Name five different Bootstrap components and describe how they work.

1. **Collapse**

The collapse JavaSacript plugin is used to show and hide content. Buttons or anchors are used as triggers that are mapped to specific elements you toggle.

e.g.

<a class="btn btn-primary" data-bs-toggle="collapse" href="#collapseExample" role="button" aria-expanded="false" aria-controls="collapseExample">

Link with href

</a>

<button class="btn btn-primary" type="button" data-bs-toggle="collapse" data-bs-target="#collapseExample" aria-expanded="false" aria-controls="collapseExample">

Button with data-bs-target

</button>

1. **Dropdown**

Dropdowns are toggleable, contextual overlays for displaying lists of links and more. They’re made interactive with the included JavaScript plugin. They’re toggled by cliking, not by hovering.

e.g.

<div class="dropdown">

<button class="btn btn-secondary dropdown-toggle" type="button" data-bs-toggle="dropdown" aria-expanded="false">

Dropdown button

</button>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

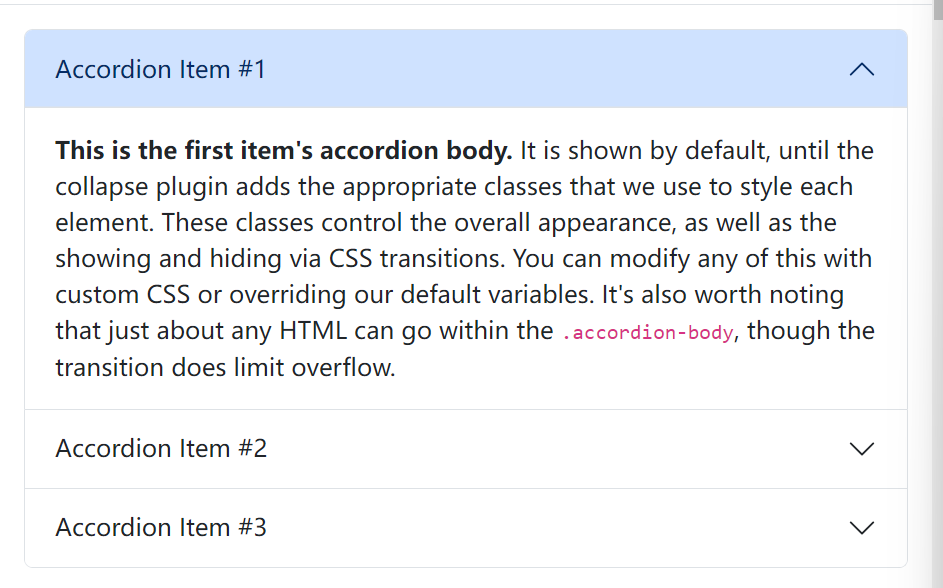
</ul>

</div>

1. **Accordion**

The accordion uses collapse internally to make it collapsible. To render an accordion that’s expanded, add the .open class on the .accordion.

e.g.



<div class="accordion" id="accordionExample">

<div class="accordion-item">

<h2 class="accordion-header">

<button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne">

Accordion Item #1

</button>

</h2>

<div id="collapseOne" class="accordion-collapse collapse show" data-bs-parent="#accordionExample">

<div class="accordion-body">

<strong>This is the first item's accordion body.</strong> It is shown by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.

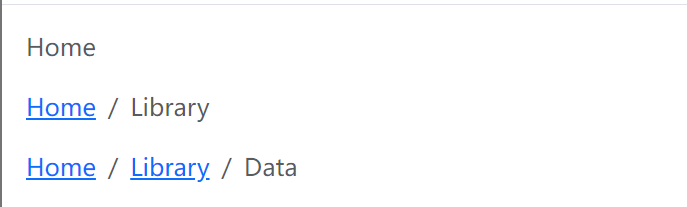
</div>

</div>

</div>

1. **Breadcrumb**

Indicate the current page’s location within a navigational hierarchy that automatically adds separators via CSS



