col order-5">
      Second in DOM, with a larger order
    </div>
    <div class="col order-1">
      Third in DOM, with an order of 1
    </div>
  </div>
</div>
```

Third in DOM, ordered first	Second in DOM, unordered	First in DOM, ordered last

Reordering – cont'd



Offset classes

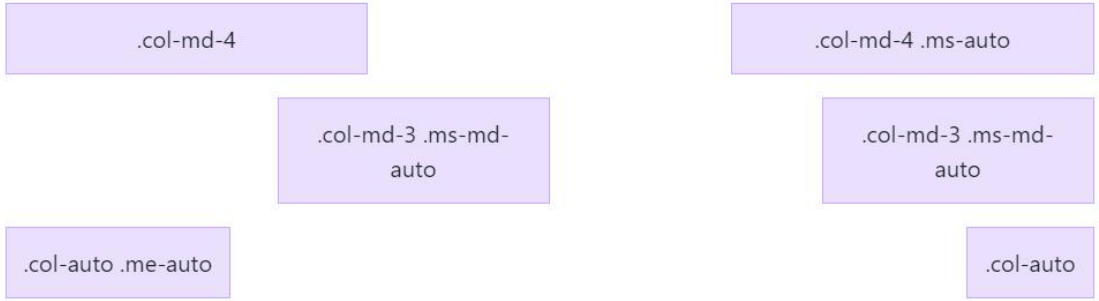
- Move columns to the right using .offset-md-* classes. These classes increase the left margin of a column by * columns.

```
<div class="container text-center">
  <div class="row">
    <div class="col-sm-5 col-md-6">.col-sm-5 .col-md-6</div>
    <div class="col-sm-5 offset-sm-2 col-md-6 offset-md-0">.col-sm-5 .offset-sm-2
  </div>
  <div class="row">
    <div class="col-sm-6 col-md-5 col-lg-6">.col-sm-6 .col-md-5 .col-lg-6</div>
    <div class="col-sm-6 col-md-5 offset-md-2 col-lg-6 offset-lg-0">.col-sm-6 .co
  </div>
</div>
```

```
<div class="container text-center">
  <div class="row">
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4 offset-md-4">.col-md-4 .offset-md-4</div>
  </div>
  <div class="row">
    <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
    <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
  </div>
  <div class="row">
    <div class="col-md-6 offset-md-3">.col-md-6 .offset-md-3</div>
  </div>
</div>
```

.col-sm-5 .col-md-6	.col-sm-5 .offset-sm-2 .col-md-6 .offset-md-0
.col-sm-6 .col-md-5 .col-lg-6	.col-sm-6 .col-md-5 .offset-md-2 .col-lg-6 .offset-lg-0

Reordering – cont'd




```
<div class="container text-center">
  <div class="row">
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4 ms-auto">.col-md-4 .ms-auto</div>
  </div>
  <div class="row">
    <div class="col-md-3 ms-md-auto">.col-md-3 .ms-md-auto</div>
    <div class="col-md-3 ms-md-auto">.col-md-3 .ms-md-auto</div>
  </div>
  <div class="row">
    <div class="col-auto me-auto">.col-auto .me-auto</div>
    <div class="col-auto">.col-auto</div>
  </div>
</div>
```

Margin utilities

- With the move to flexbox in v4, you can use margin utilities like .me-auto to force sibling columns away from one another.

