FULL STACK DEVELOPMENT USING OPEN-SOURCE TECHNOLOGY

By: Mrs. Nidhi Divecha (ME CMPN)

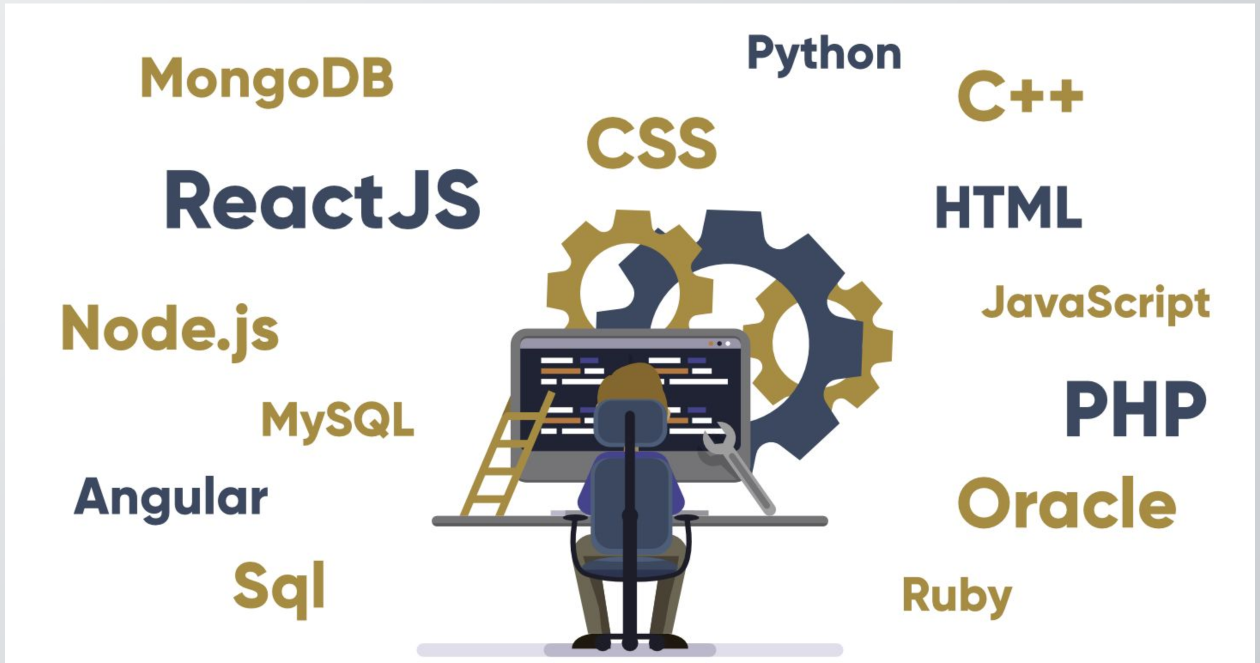
(UNIT 1)

WHAT YOU'LL LEARN

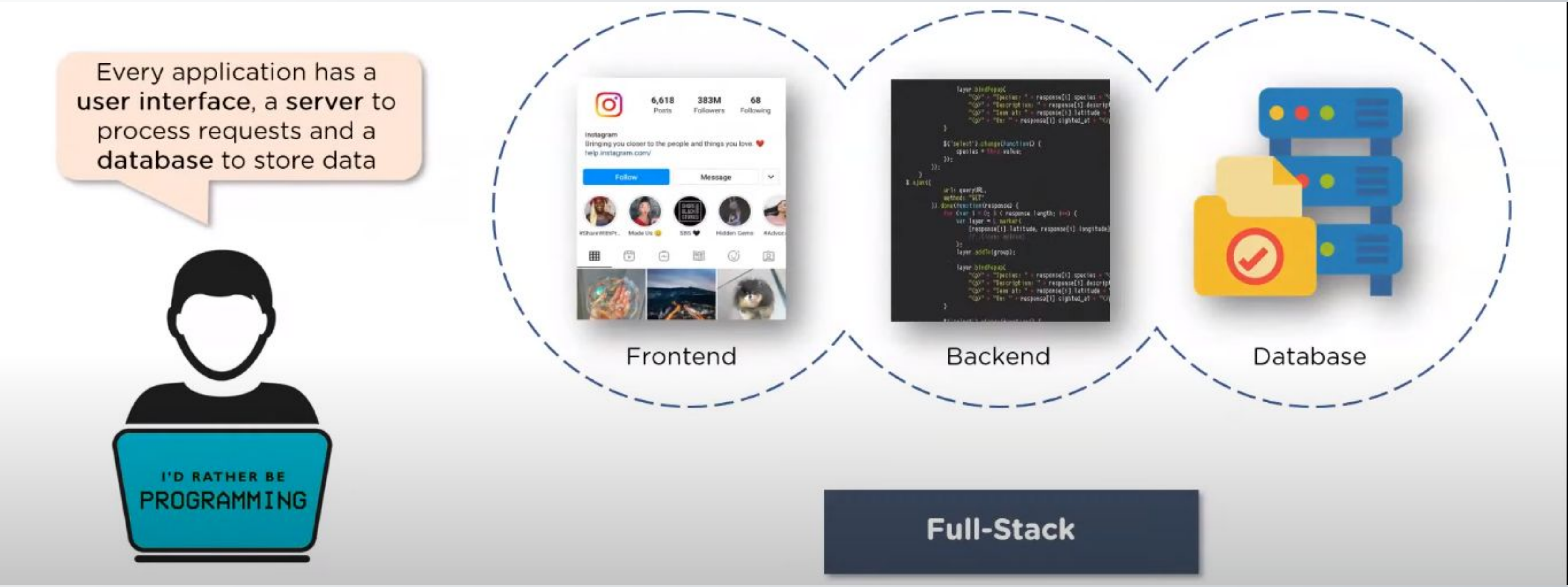
- Learn core concepts of both the front end and backend programming course
- Learn all about SQL and NoSQL databases
- Get familiar with the latest web developer technologies and ecosystems
- Learn complete web development process

WHAT IS FULL STACK DEVELOPMENT ?

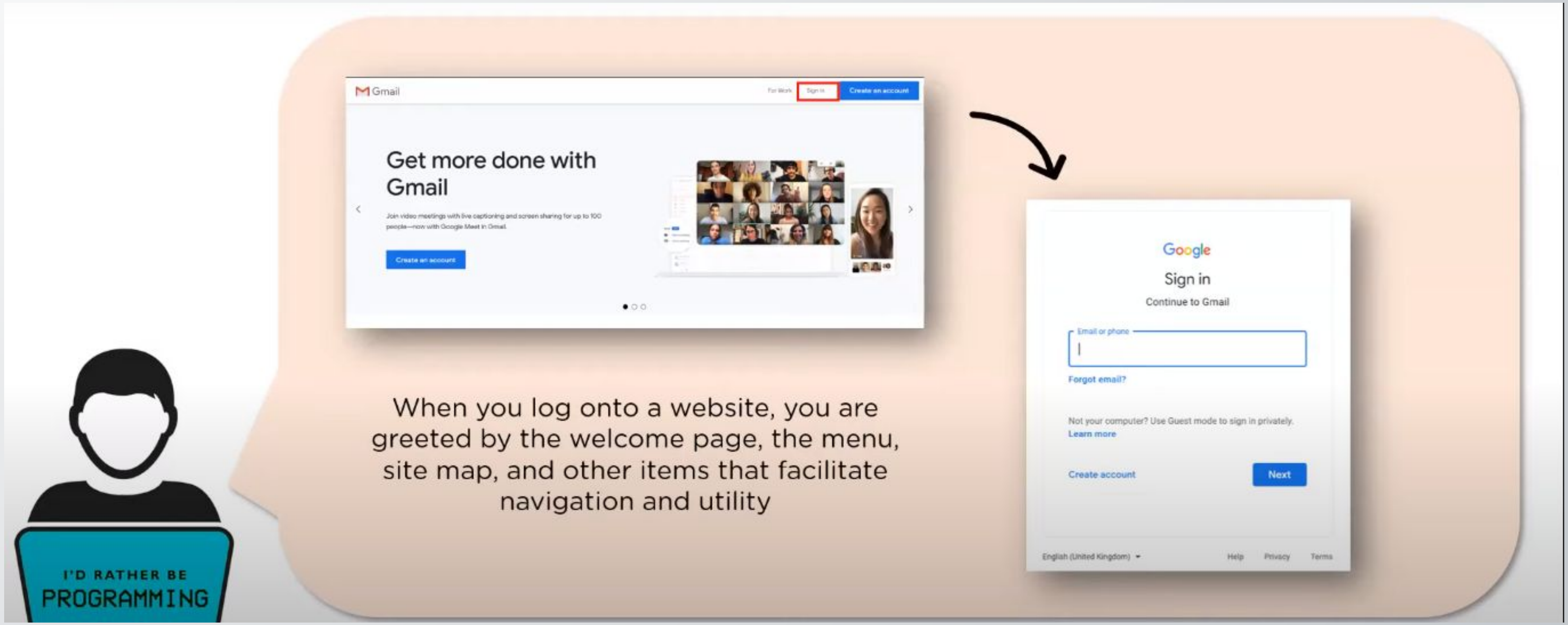
- It refers to the development of both **front end**(client side) and **back end**(server side) portions of web application.
- Full stack web developers can design complete web application and websites. They work on the frontend, backend, database and debugging of web application or websites.



- Every application has a user interface (front-end), a server (back-end) to process requests and a database to store the data.


Every application has a **user interface**, a **server** to process requests and a **database** to store data
I'D RATHER BE PROGRAMMING
Frontend
Backend
Database
Full-Stack

WHAT IS FRONT-END DEVELOPMENT?



Join video meetings with live captioning and screen sharing for up to 100 people—now with Google Meet in Gmail.

Not your computer? Use Guest mode to sign in privately.

When you log onto a website, you are greeted by the welcome page, the menu, site map, and other items that facilitate navigation and utility

- Essentially, web development has two parts – frontend and backend development.
- Thus, every web or mobile application includes two parts, a frontend, and a backend.
- **While the frontend comprises the visible part of the application with which users interact (user interface), the backend is where all the actual magic happens.**
- The backend of an application includes business logic (how the system functions and how the data flows via a series of tasks), how the data is stored, and where the solution runs.
- Both the frontend and backend combine to create the Full Stack.

TECHNOLOGY RELATED TO FULL STACK DEVELOPMENT:

Front end:

- It is the visible part of website or web application which is responsible for user experience. The user directly interacts with the front-end portion of the web application or website.

Front end Languages:

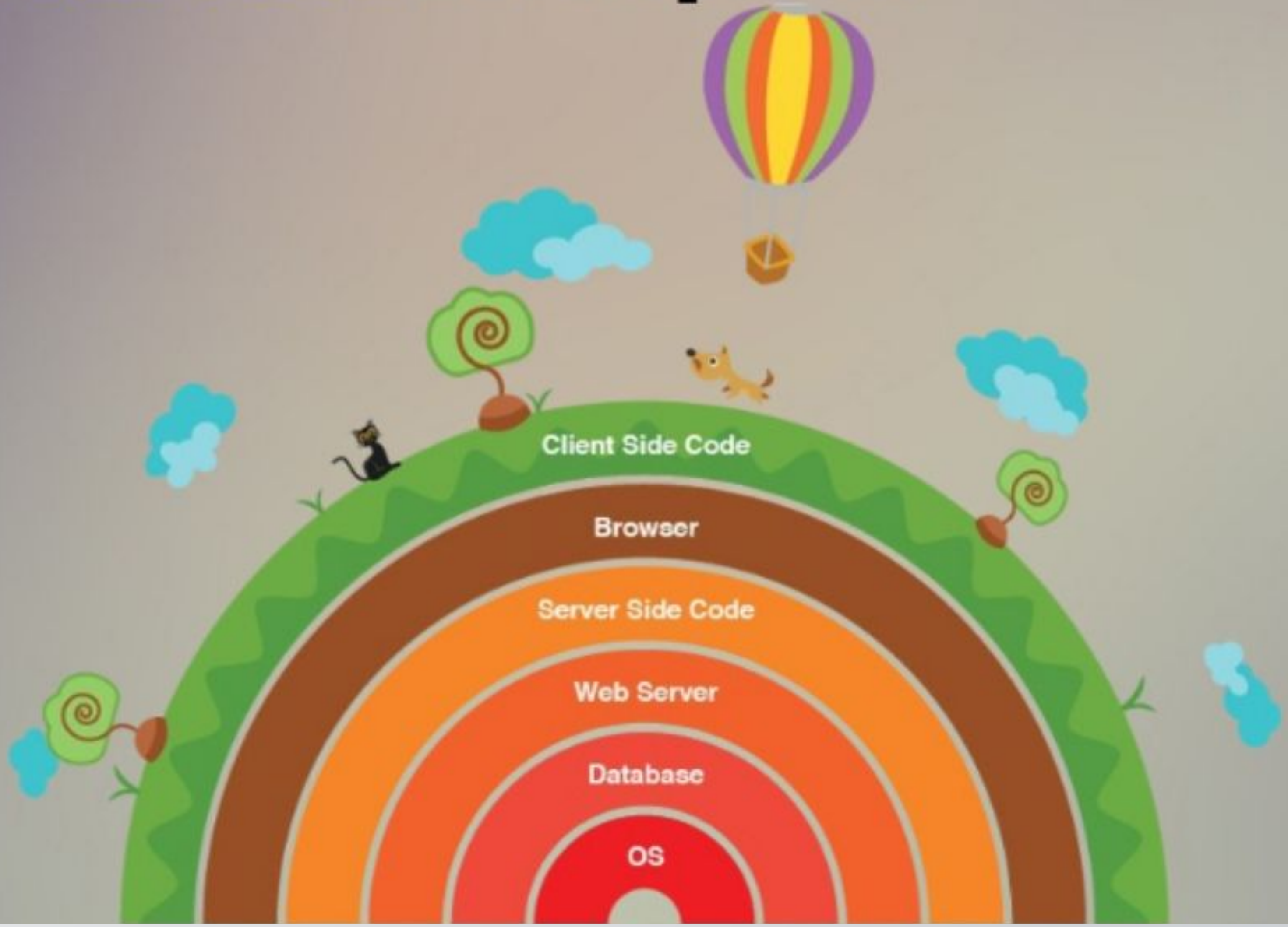
- **HTML:** HTML stands for Hyper Text Markup Language. It is used to design the front-end portion of web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within tag which defines the structure of web pages.
- **CSS:** Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
- **JavaScript:** JavaScript is a famous scripting language used to create the magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

