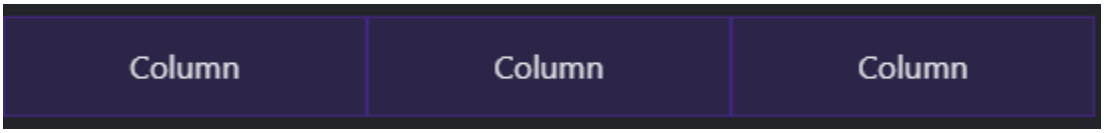
Topics:

Bootstrap Grid System.

Grid system is bootstrap's mobile-first flexbox grid to build layouts of all shapes and sizes thanks to a twelve-column system, six default responsive tiers, Sass variables and mixins, and dozens of predefined classes. Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with a flexbox and is fully responsive. Below is an example and an in-depth explanation of how the grid system comes together.

Example:

```
<div class="container text-center">
  <div class="row">
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
  </div>
</div>
```



The above example creates three equal-width columns across all devices and viewports using our predefined grid classes. Those columns are centered in the page with the parent .container.

How it works.

Our grid supports six responsive breakpoints. Breakpoints are based on min-width media queries, meaning they affect that breakpoint and all those above it (e.g., .col-sm-4 applies to sm, md, lg, xl, and xxl). This means you can control container and column sizing and behavior by each breakpoint.

Containers center and horizontally pad your content. Use .container for a responsive pixel width, .container-fluid for width: 100% across all viewports and devices, or a responsive container (e.g., .container-md) for a combination of fluid and pixel widths.

Rows are wrappers for columns. Each column has horizontal padding (called a gutter) for controlling the space between them. This padding is then counteracted on the rows with

negative margins to ensure the content in your columns is visually aligned down the left side. Rows also support modifier classes to uniformly apply column sizing and gutter classes to change the spacing of your content.

Columns are incredibly flexible. There are 12 template columns available per row, allowing you to create different combinations of elements that span any number of columns. Column classes indicate the number of template columns to span (e.g., col-4 spans four). widths are set in percentages so you always have the same relative sizing.

Gutters are also responsive and customizable. Gutter classes are available across all breakpoints, with all the same sizes as our margin and padding spacing. Change horizontal gutters with .gx-* classes, vertical gutters with .gy-*, or all gutters with .g-* classes. .g-0 is also available to remove gutters.
Grid options
Bootstrap's grid system can adapt across all six default breakpoints, and any breakpoints you customize. The six default grid tiers are as follows:

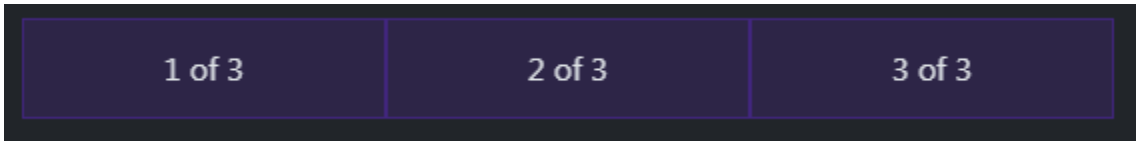
Extra small (xs)
Small (sm)
Medium (md)
Large (lg)
Extra large (xl)
Extra extra large (xxl)
As noted above, each of these breakpoints have their own container, unique class prefix, and modifiers. Here's how the grid changes across these breakpoints:

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥140
Container `max-width`	None (auto)	540px	720px	960px	1140px	1320
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`	`.col-xxl-`
# of columns	12					
Gutter width	1.5rem (.75rem on left and right)					
Custom gutters	Yes					
Nestable	Yes					
Column ordering	Yes					

Auto-layout columns

Utilize breakpoint-specific column classes for easy column sizing without an explicit numbered class like .col-sm-6.

Equal-width
For example, here are two grid layouts that apply to every device and viewport, from xs to xxl. Add any number of unit-less classes for each breakpoint you need and every column will be the same width.




```
<div class="container text-center">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      2 of 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      2 of 3
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

Note: We can initiate columns in rows but not rows in columns

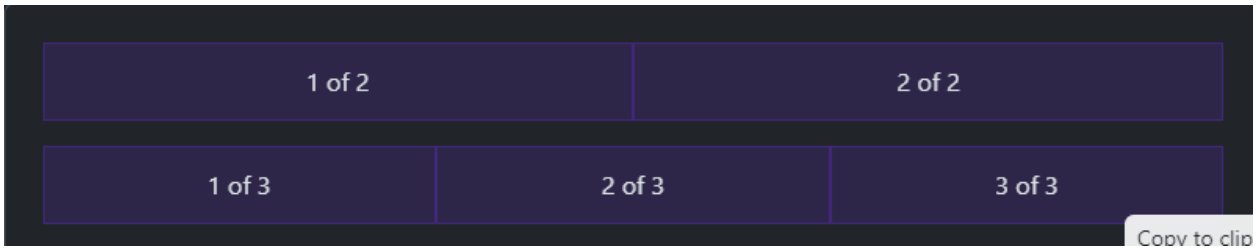
```
<div class="col">
      <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      2 of 3
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
    </div>
```

Equal-width

For example, here are two grid layouts that apply to every device and viewport, from xs to xxl. Add any number of unit-less classes for each breakpoint you need and every column will be the same width.