```
<div class="col-3 p-3 mb-2">
  .col-3: width of 25%
</div>

<div class="col-sm-9 p-3">
  .col-sm-9: width of 75% above sm breakpoint
</div>
```

.col-3: width of 25%

.col-sm-9: width of 75% above sm breakpoint

FPT Education
FPT UNIVERSITY

Exercise 6: Build website layout using Bootstrap Grid

PETER VERKUILEN | *Design + Consulting*

Welcome | Designs | About

Peter Verkuilen *Design + Consulting* / Contacts: (917) 318-7924 or peter@peterverkuilen.com

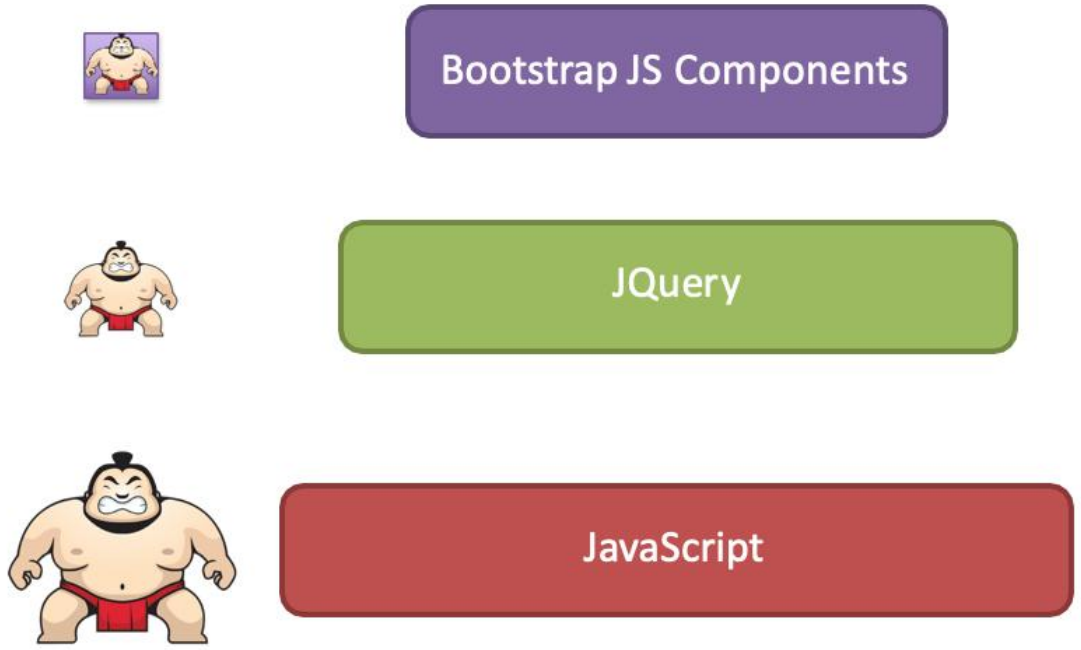
All content copyright © 2011 Peter Verkuilen | Top

FPT Education
FPT UNIVERSITY

Bootstrap JavaScript Components

Bootstrap and JavaScript

- Bootstrap's JavaScript support is through JS Plugins
  - Plugins written based on Jquery
  - Plugins can be individually included

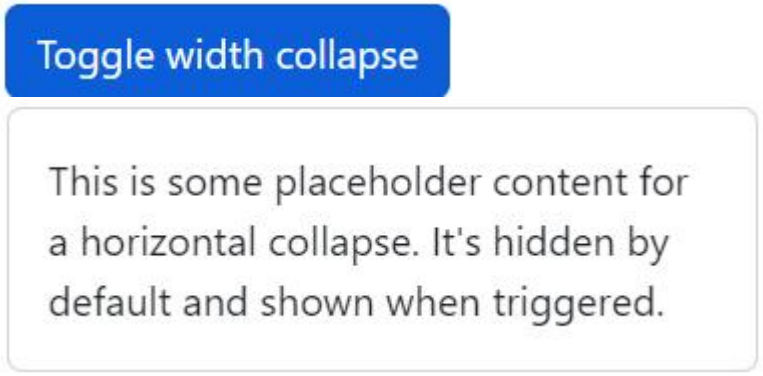


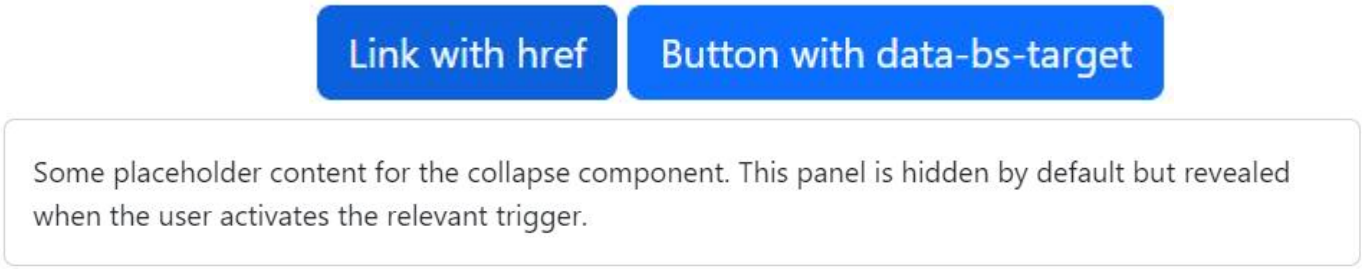
Bootstrap JS Components

- JS components can all be used without writing a single line of JavaScript:
  - data-* attributes (e.g., data-toggle, data-spy)
  - Straightforward approach to use plugins
  - We will explore this approach in this module
- Full JS API available if needed
  - Need to know JQuery syntax and JavaScript

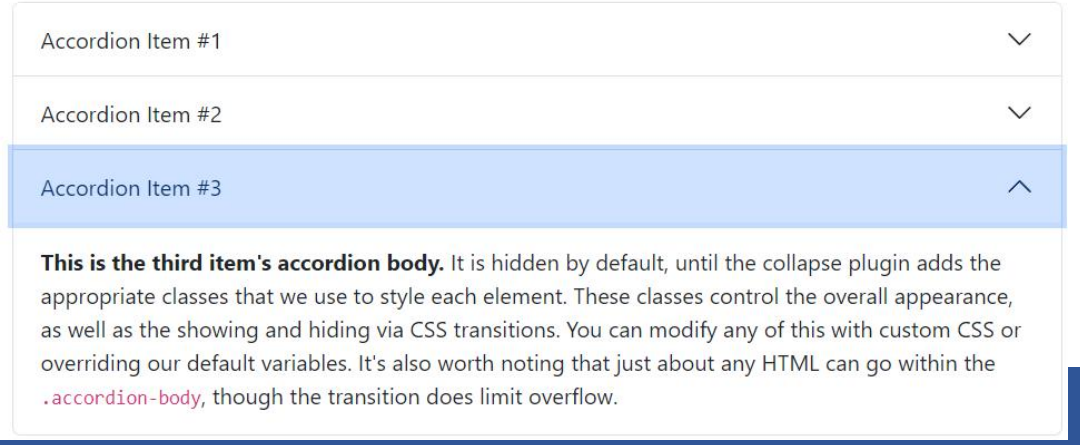
Collapse and Accordion

- **Collapse**: Toggle the visibility of content across your project with a few classes and our JavaScript plugins.





- **Accordion**: Build vertically collapsing accordions in combination with our Collapse JavaScript plugin.



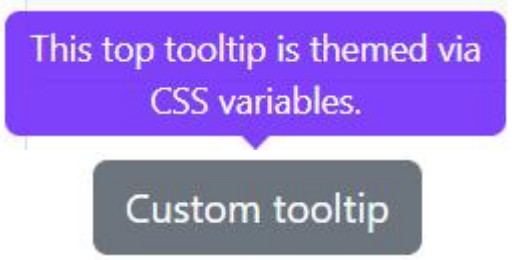
Tooltips, Popovers and Modals

- Revealing content upon interacting with an element on the web page
- Information displayed as an overlay
- Order of flexibility:
- Tooltip → Popover → Modal
- Modal:
  - More detailed information can be presented than tooltips and popovers.
  - Modal contains header, body and footer.
  - Can use the Bootstrap grid in the body to organize content.

FPT Education
FPT UNIVERSITY

Tooltips

- Custom tooltips