Front-End Technologies
HTML5
CSS3
JavaScript
AngularJS
ReactJS

FRONT-END TECHNOLOGIES

- **HTML** – TAGS, ATTRIBUTES, LISTS, FORMS , ETC
- **HTML5** – VIDEO/AUDIO, CANVAS
- **CSS3** – POSITIONING , GRADIENTS, TRANSISTIONS
- **BOOTSTRAP** – ALERTS, GRIDS
- **JAVASCRIPT** – ARRAYS, OBJECTS, FORM VALIDATION

Layers in Full Stack Development



Back end:

- It refers to the server-side development of web application or website with a primary focus on how the website works. It is responsible for managing the database through queries and APIs by client-side commands. This type of website mainly consists of three parts front end, back end, and database.
- **PHP:** PHP is a server-side scripting language designed specifically for web development. Since, PHP code executed on server side, so it is called server-side scripting language.
- C++ It is a general-purpose programming language and widely used now a days for competitive programming. It is also used as backend language.
- **Java:** Java is one of the most popular and widely used programming language and platform. It is highly scalable. Java components are easily available.
- **Python:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
- **JavaScript:** Javascript can be used as both (front end and back end) programming languages.
- **Node.js:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that NodeJS is not a framework and it's not a programming language. Most of the people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as Paypal, Uber, Netflix, Wallmart and so on.

BACK-END TECHNOLOGIES

- **NODE.JS** – TEMPLATES, AUTHENTICATION
- **PHP** – OOP, MYSQL
- **JAVASCRIPT**
- **PYTHON** – DJANGO, FLASK

DATABASES

- MYSQL
- PostgreSQL
- MongoDB
- CouchDB
- Neo4J

- As the smart phone usage, it set to double in next 3 years, the demand for mobile app developers for both Android and iOS is on the rise.
- If you have the fundamental knowledge of programming languages , then a full stack mobile app development course can prepare you a bright career in this field.

BOOTSTRAP INTRODUCTION

- Bootstrap is a **free and open-source tool** collection for creating responsive websites and web applications.
- It is the most popular **HTML, CSS, and JavaScript framework** for developing responsive, mobile-first websites.
- Bootstrap is promoted as being ***One framework, every device***. This is because websites built with Bootstrap will automatically scale between devices — whether the device is a mobile phone, tablet, laptop, desktop computer, screen reader, etc.
- All thanks to Bootstrap developers - **Mark Otto and Jacob Thornton of Twitter.**
- Bootstrap 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
- Bootstrap 5 is the newest version of Bootstrap.

WHY USE BOOTSTRAP?

- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)
- **Easy customization**: It provides some built-in components and functionalities that are easy for customizing.
- It is an open-source framework with web-based customization.
- It is Free! Available on www.getbootstrap.com

ADVANTAGES AND FEATURES OF BOOTSTRAP

Bootstrap Advantages

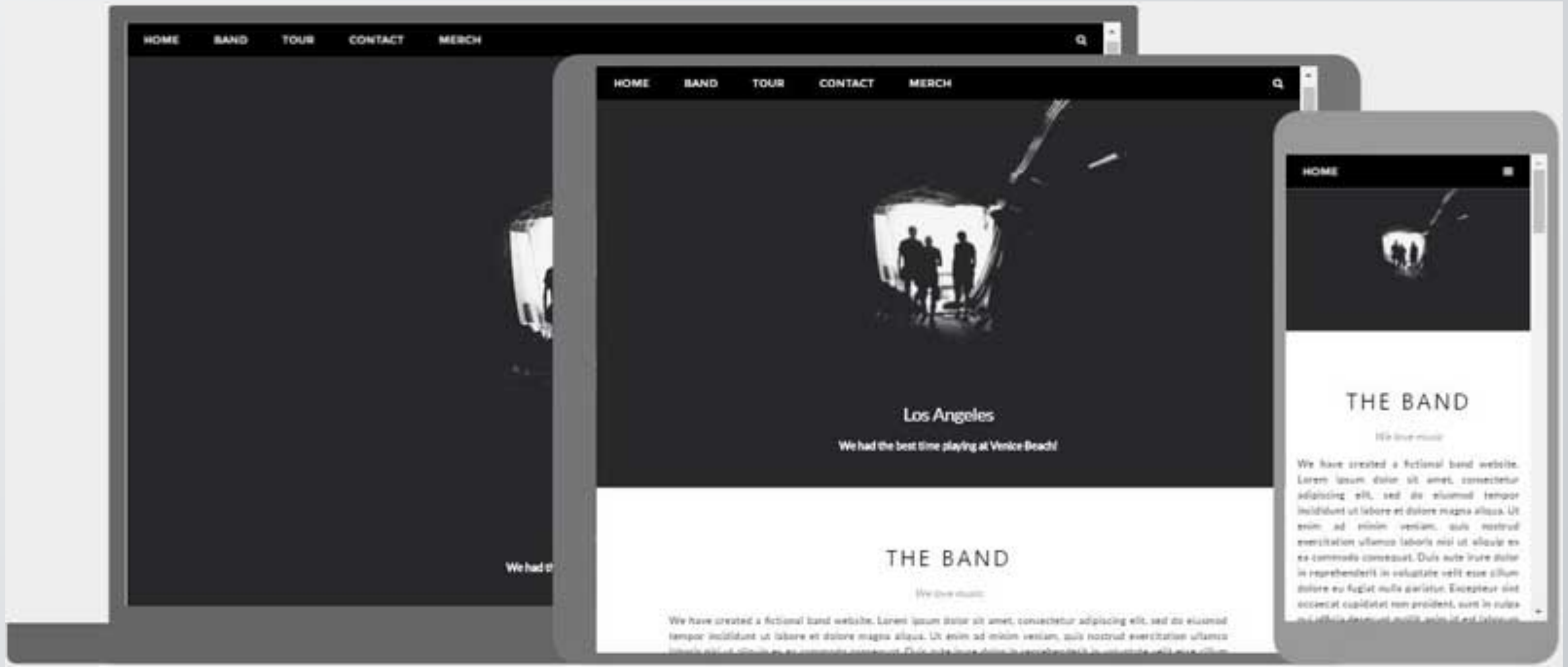
- Not Reinventing the Wheel. We don’t have to write more code.
- Extensive UI with Bootstrap themes.
- Open Source. You get to know the source code and change it.

Bootstrap Features

- HTML and CSS based design templates for forms, buttons, navigation, and other components.
- Modularity. Change or remove components or component colors.
- Re-usable code with CSS classes.
- Out of the box JavaScript components with additional features like tooltips, modal windows, etc.

WHAT IS RESPONSIVE WEB DESIGN?

- **Responsive web design** is about creating web sites **which automatically adjust themselves** to look good on all devices, from small phones to large desktops.
- Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones):

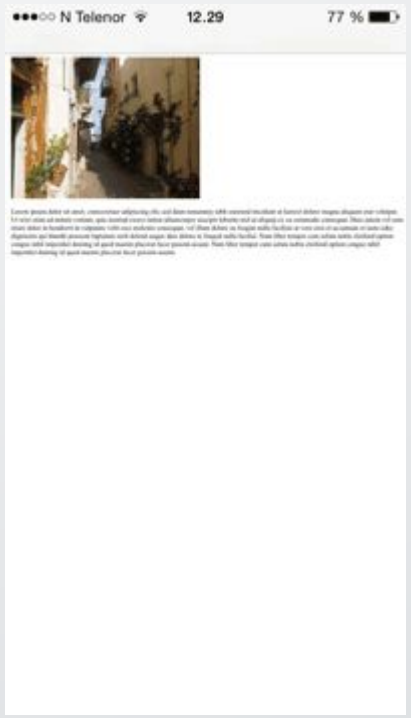


Setting The Viewport

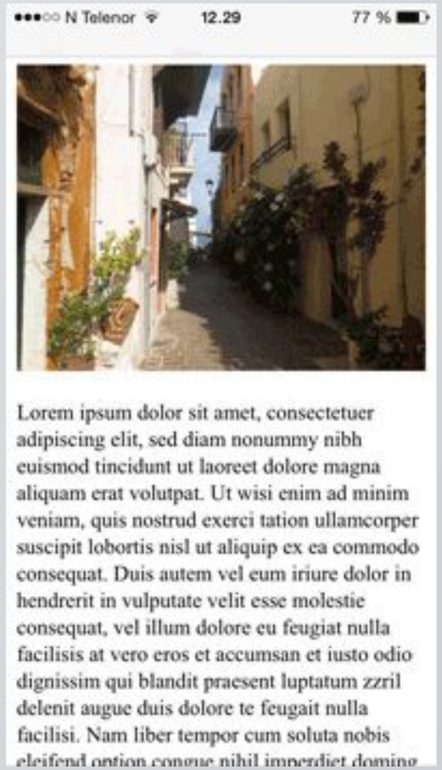
- To create a responsive website, add the following <meta> tag to all your web pages:

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

- This will set the **viewport of your page**, which will give the browser instructions on how to control the page's dimensions and scaling.
- Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:



Without the viewport meta tag:



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

With the viewport meta tag:

BOOTSTRAP COMPONENTS

- DROPDOWN
- BUTTON GROUP
- NAVIGATION
- PAGINATION
- THUMBNAILS
- ALERTS
- PROGRESS BARS
- PANELS

BOOTSTRAP CSS

- TABLES
- FORMS
- BUTTONS
- IMAGES
- ETC

BOOTSTRAP JAVASCRIPT

- Modals
- Dropdown
- Popover
- Alerts
- Collapse
- Carousal
- Tool tip etc

BOOTSTRAP CONTAINERS

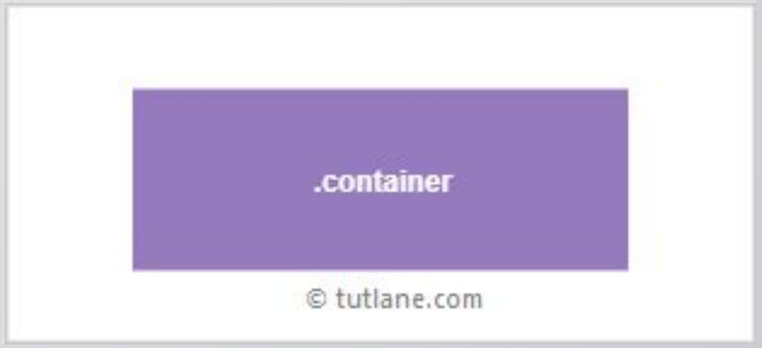
- In bootstrap, **container** is used to set the content’s margin.
- It contains row elements, and the row elements are container of columns. This is known as **grid system**.

There are two container classes in bootstrap:

Bootstrap Container Class

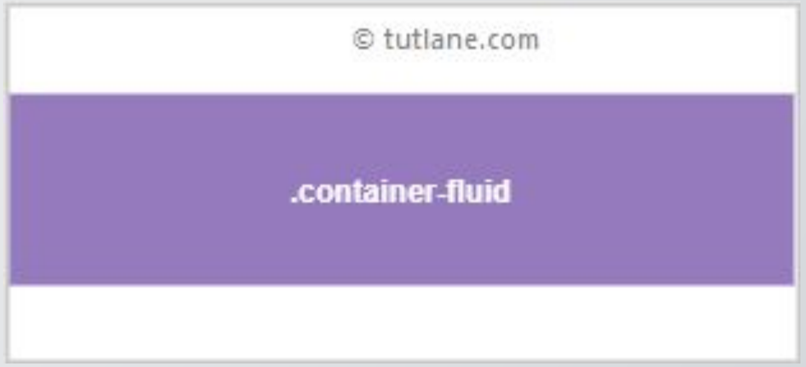
- In bootstrap, the .container class is useful to provide a responsive fixed width container that means the max-width will vary at each breakpoint.
- Following is the pictorial representation of site layout when we use .container class to wrap site contents.

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



Bootstrap Container-fluid Class

- In bootstrap, the .container-fluid class is useful to provide the responsive full-width container that means the container will be 100% wide all the time.
- Following is the pictorial representation of site layout when we use .container-fluid class to wrap the site contents.