<nav aria-label="breadcrumb">

<ol class="breadcrumb">

<li class="breadcrumb-item"><a href="#">Home</a></li>

<li class="breadcrumb-item active" aria-current="page">Library</li>

</ol>

</nav>

Carousel

A slideshow component for cycling through elements -images or slides of text-like a carousel



<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>

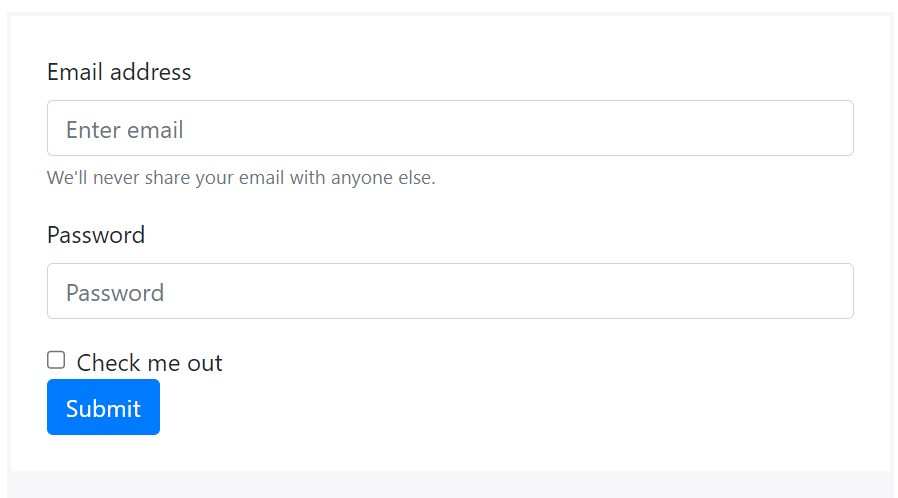
<span class="visually-hidden">Next</span>

</button>

</div>

1. **Form**

Examples and usage guidelines for form control styles, layout options, and cutom components for creating a wide variety of forms.



<form>

<div class="form-group">

<label for="exampleInputEmail1">Email address</label>

<input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email">

<small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>

</div>

<div class="form-group">

<label for="exampleInputPassword1">Password</label>

<input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">

</div>

<div class="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>

2. How does the Bootstrap grid system work?

Bootstrap’s grid system uses a series of containers, rows, and columns to layout and align content. It’s built with flexbox and is fully responsive.

e.g.

<div class=”container”>

<div class=”row”>

<div class=”col-sm”>

One of the three columns

</div>

<div class=”col-sm”>

One of the three columns

</div>

<div class=”col-sm”>

One of the three columns

</div>

</div>

</div>

The above example creates three columns on small, dedium, large, and extra large devices using our predefined grid classes. Those columns are centered in the page with the parent .container.

Container provides a mean to center and horizontally pad your site’s contents. Use . container for a responsive pixel width or .container-fluid for width: 100% across all viewport and device sizes.

Rows are wrappers for columns. Each column has horizontal padding for controlling the space between them. This padding is then counteracted on the rows with negative margins. This way, all the content in your columns is visually aligned down the left side.

In a grid layout, content must be placed within columns and only columns may be immediate children of row.

Thanks to flexbox, grid columns without a specified width will automatically layout as equal width columns. For example, four instances of .col-sm will each automaticall be 25% wide from the small breakpoint and up.

Column classes indicate the number of columns you’d like to use out of the possible 12 per row.

Columns widths are set in percentages, so they’re always fluid and sized releative to their parent element.

Columns have horizontal padding to create the gutters between individual columns, however, you can remove the margin from rows and padding from columns with .no-gutters on the .row

To make the grid responsive, there are five grid breakpoints, one for each responsive breakpoint: all breakpoints (extra small), small, medium, large, and extra large.

Grid breakpoints are based on minimum width media queries, mearning they apply to that one breakpoint and all those above it(e.g., col-sm-4) applies to small, medium, large, and extra large devices, but not the first xs breakpoint).

You can use predefined grid classes (like ,col-4) or Sass mixins for more semative markup