```
<button type="button" class="btn btn-secondary"
        data-bs-toggle="tooltip" data-bs-placement="top"
        data-bs-custom-class="custom-tooltip"
        data-bs-title="This top tooltip is themed via CSS variables.">
  Custom tooltip
</button>
```

```
const myTooltipEl = document.getElementById('myTooltip')
const tooltip = bootstrap.Tooltip.getOrCreateInstance(myTooltipEl)

myTooltipEl.addEventListener('hidden.bs.tooltip', () => {
  // do something...
})

tooltip.hide()
```

- Directions

https://getbootstrap.com/docs/5.3/components/tooltips/

Popovers

- Popovers are opt-in for performance reasons, so **you must initialize them yourself**.
- Zero-length title and content values will never show a popover.

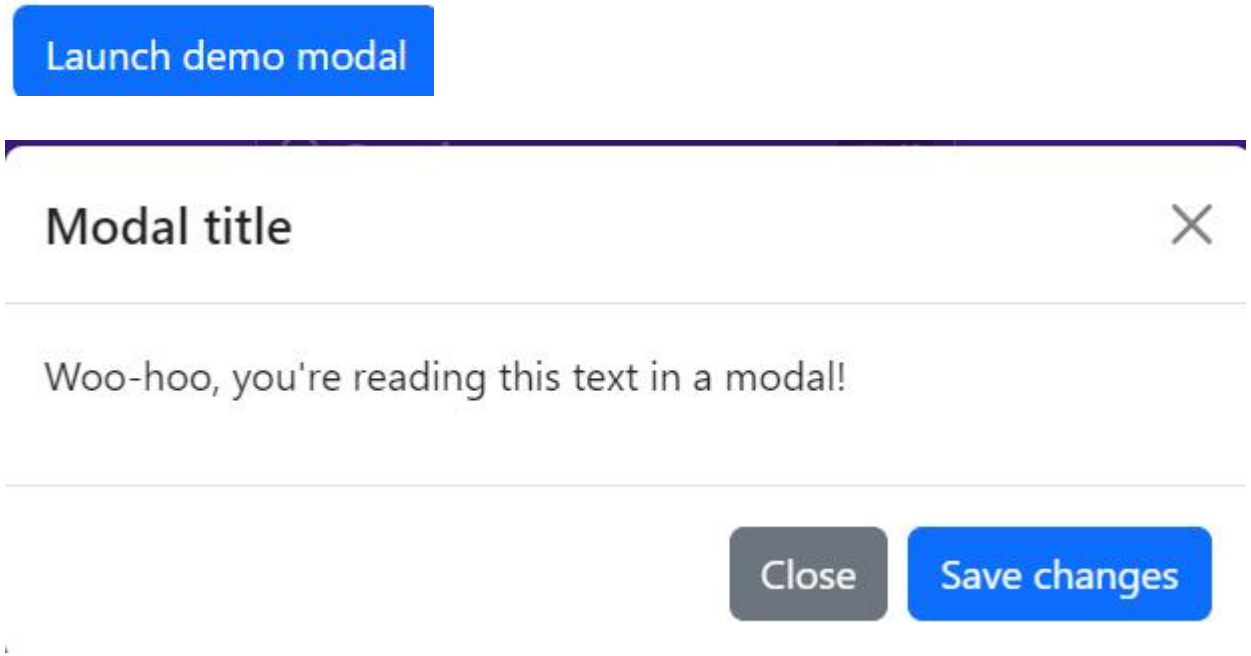

```
const exampleEl = document.getElementById('example')
const popover = new bootstrap.Popover(exampleEl, options)
```

```
<button type="button" class="btn btn-lg btn-danger" data-bs-
toggle="popover" data-bs-title="Popover title" data-bs-content="And
here's some amazing content. It's very engaging. Right?">
Click to toggle popover
</button>
```

https://getbootstrap.com/docs/5.3/components/popovers/

Modals

- Clicking on the modal “backdrop” will automatically close the modal.
- Bootstrap only supports one modal window at a time. Nested modals aren’t supported as we believe them to be poor user experiences.