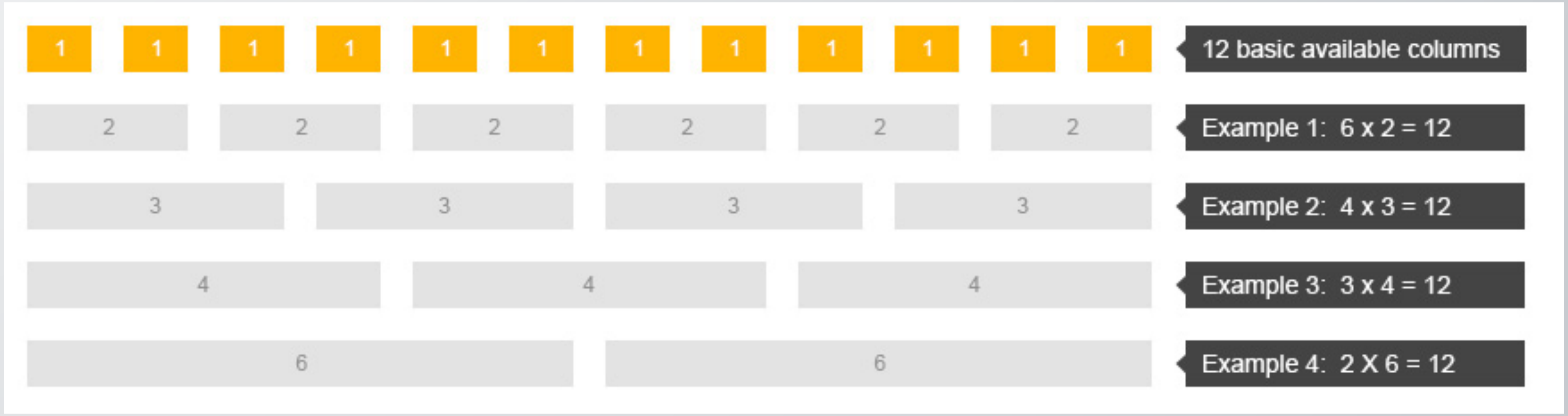
```
<div class="container-fluid">
   <!-- Content here -->
</div>
```

- Bootstrap has a concept of containing element to **wrap site contents.**
- <div class= “container”>
  - <h1>My first Bootstrap page</h1>
  - <p>This is some text</p>
  - </div>

- <div class= “container-fluid”>
  - <h1>My first Bootstrap page</h1>
  - <p>This is some text</p>
  - </div>

BOOTSTRAP GRID SYSTEM

- Bootstrap’s grid system allows up to **12 columns** across the page.
- The bootstrap grid system will use a series of containers, rows, and columns to define the layout and to align the content appropriately based on the device.
- You can divide the container in rows and each row in columns with space multiple of 12.
- You can use any combination that the sum be equal to 12

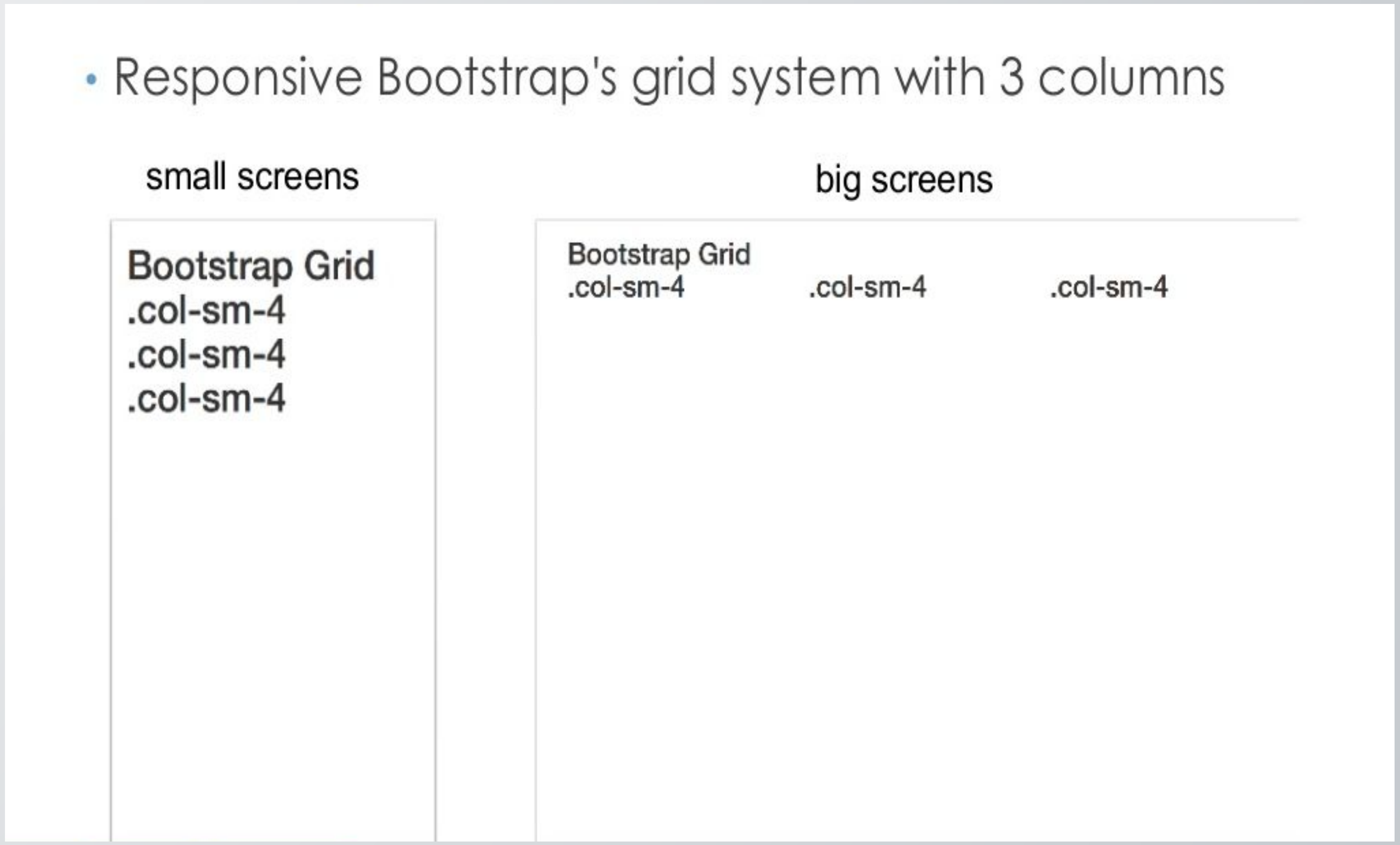


STRUCTURE OF BOOTSTRAP GRID

- To create responsive web page layouts using a bootstrap grid system, we need to use rows and columns within the container or container-fluid like as shown below.

-
  ```
  <div class="container">
      <div class="row">
        <div class="col-*-*"></div>
        <div class="col-*-*"></div>
        <div class="col-*-*"></div>
      </div>
  </div>
  ```
- In the bootstrap grid system, we are allowed to add up to **12** columns within a row. If we add more than **12** columns within a row, those extra columns will wrap into a new line.

```
<div class="container">
<h1>Bootstrap Grid</h1>
<div class="row">
<div class="col-sm-4">.col-sm-4</div>
<div class="col-sm-4">.col-sm-4</div>
<div class="col-sm-4">.col-sm-4</div>
</div>
</div>
```



- Responsive Bootstrap's grid system with 3 columns

small screens

Bootstrap Grid
.col-sm-4
.col-sm-4
.col-sm-4

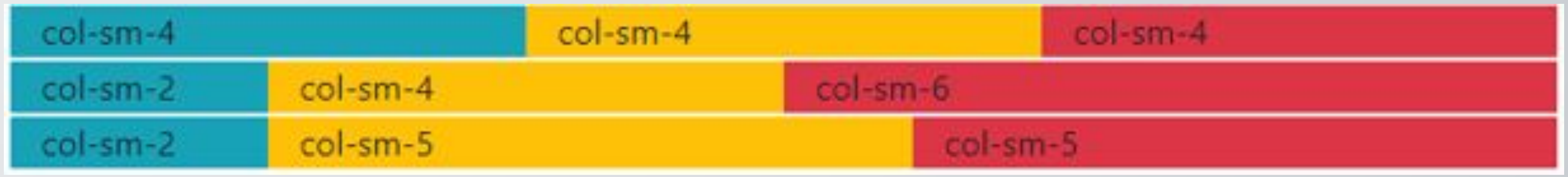
big screens

Bootstrap Grid
.col-sm-4 .col-sm-4 .col-sm-4

BOOTSTRAP GRID CLASSES

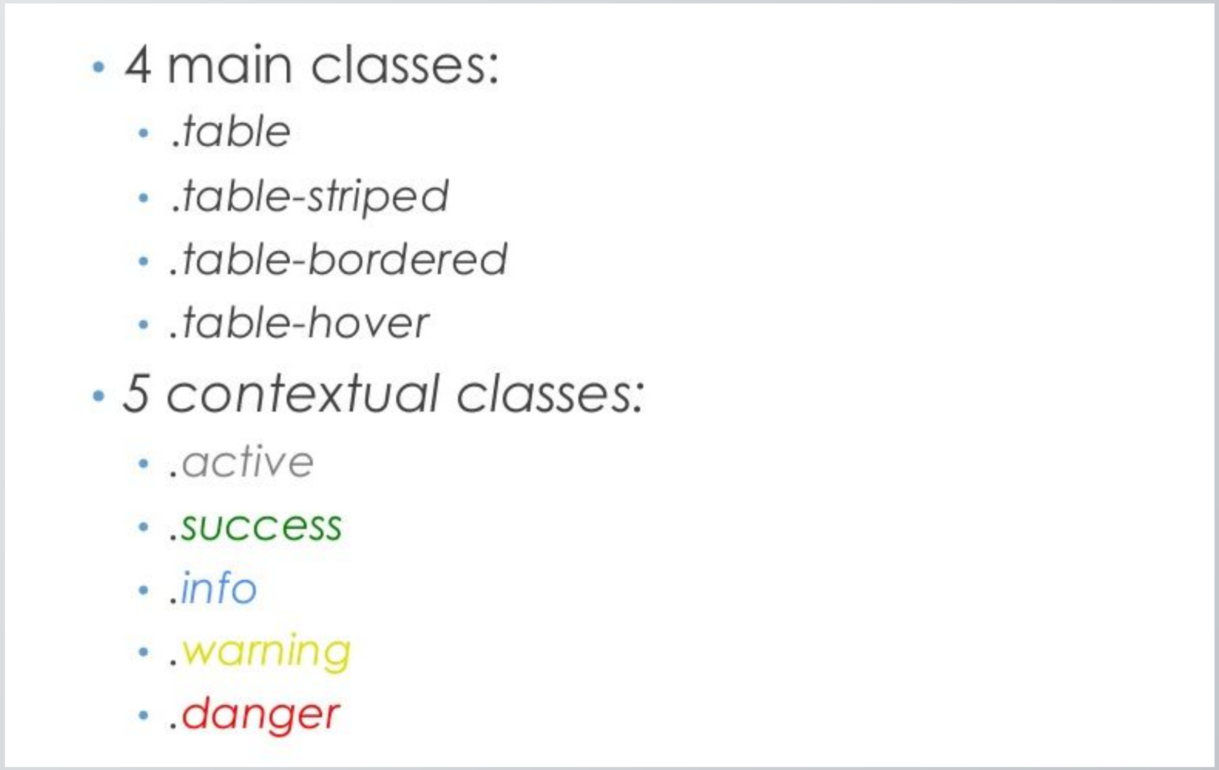
- Bootstrap 4 has included 5 predefined grid classes to scale the content depending on the device or viewport size.
- .col-* (Extra small – for phones)
- .col-sm-* (Small – for tablets)
- .col-md-* (Medium- for typical desktops/laptops)
- .col-lg-* (Large for large desktops)
- .col-xl-*

-
  ```
  <div class="container">
      <div class="row">
         <div class="col-sm-4">col-sm-4</div>
         <div class="col-sm-4">col-sm-4</div>
         <div class="col-sm-4">col-sm-4</div>
      </div>
      <div class="row">
         <div class="col-sm-2">col-sm-2</div>
         <div class="col-sm-4">col-sm-4</div>
         <div class="col-sm-6">col-sm-6</div>
      </div>
      <div class="row">
         <div class="col-sm-2">col-sm-2</div>
         <div class="col-sm-5">col-sm-5</div>
         <div class="col-sm-5">col-sm-5</div>
      </div>
  </div>
  ```