```
<div class="container text-center">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      2 of 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      2 of 3
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
```

```
</div>
```



Setting one column width

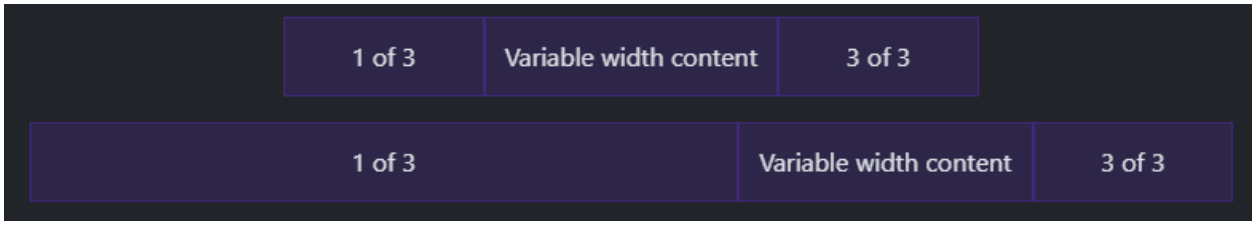
Auto-layout for flexbox grid columns also means you can set the width of one column and have the sibling columns automatically resize around it. You may use predefined grid classes (as shown below), grid mixins, or inline widths. Note that the other columns will resize no matter the width of the center column.

```
<div class="container text-center">
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-6">
      2 of 3 (wider)
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-5">
      2 of 3 (wider)
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

Variable width content

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

```
<div class="container text-center">
  <div class="row justify-content-md-center">
    <div class="col col-lg-2">
      1 of 3
    </div>
    <div class="col-md-auto">
      Variable width content
    </div>
    <div class="col col-lg-2">
      3 of 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-md-auto">
      Variable width content
    </div>
    <div class="col col-lg-2">
      3 of 3
    </div>
  </div>
</div>
```

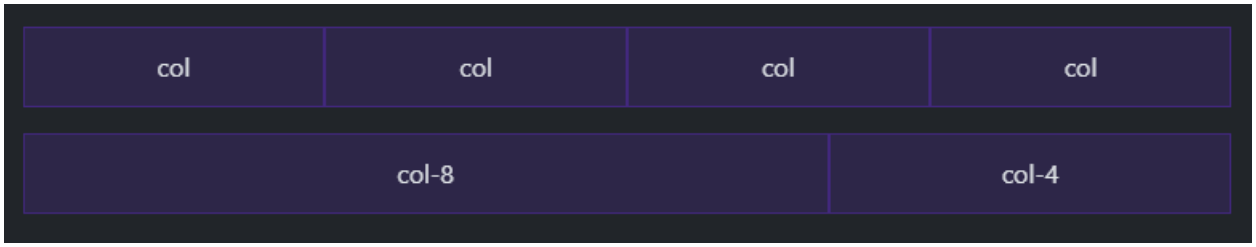


Responsive classes

Bootstrap's grid includes six tiers of predefined classes for building complex responsive layouts. Customize the size of your columns on extra small, small, medium, large, or extra large devices however you see fit.

All breakpoints
For grids that are the same from the smallest of devices to the largest, use the .col and .col-* classes. Specify a numbered class when you need a particularly sized column; otherwise, feel free to stick to .col.