```
const myModal = document.getElementById('myModal')
const myInput = document.getElementById('myInput')

myModal.addEventListener('shown.bs.modal', () => {
  myInput.focus()
})
```

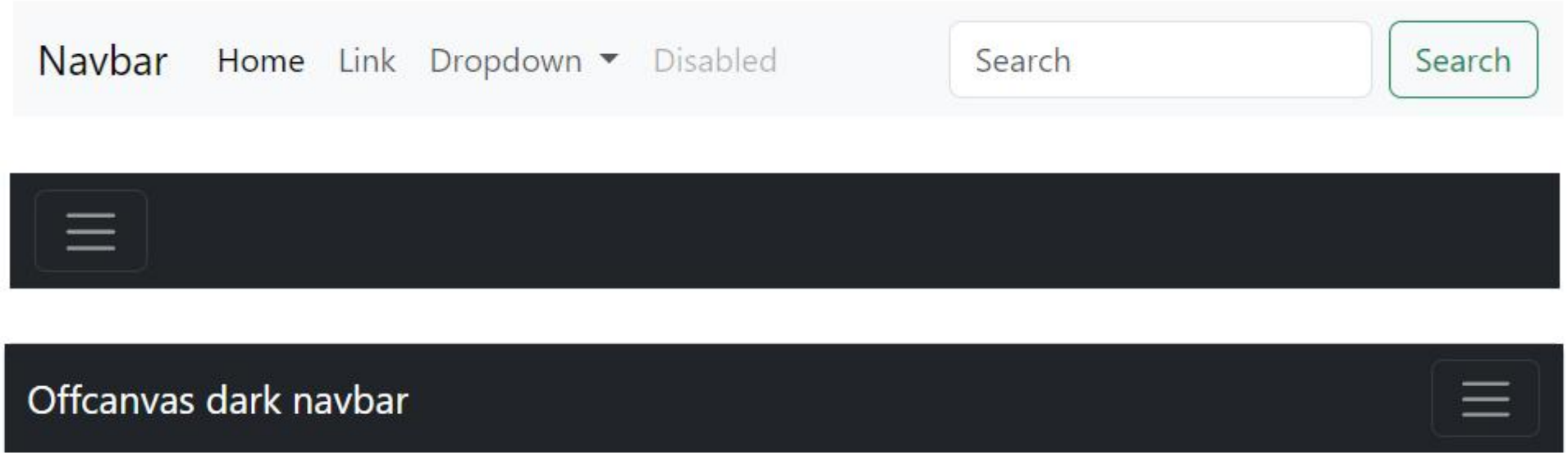
https://getbootstrap.com/docs/5.3/components/modal/

Navbar, Navs & Tabs

- Navbar:
  - A common UI component used to create a navigation menu at the top of a webpage.
- Navs (Navigation):
  - Navigation elements used to create a set of links or navigation items within a webpage. They can be horizontal or vertical and are often placed within a Navbar or a sidebar.
- Tabs:
  - Another type of navigation component that allows users to switch between different sections or views within a webpage, typically without having to navigate to a new page. Tabs are often displayed horizontally and consist of clickable labels or tabs that correspond to different content panels.

Navbar

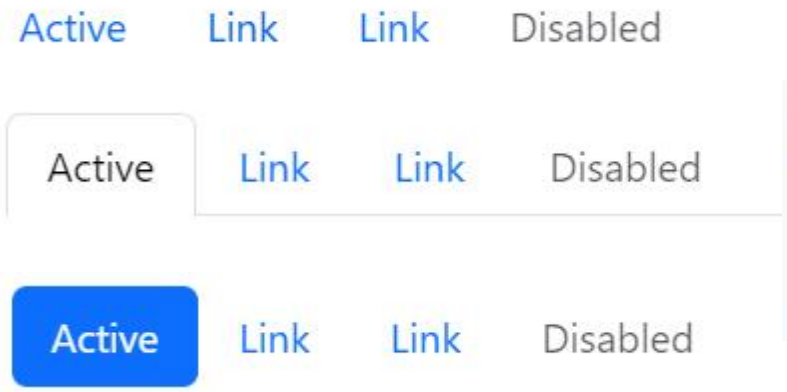
- Navbars are responsive by default, but you can easily modify them to change that. Responsive behavior depends on our Collapse JavaScript plugin.



https://getbootstrap.com/docs/5.3/components/navbar/

Navs & Tabs

- Navigation available in Bootstrap share general markup and styles, from the base .nav class to the active and disabled states. Swap modifier classes to switch between each style.



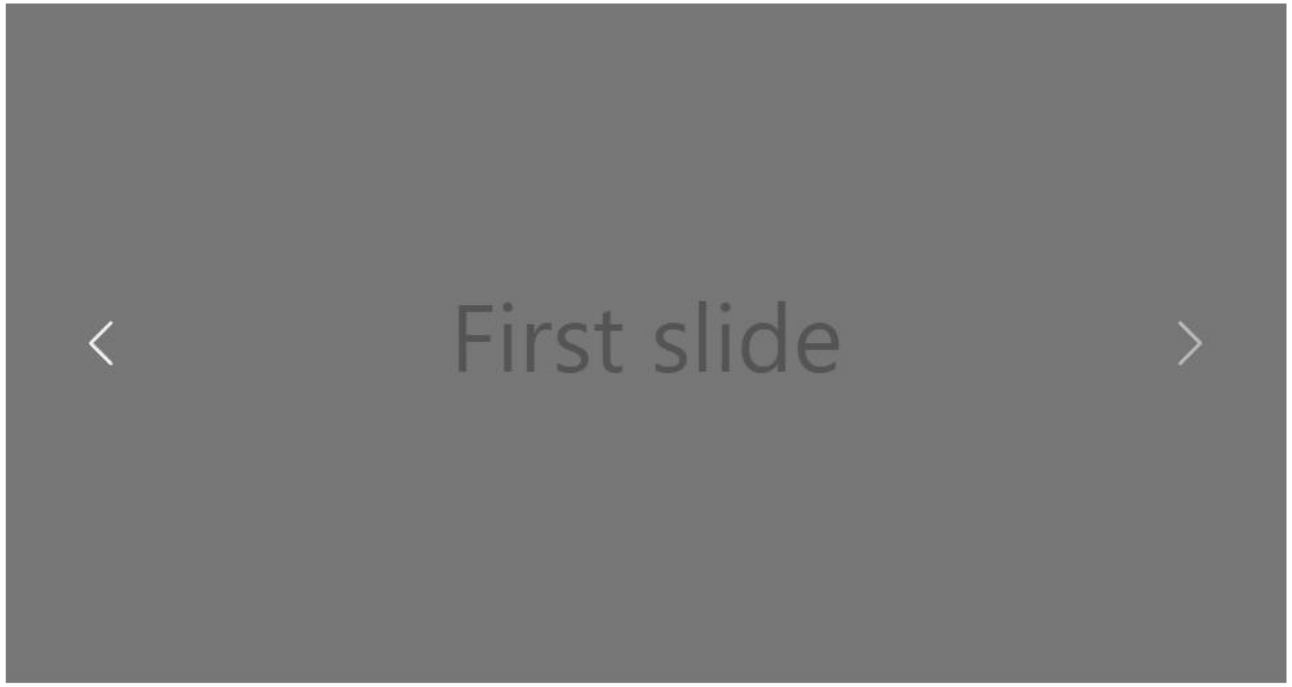