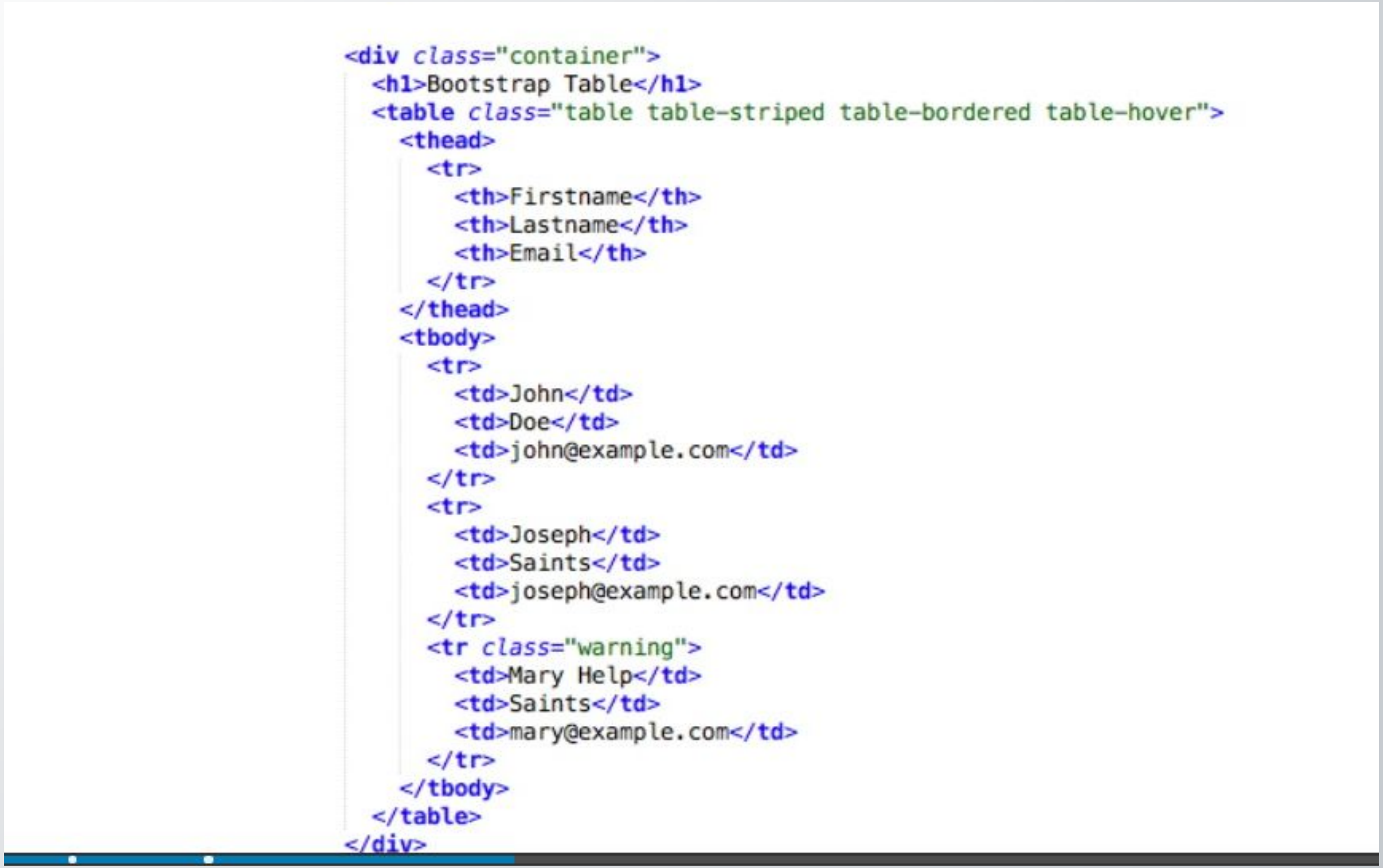


BOOTSTRAP TABLES

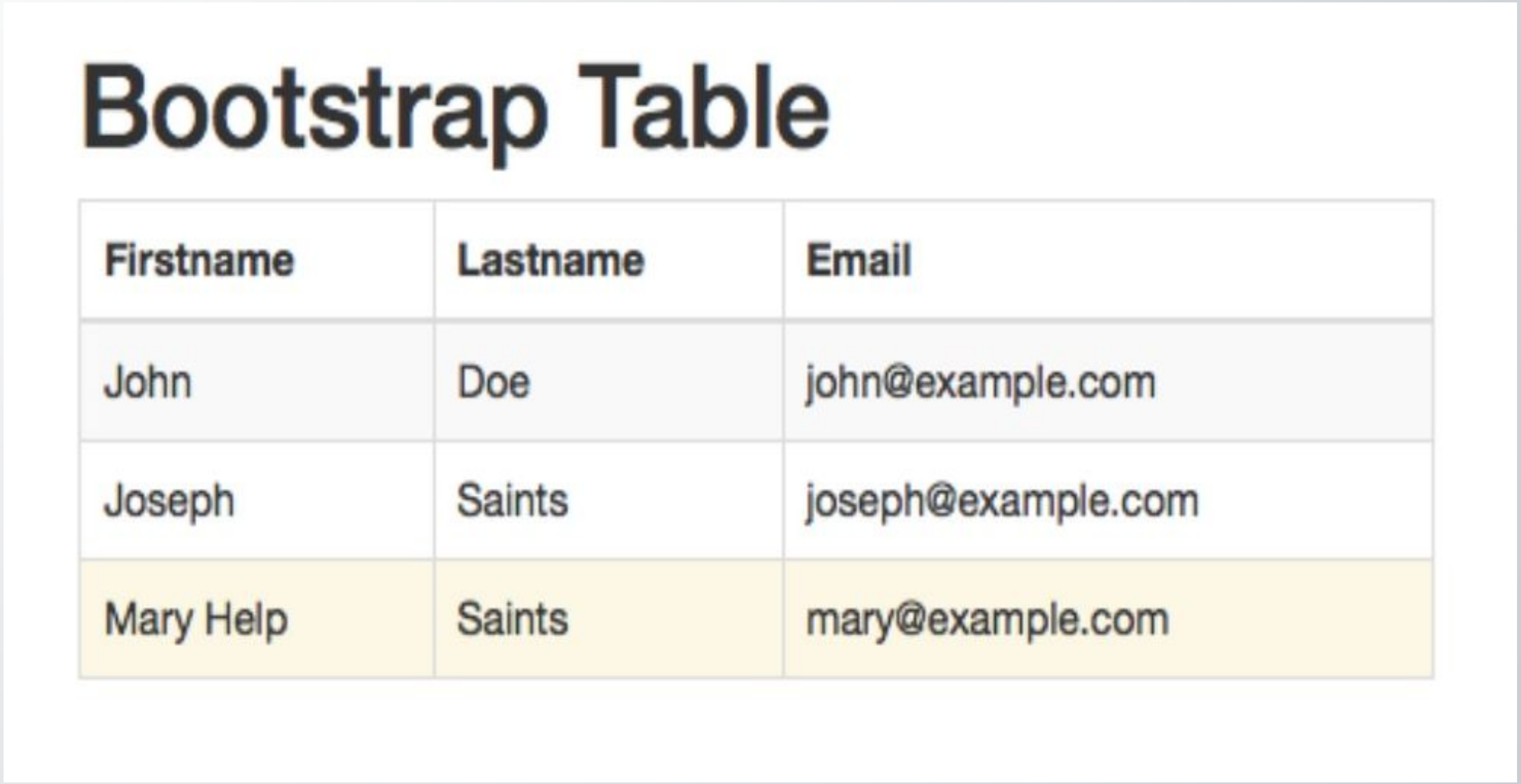
- The <table> tag defines an HTML table.
- Each table row is defined with a <tr> tag. Each table header is defined with a <th> tag. Each table data/cell is defined with a <td> tag.
- By default, the text in <th> elements are bold.
- By default, the text in <td> elements are regular and left-aligned.



- 4 main classes:
  - *.table*
  - *.table-striped*
  - *.table-bordered*
  - *.table-hover*
- *5 contextual classes:*
  - *.active*
  - *.success*
  - *.info*
  - *.warning*
  - *.danger*



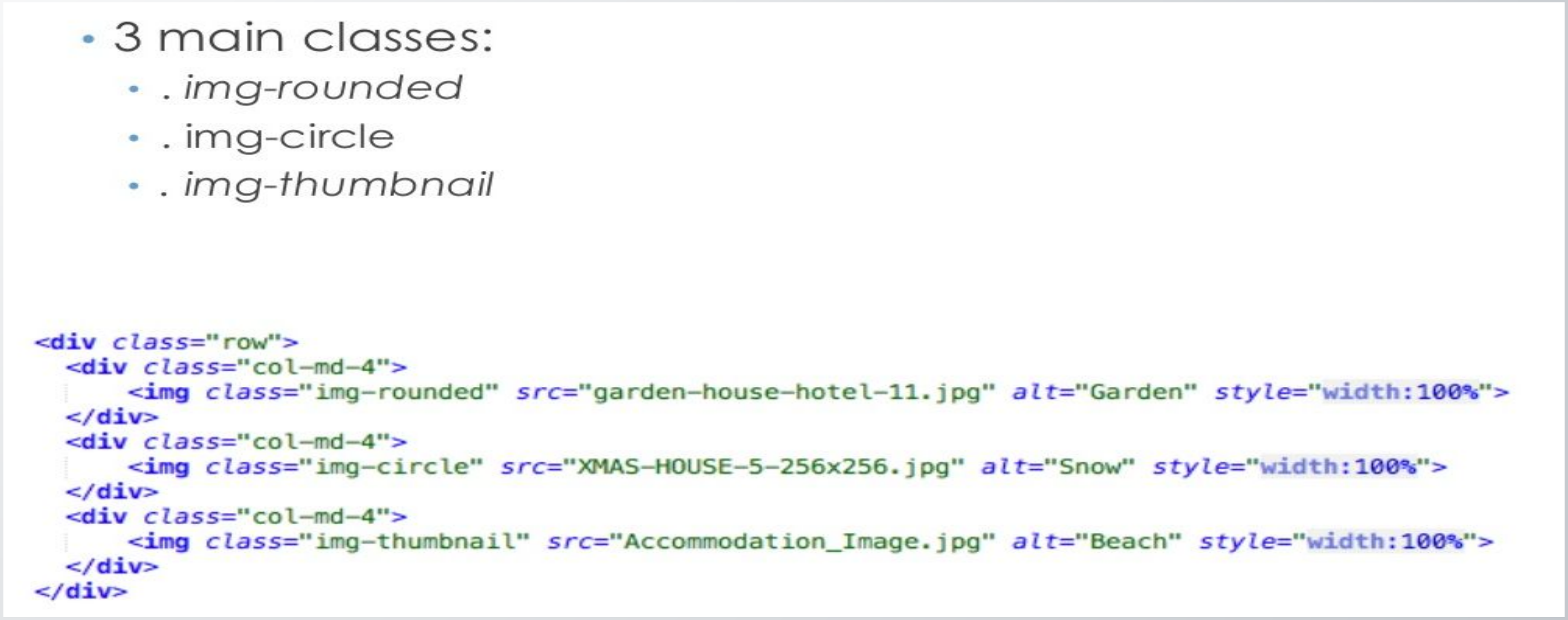
```
<div class="container">
  <h1>Bootstrap Table</h1>
  <table class="table table-striped table-bordered table-hover">
    <thead>
      <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>John</td>
        <td>Doe</td>
        <td>john@example.com</td>
      </tr>
      <tr>
        <td>Joseph</td>
        <td>Saints</td>
        <td>joseph@example.com</td>
      </tr>
      <tr class="warning">
        <td>Mary Help</td>
        <td>Saints</td>
        <td>mary@example.com</td>
      </tr>
    </tbody>
  </table>
</div>
```



Bootstrap Table

Firstname	Lastname	Email
John	Doe	john@example.com
Joseph	Saints	joseph@example.com
Mary Help	Saints	mary@example.com

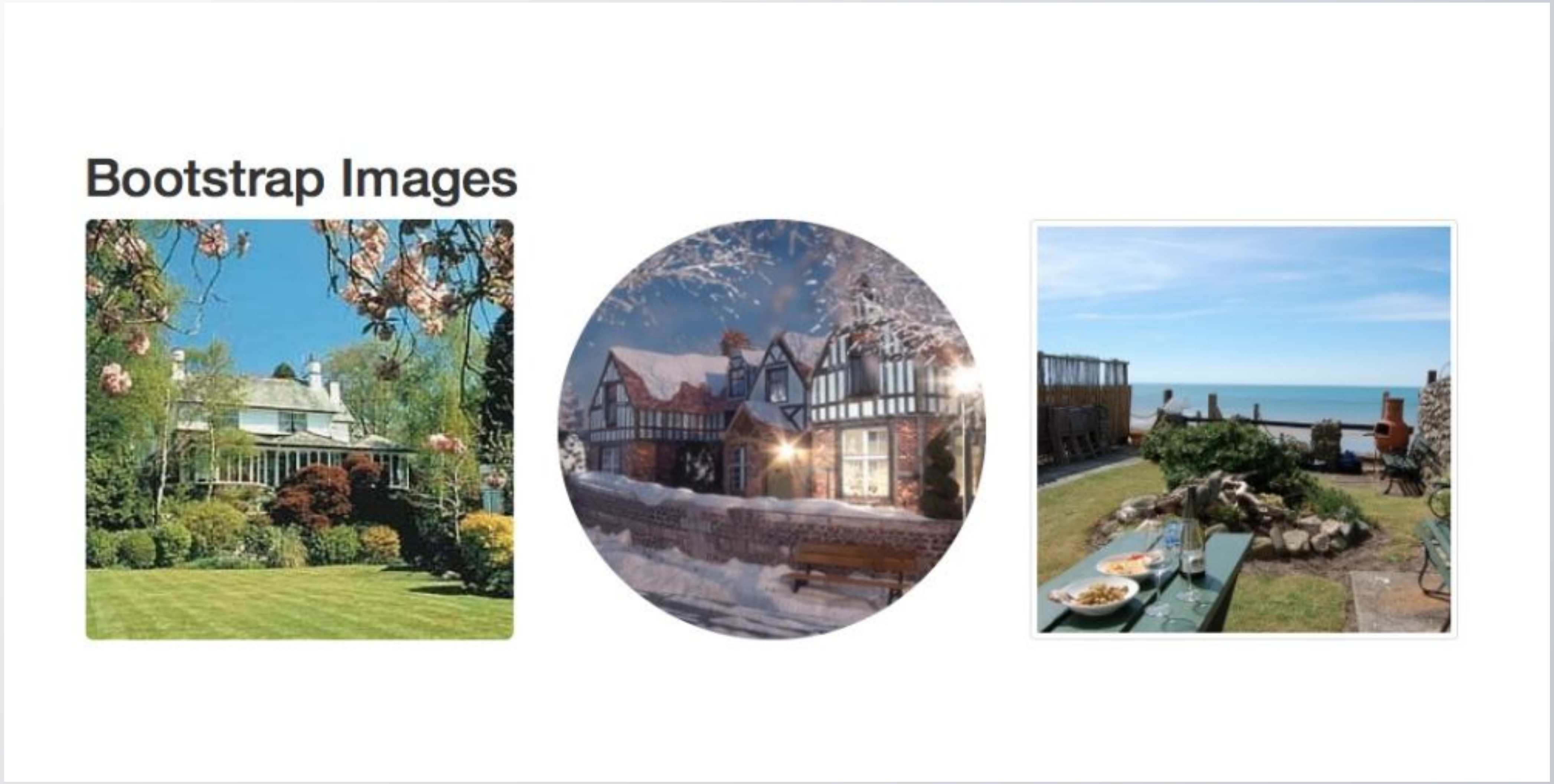
BOOTSTRAP IMAGES



- 3 main classes:
  - *. img-rounded*
  - . img-circle
  - *. img-thumbnail*

```
<div class="row">
  <div class="col-md-4">
      <img class="img-rounded" src="garden-house-hotel-11.jpg" alt="Garden" style="width:100%">
  </div>
  <div class="col-md-4">
      <img class="img-circle" src="XMAS-HOUSE-5-256x256.jpg" alt="Snow" style="width:100%">
  </div>
  <div class="col-md-4">
      <img class="img-thumbnail" src="Accommodation_Image.jpg" alt="Beach" style="width:100%">
  </div>
</div>
```