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



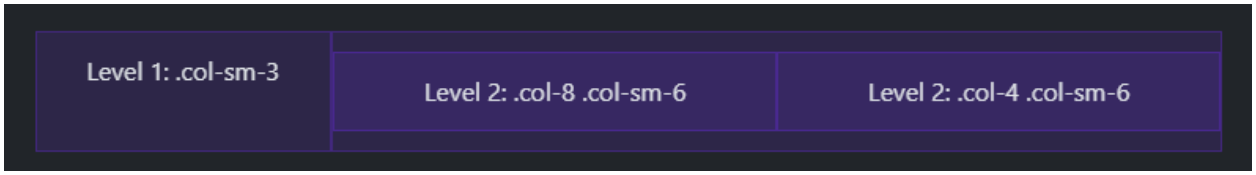
Nesting and offsetting columns

Nesting

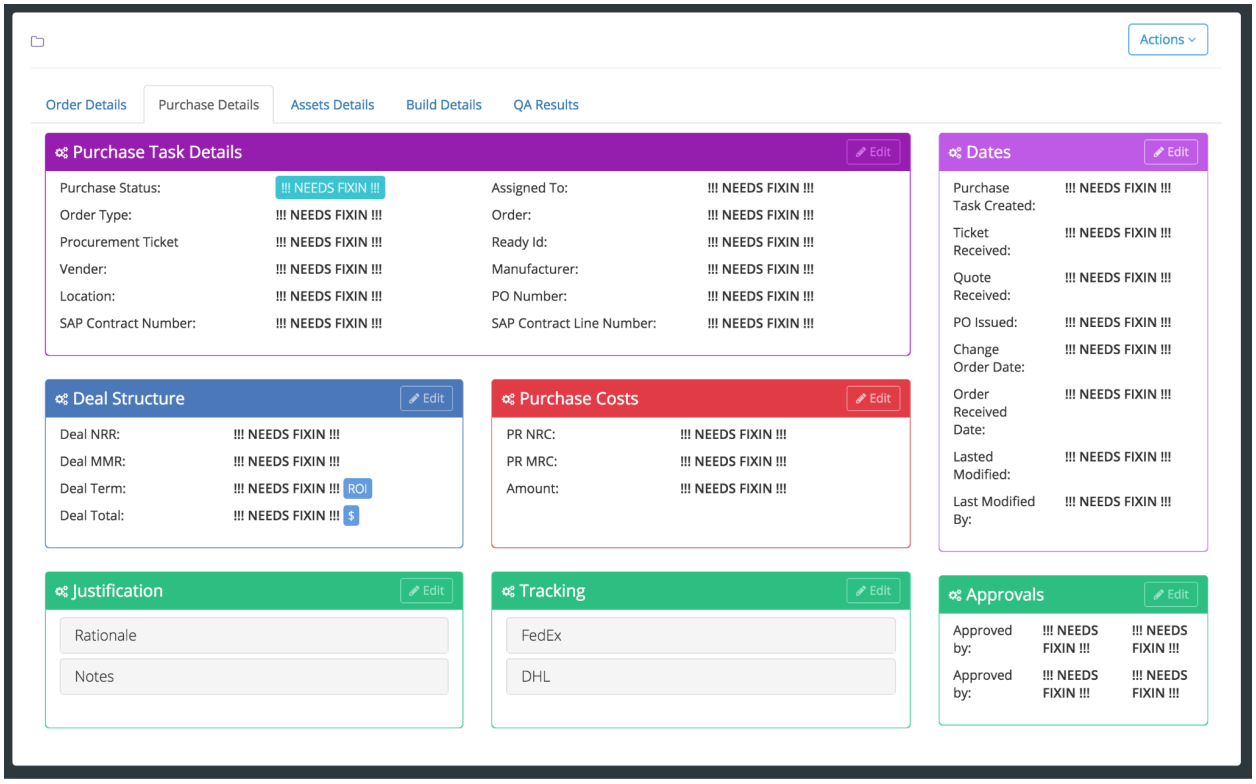
To nest your content with the default grid, add a new .row and set of .col-sm-* columns within an existing .col-sm-* column. Nested rows should include a set of columns that add up to 12 or fewer (it is not required that you use all 12 available columns).

```
<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>
```



Example of bootstrap nesting on a website



Introduction to utility-classes

Changing display

Use our display utilities for responsively toggling common values of the display property. Mix it with our grid system, content, or components to show or hide them across specific viewports.

Flexbox options

Bootstrap is built with flexbox, but not every element's display has been changed to display: flex as this would add many unnecessary overrides and unexpectedly change key browser behaviors. Most of our components are built with flexbox enabled.

Should you need to add display: flex to an element, do so with .d-flex or one of the responsive variants (e.g., .d-sm-flex). You'll need this class or display value to allow the use of our extra flexbox utilities for sizing, alignment, spacing, and more.

Margin and padding

Use the margin and padding spacing utilities to control how elements and components are spaced and sized. Bootstrap includes a six-level scale for spacing utilities, based on a 1rem value default $spacer variable. Choose values for all viewports (e.g., .me-3 for margin-right: 1rem in LTR), or pick responsive variants to target specific viewports (e.g., .me-md-3 for margin-right: 1rem —in LTR— starting at the md breakpoint).

Toggle visibility

When toggling display isn't needed, you can toggle the visibility of an element with our visibility utilities. Invisible elements will still affect the layout of the page, but are visually hidden from visitors.