```
const tabEl = document.querySelector('button[data-bs-toggle="tab"]')
tabEl.addEventListener('shown.bs.tab', event => {
  event.target // newly activated tab
  event.relatedTarget // previous active tab
})
```

https://getbootstrap.com/docs/5.3/components/navs-tabs/

Carousel

- The carousel is a slideshow for cycling through a series of content, built with CSS 3D transforms and a bit of JavaScript. It works with a series of images, text, or custom markup. It also includes support for previous/next controls and indicators.
- **Carousels must be manually initialized** using the carousel constructor method. Without initialization, some of the event listeners (specifically, the events needed touch/swipe support) will not be registered until a user has explicitly activated a control or indicator.

Carousel – cont'd



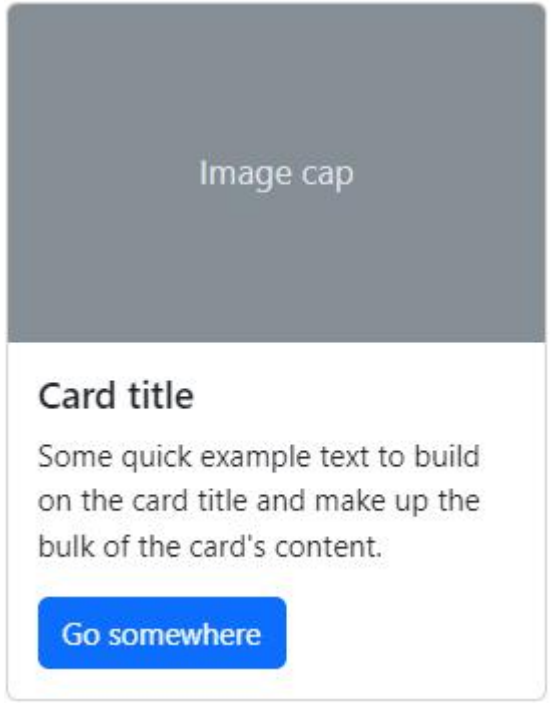