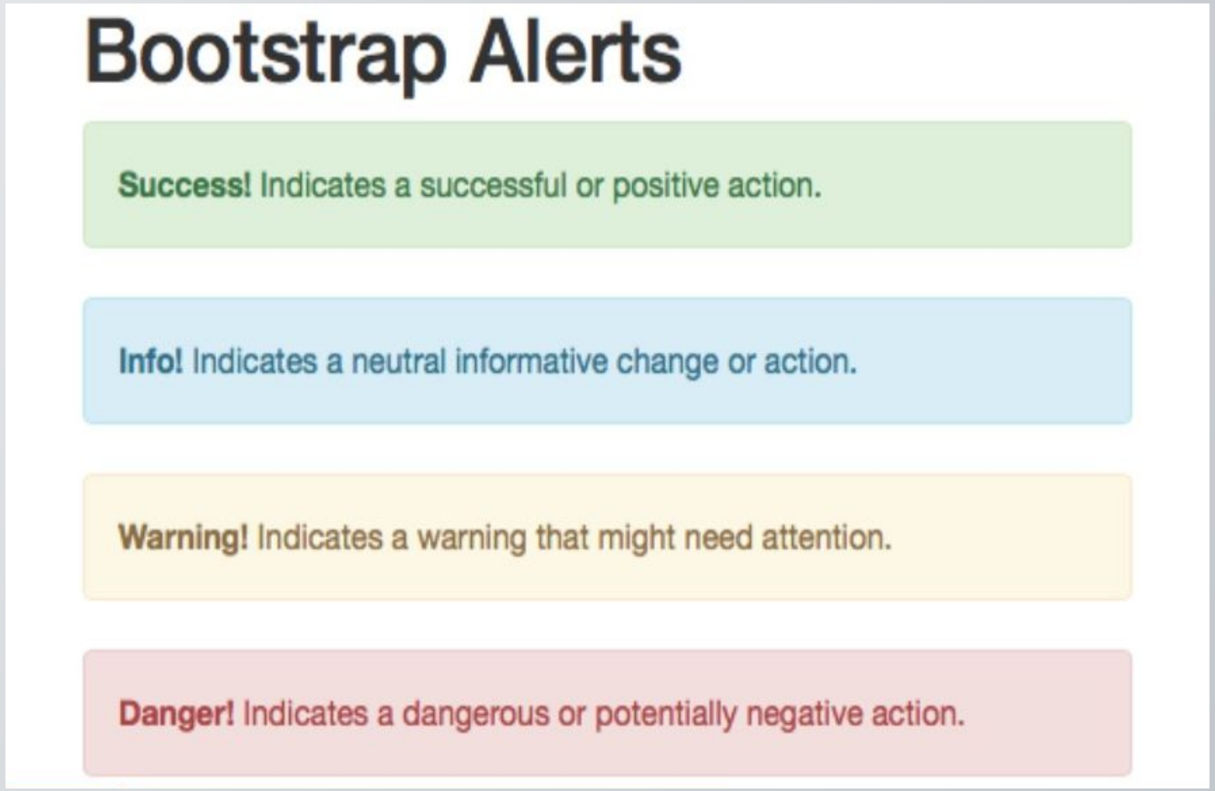
img-thumbnail to give an image a rounded 1px border appearance



Bootstrap Images

BOOTSTRAP ALERTS

- Bootstrap provides an easy way to create predefined alert messages.
- Alerts are created with the .alert class, followed by one of the four contextual classes
- .alert-success (green)
- .alert-info (light blue)
- .alert-warning (yellow)
- .alert-danger (red)



Bootstrap Alerts

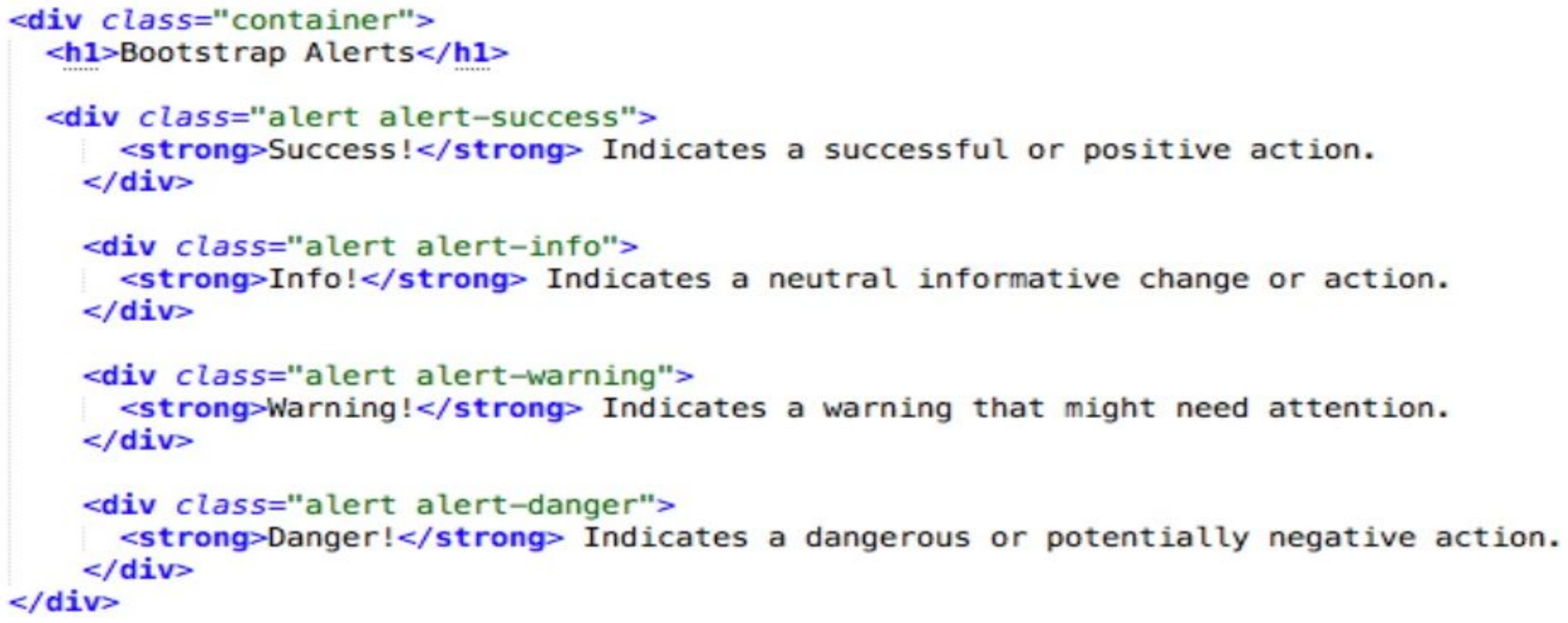
Success! Indicates a successful or positive action.

Info! Indicates a neutral informative change or action.

Warning! Indicates a warning that might need attention.

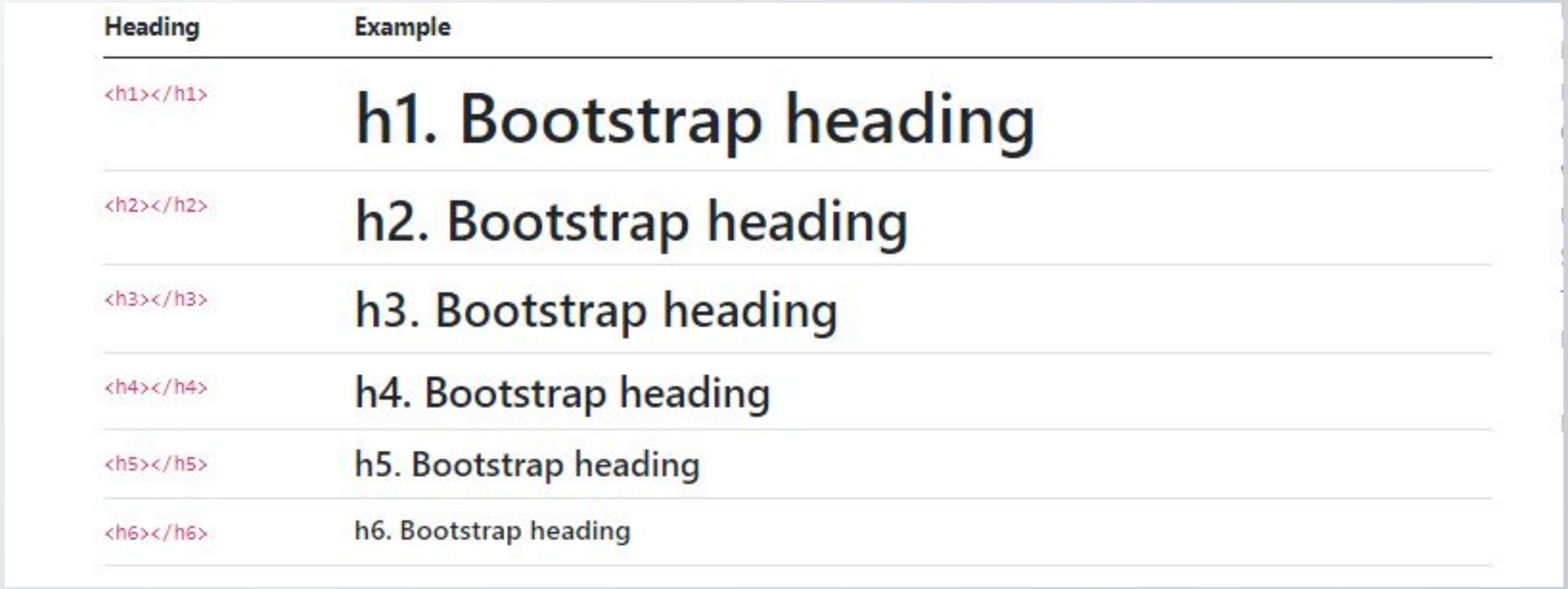
Danger! Indicates a dangerous or potentially negative action.