```
<div id="carouselExample" class="carousel slide">
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="..." class="d-block w-100" alt="...">
    </div>
    <div class="carousel-item">
      <img src="..." class="d-block w-100" alt="...">
    </div>
    <div class="carousel-item">
      <img src="..." class="d-block w-100" alt="...">
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-target="#carouselExample" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#carouselExample" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
```

https://getbootstrap.com/docs/5.3/components/carousel/

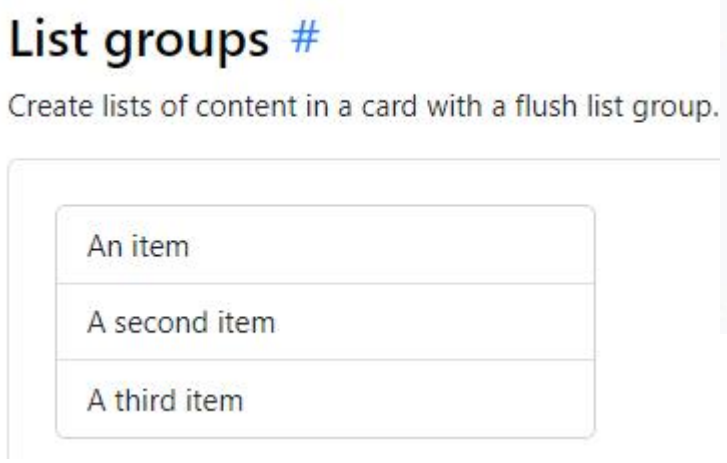
Cards

- A **card** is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options. If you're familiar with Bootstrap 3, cards replace our old panels, wells, and thumbnails. Similar functionality to those components is available as modifier classes for cards.

Cards – cont'd




```
<div class="card" style="width: 18rem;">
  <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 the card title and make up the bul
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```




```
<div class="card" style="width: 18rem;">
  <ul class="list-group list-group-flush">
    <li class="list-group-item">An item</li>
    <li class="list-group-item">A second item</li>
    <li class="list-group-item">A third item</li>
  </ul>
</div>
```

https://getbootstrap.com/docs/5.3/components/card/

FPT Education
FPT UNIVERSITY

Exercise 7: Demo about Cards Column

Cards Columns

Some text inside the first card

Some text inside the first card

Some text inside the first card

FPT Education
FPT UNIVERSITY

Bootstrap Forms

Bootstrap ‘s form controls

- Bootstrap’s form controls expand on our Rebooted form styles with classes. Use these classes to opt into their customized displays for a more consistent rendering across browsers and devices.

https://getbootstrap.com/docs/5.3/forms/overview/

Form control
Style textual inputs and textareas with support for multiple states.

Select
Improve browser default select elements with a custom initial appearance.

Checks & radios
Use our custom radio buttons and checkboxes in forms for selecting input options.

Range
Replace browser default range inputs with our custom version.

Input group
Attach labels and buttons to your inputs for increased semantic value.

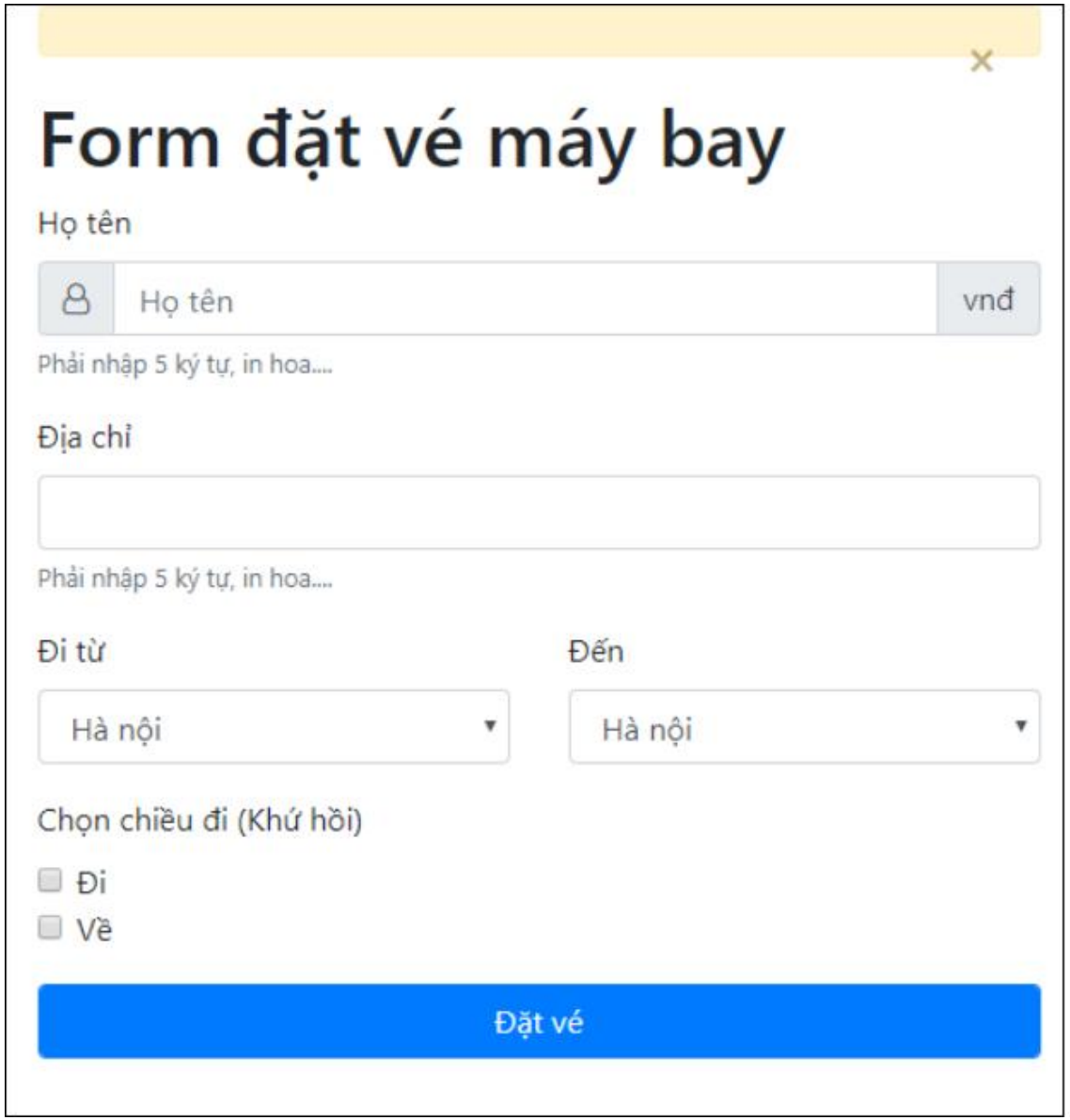
Floating labels
Create beautifully simple form labels that float over your input fields.