Bootstrap Alerts



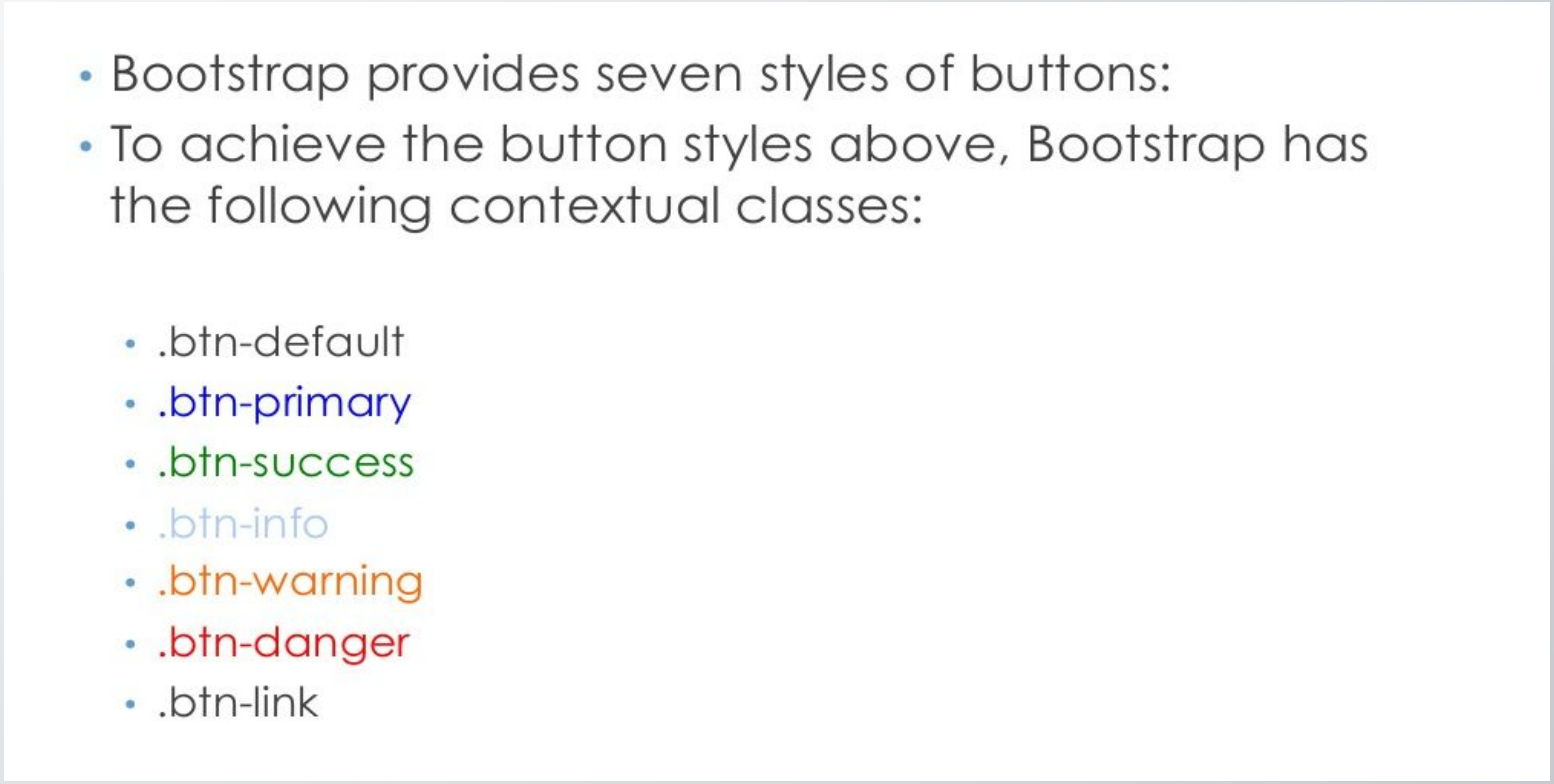
```html
<div class="container">
  <h1>Bootstrap Alerts</h1>

  <div class="alert alert-success">
      <strong>Success!</strong> Indicates a successful or positive action.
    </div>

    <div class="alert alert-info">
      <strong>Info!</strong> Indicates a neutral informative change or action.
    </div>

    <div class="alert alert-warning">
      <strong>Warning!</strong> Indicates a warning that might need attention.
    </div>

    <div class="alert alert-danger">
      <strong>Danger!</strong> Indicates a dangerous or potentially negative action.
    </div>
</div>
```

HEADINGS & PARAGRAPHS



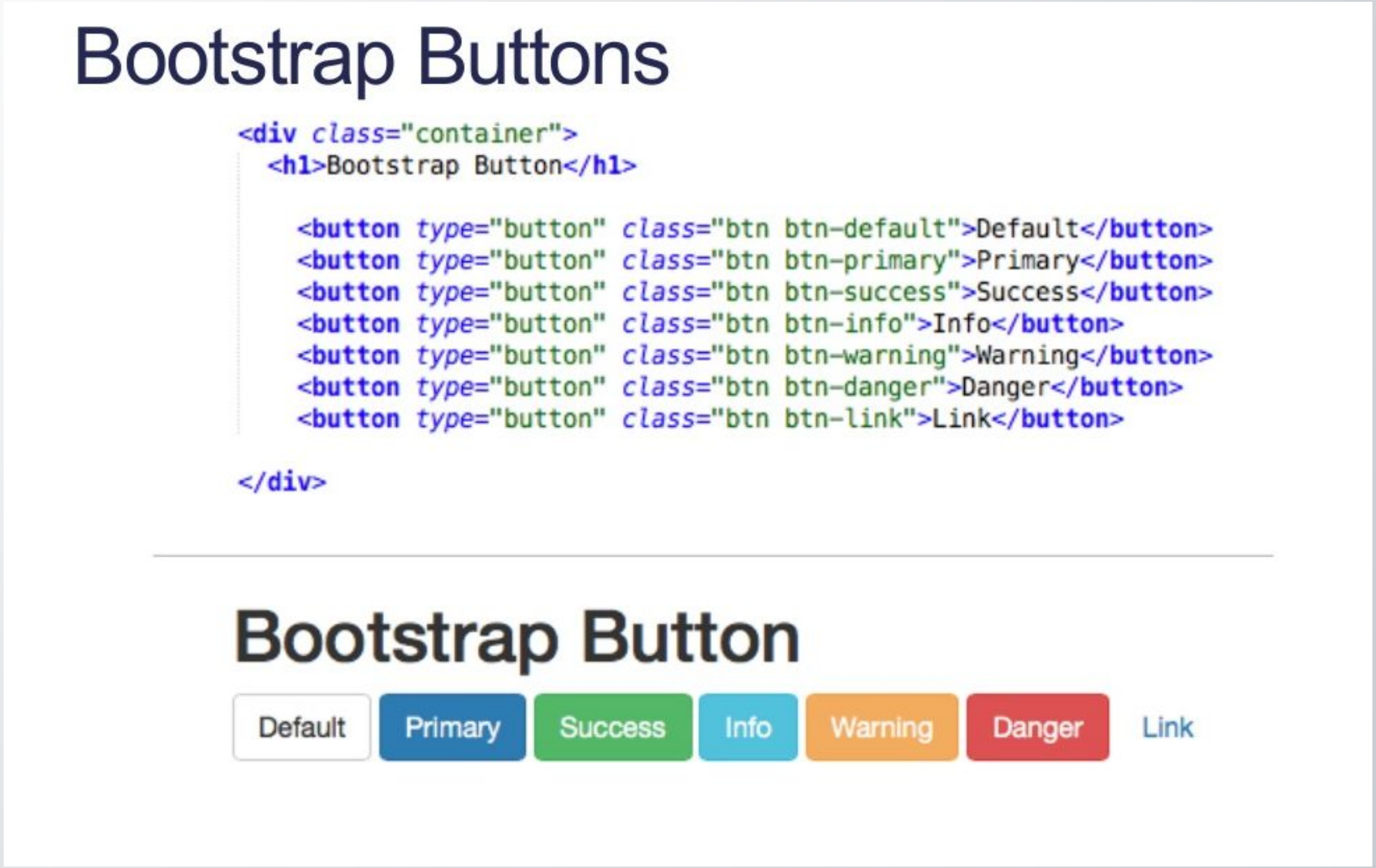
Heading	Example
`<h1></h1>`	h1. Bootstrap heading
`<h2></h2>`	h2. Bootstrap heading
`<h3></h3>`	h3. Bootstrap heading
`<h4></h4>`	h4. Bootstrap heading
`<h5></h5>`	h5. Bootstrap heading
`<h6></h6>`	h6. Bootstrap heading

BOOTSTRAP BUTTONS



- Bootstrap provides seven styles of buttons:
- To achieve the button styles above, Bootstrap has the following contextual classes:
  - .btn-default
  - .btn-primary
  - .btn-success
  - .btn-info
  - .btn-warning
  - .btn-danger
  - .btn-link

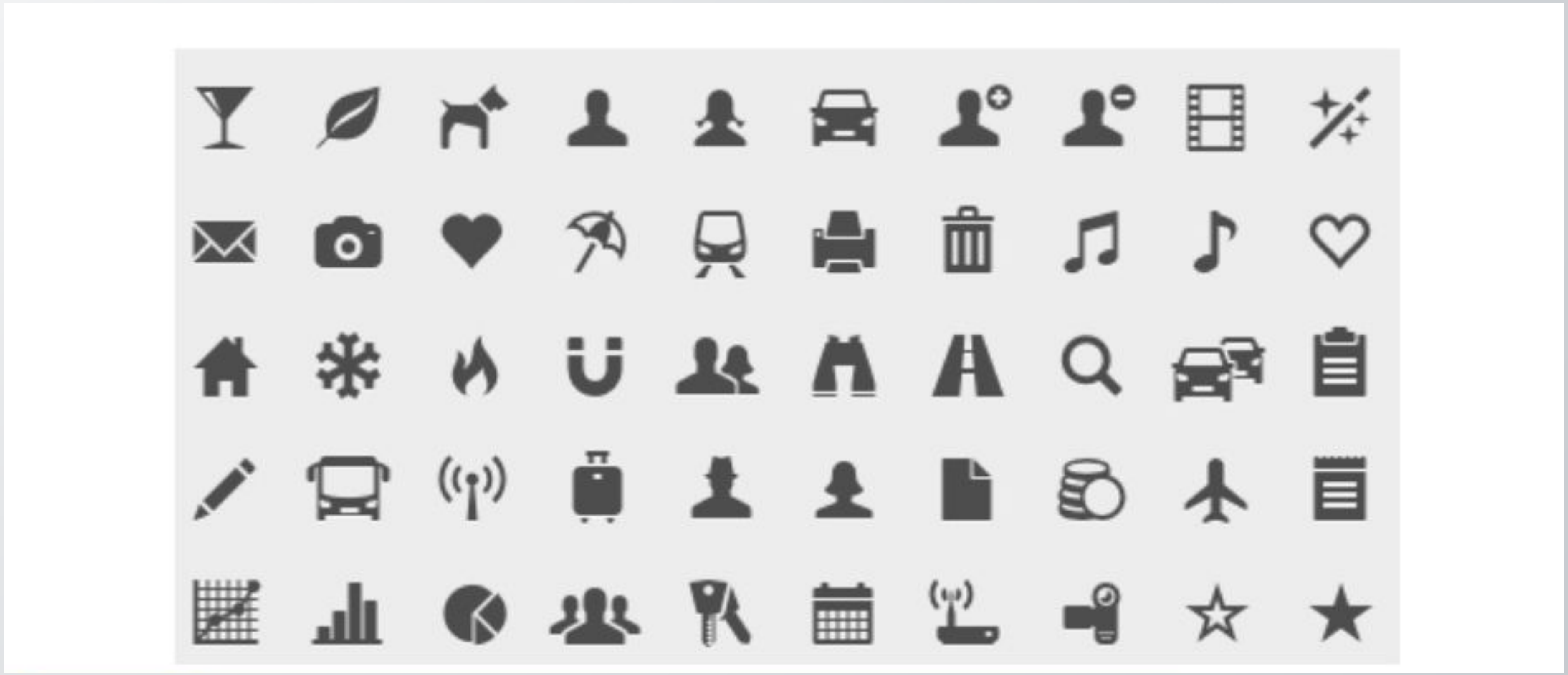
Bootstrap Buttons

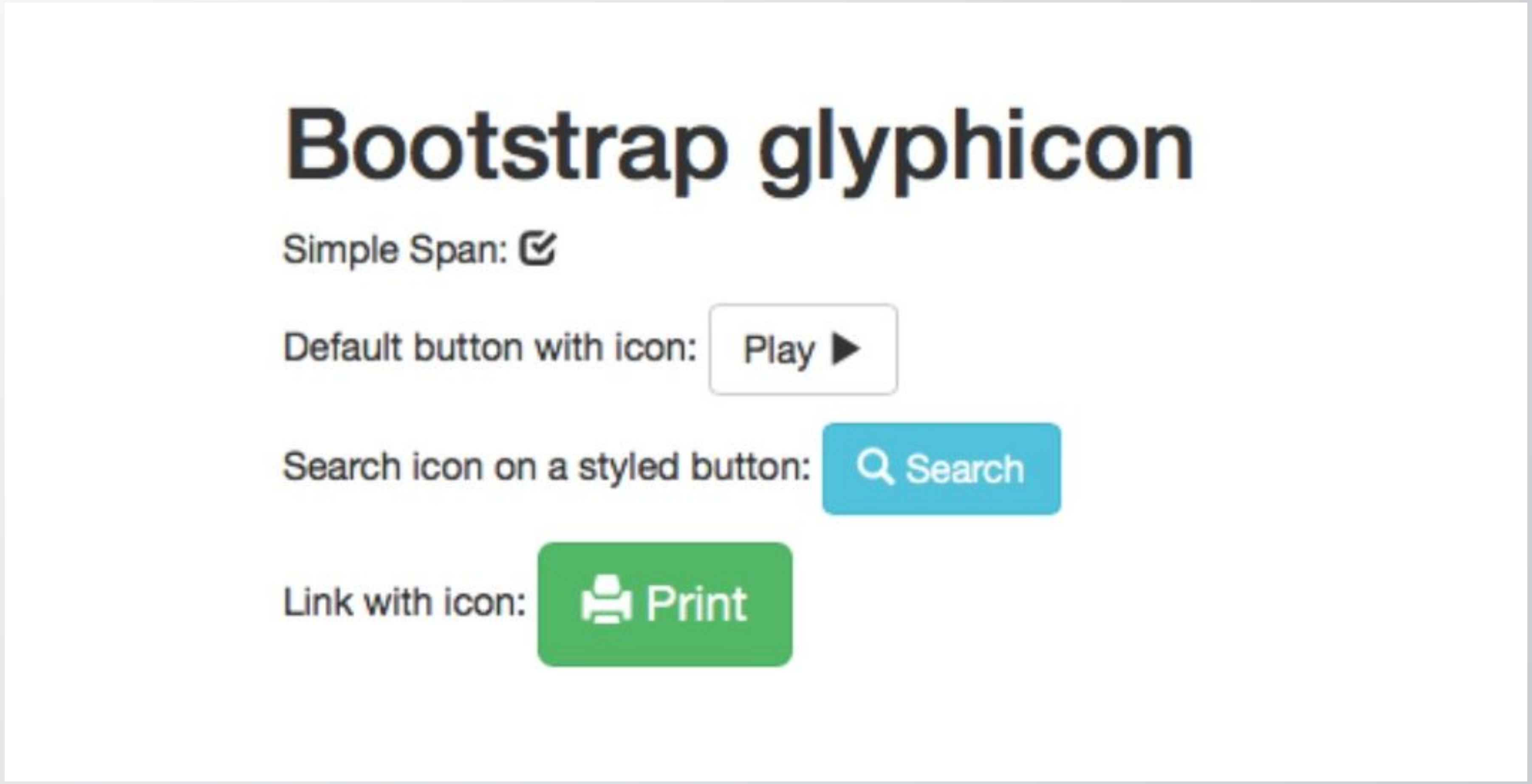


```
<div class="container">
  <h1>Bootstrap Button</h1>

    <button type="button" class="btn btn-default">Default</button>
    <button type="button" class="btn btn-primary">Primary</button>
    <button type="button" class="btn btn-success">Success</button>
    <button type="button" class="btn btn-info">Info</button>
    <button type="button" class="btn btn-warning">Warning</button>
    <button type="button" class="btn btn-danger">Danger</button>
    <button type="button" class="btn btn-link">Link</button>

</div>
```