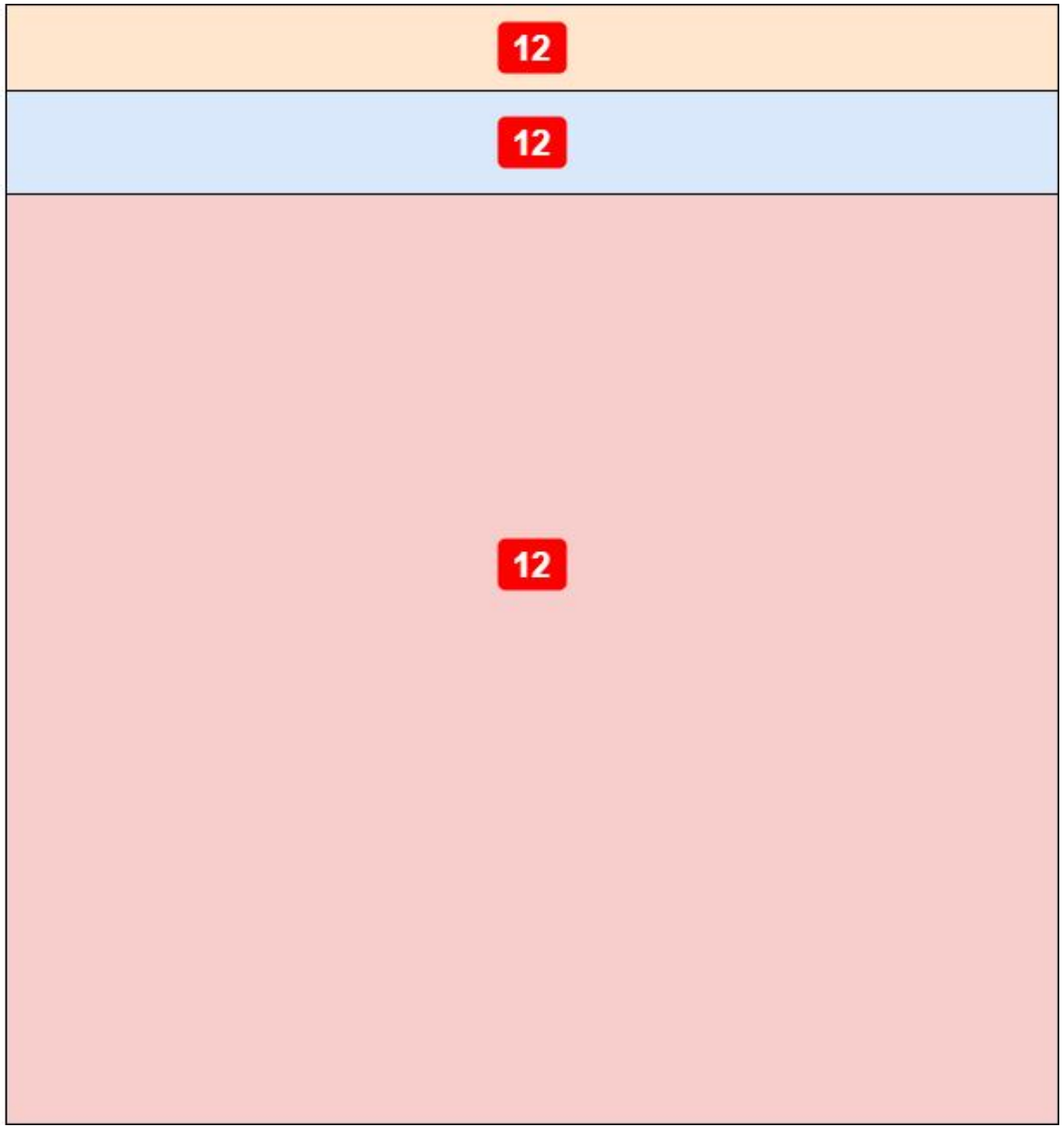
Layout
Create inline, horizontal, or complex grid-based layouts with your forms.

Validation
Validate your forms with custom or native validation behaviors and styles.

FPT Education
FPT UNIVERSITY

Exercise 8: Demo about Form Controls


Form đặt vé máy bay
Họ tên
Họ tên
vnđ
Phải nhập 5 ký tự, in hoa....
Địa chỉ
Phải nhập 5 ký tự, in hoa....
Đi từ
Hà nội
Đến
Hà nội
Chọn chiều đi (Khứ hồi)
Đi
Về
Đặt vé


12
12
12

Form đặt vé máy bay

Họ tên

Họ tên

vnđ

Phải nhập 5 ký tự, in hoa....

Địa chỉ

Phải nhập 5 ký tự, in hoa....

Đi từ

Hà nội

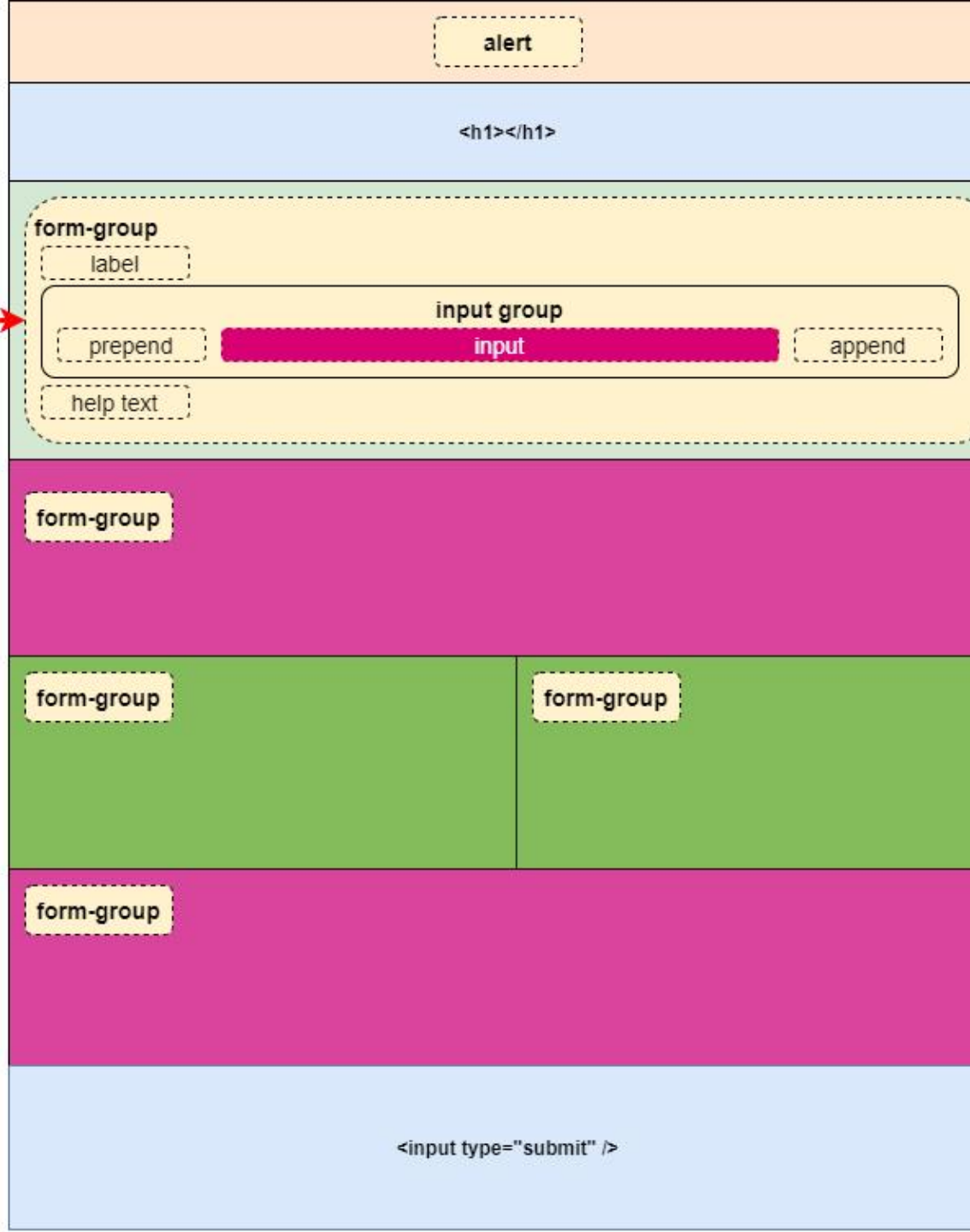
Đến

Hà nội

Chọn chiều đi (Khứ hồi)

- [ ] Đi
- [ ] Về

Đặt vé



FPT Education
FPT UNIVERSITY

Lab 2: Build the website interface

Summary

- Concepts were introduced:
  - Identify the purpose of using front-end UI frameworks in web design and development
  - Set up and Configure project with Bootstrap
  - Use the Bootstrap grid system to design responsive websites
  - Recognize how the JavaScript components provide dynamic behavior
  - Create various navigational elements using Tabs and Accordion
  - Create modals, tooltips and popovers to reveal content in your website
  - Build a carousel to show dynamic content on your website