BOOTSTRAP GLYPHICONS(ICONS)

- Bootstrap provides 260 glyphicons





Bootstrap glyphicon

Simple Span: ☑

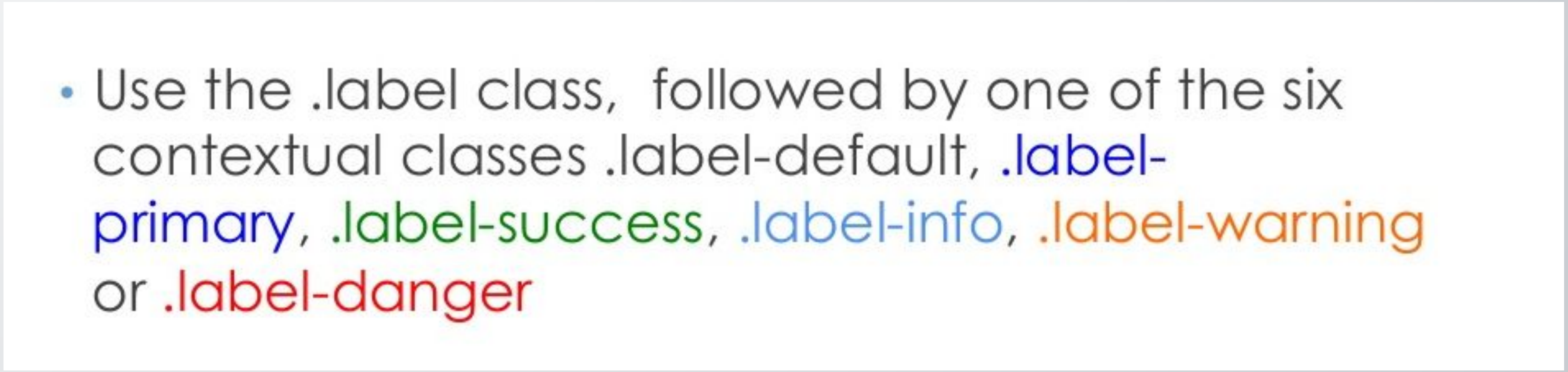
Default button with icon: Play ▶

Search icon on a styled button: Search

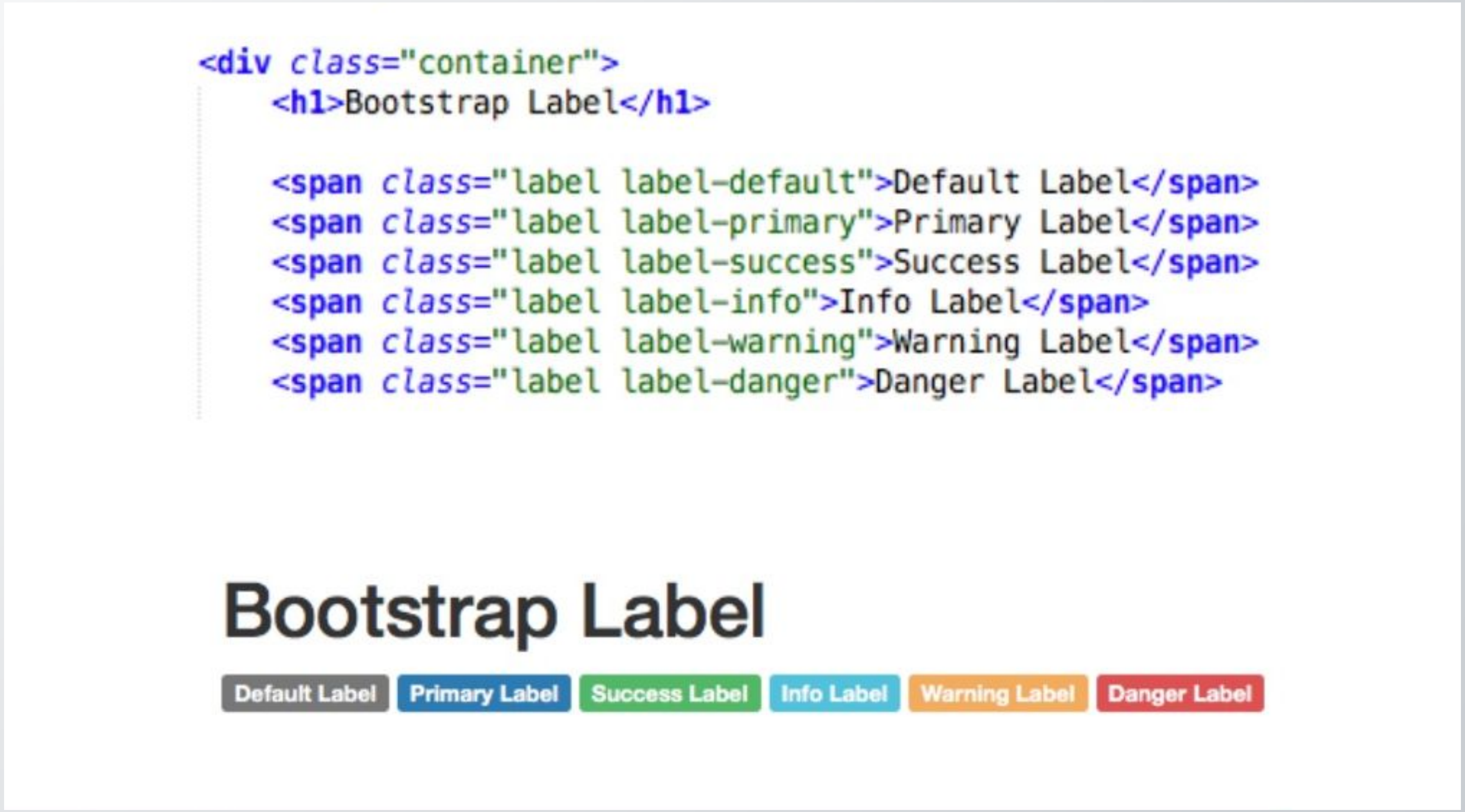
Link with icon: Print

BOOTSTRAP LABELS(BADGES)

- Labels are used to provide information about something.
- Bootstrap create labels with colorful backgrounds to highlight the text inside the label
- Use the .label class, followed by one of the **six contextual classes**



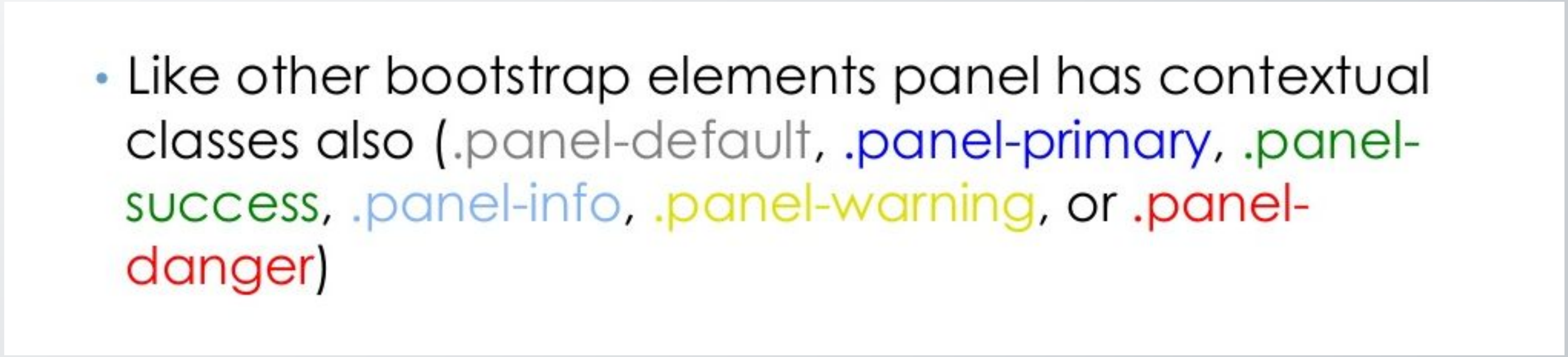
- Use the .label class, followed by one of the six contextual classes .label-default, .label-primary, .label-success, .label-info, .label-warning or .label-danger



```
<div class="container">
    <h1>Bootstrap Label</h1>

    <span class="label label-default">Default Label</span>
    <span class="label label-primary">Primary Label</span>
    <span class="label label-success">Success Label</span>
    <span class="label label-info">Info Label</span>
    <span class="label label-warning">Warning Label</span>
    <span class="label label-danger">Danger Label</span>
```

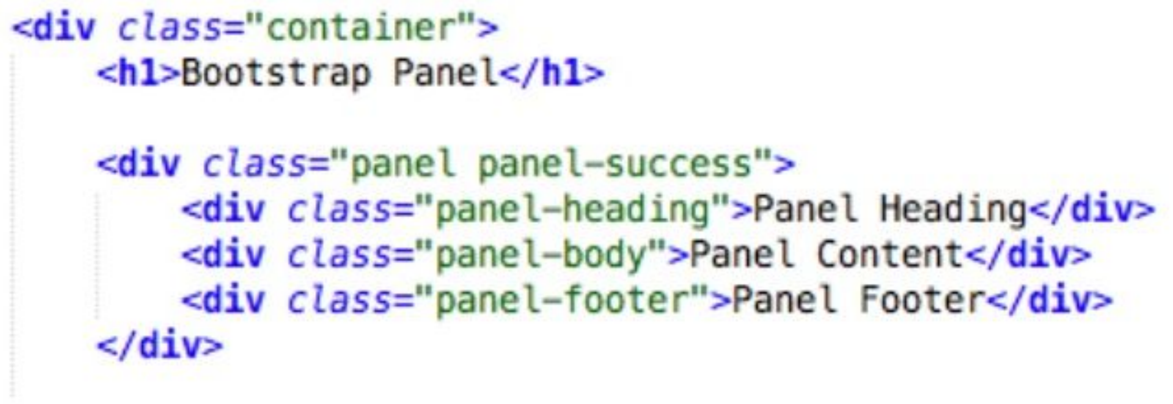
BOOTSTRAP PANELS

- A panel in bootstrap in a bordered box with some padding around its content that can be used to highlight or separate some information.

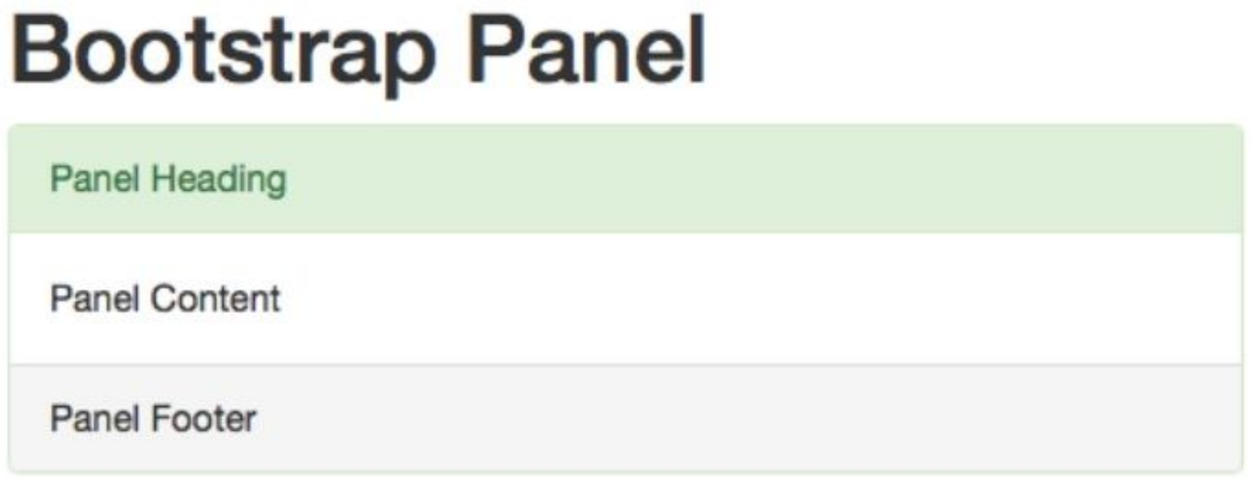


- Like other bootstrap elements panel has contextual classes also (.panel-default, .panel-primary, .panel-success, .panel-info, .panel-warning, or .panel-danger)

Bootstrap Panels



```
<div class="container">
    <h1>Bootstrap Panel</h1>

    <div class="panel panel-success">
        <div class="panel-heading">Panel Heading</div>
        <div class="panel-body">Panel Content</div>
        <div class="panel-footer">Panel Footer</div>
    </div>
```



Bootstrap Panel

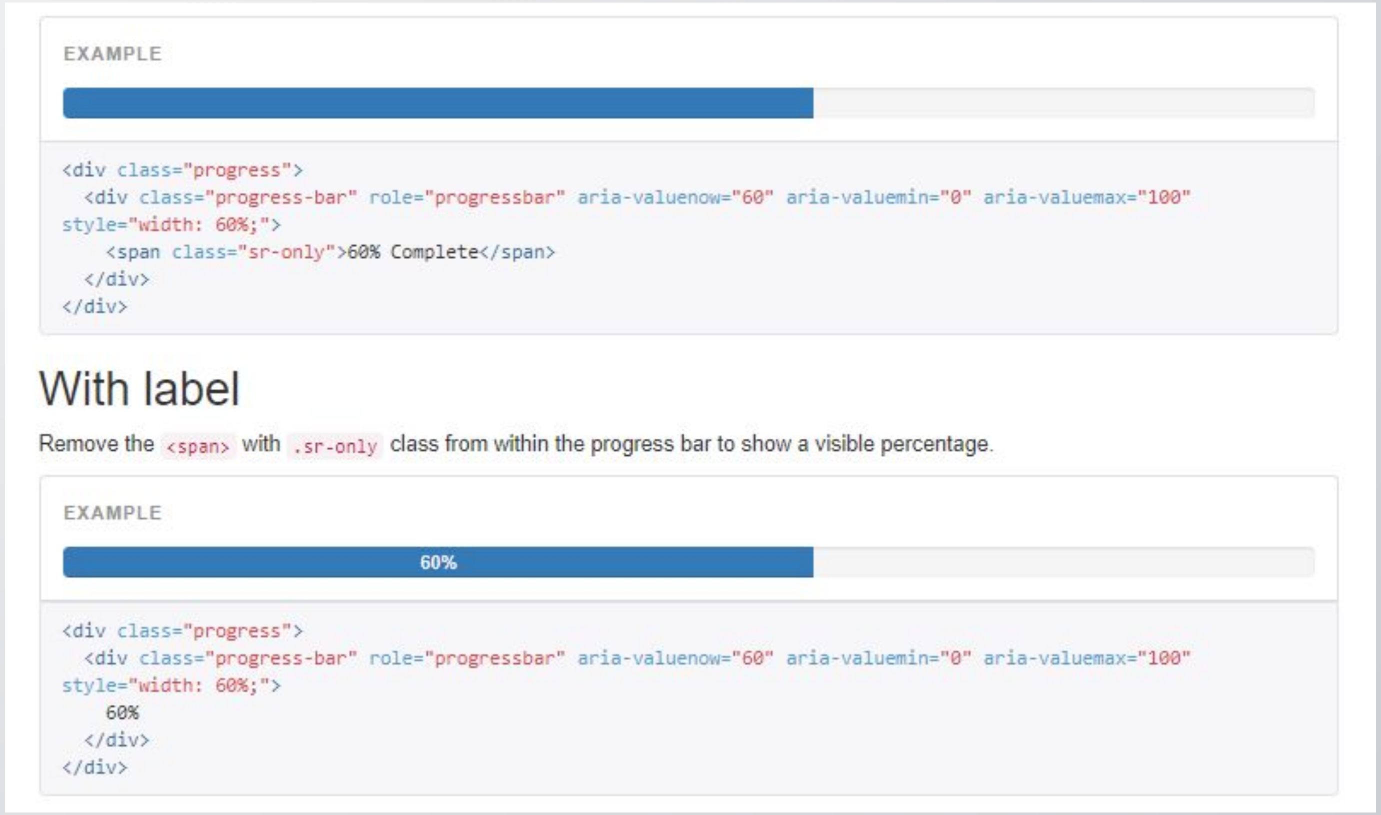
Panel Heading

Panel Content

Panel Footer

BOOTSTRAP PROGRESS BARS

- Provide up-to-date feedback on the progress of a workflow or action with simple yet flexible progress bars.
- Basic example



```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
style="width: 60%;">
    <span class="sr-only">60% Complete</span>
  </div>
</div>
```

With label

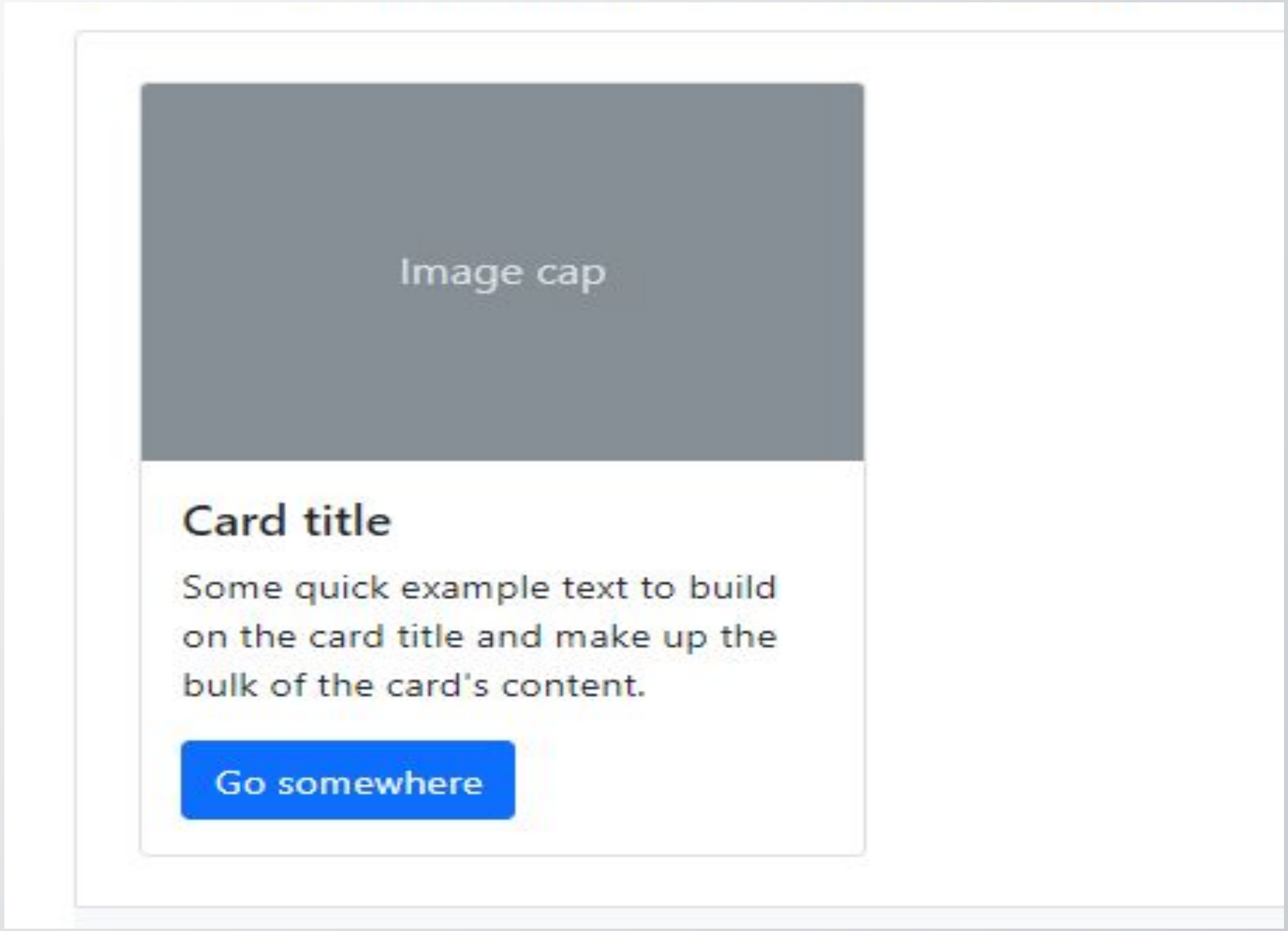
Remove the `<span>` with `.sr-only` class from within the progress bar to show a visible percentage.

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
style="width: 60%;">
    60%
  </div>
</div>
```

BOOTSTRAP CARDS

- Bootstrap's cards provide a flexible and extensible content container with multiple variants and options.

```
<div class="card" style="width: 18rem;">
    <img src="kids.jpg" class="card-img-top" alt="...">
    <div class="card-body">
      <h5 class="card-title">KIDS</h5>
      <p class="card-text">"Children are the hands by which we take hold of heaven."</p>
      <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
  </div>
```



Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

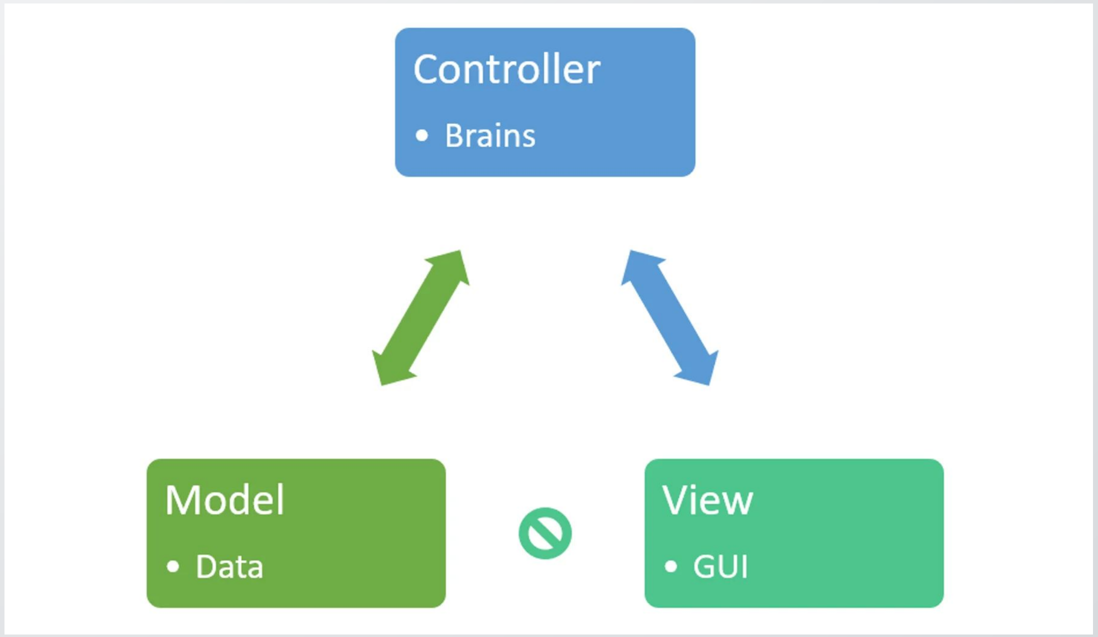
MODEL VIEW CONTROLLER (MVC)

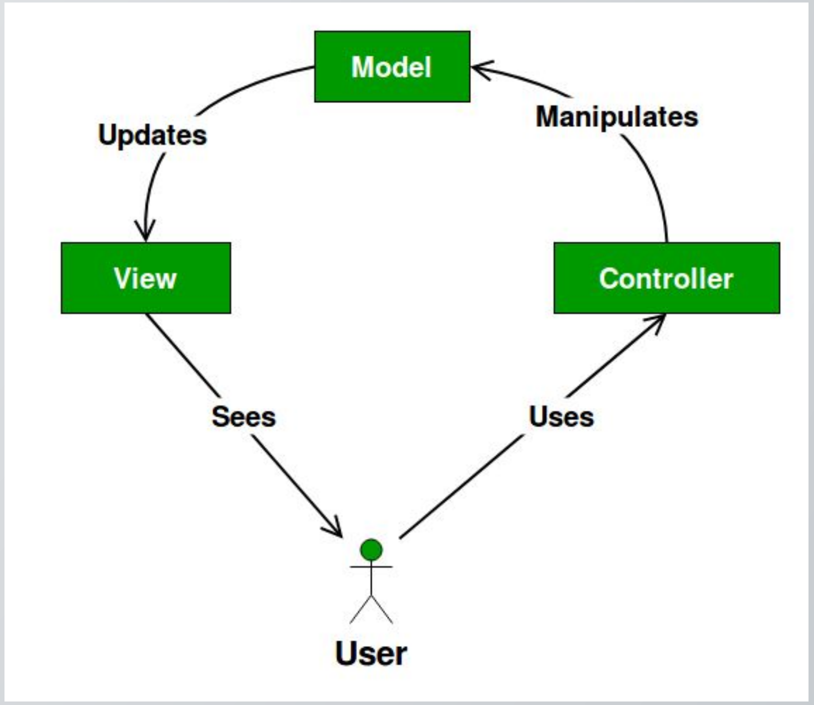
- Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications.

A Model View Controller pattern is made up of the following three parts −

- **Model** − The lowest level of the pattern which is responsible for maintaining data.
- **View** − This is responsible for displaying all or a portion of the data to the user.
- **Controller** − Software Code that controls the interactions between the Model and View.

- The **Model** contains only the pure application data, it contains no logic describing how to present the data to a user.
- The **View** presents the model's data to the user. The view knows how to access the model's data, but it does not know what this data means or what the user can do to manipulate it.
- The **Controller** exists between the view and the model. It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model. Since the view and the model are connected through a notification mechanism, the result of this action is then automatically reflected in the view.


Controller
• Brains
Model
• Data
View
• GUI


Model
Updates
Manipulates
View
Controller
Sees
Uses
User