

# MODULE 4

## WEB DESIGN with CLIENT SIDE SCRIPTING



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

SUB TOPIC 1

# Grid System and Media Queries



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

Use our powerful mobile-first flexbox grid to build layouts of all shapes and sizes thanks to a twelve column system, five default responsive tiers, Sass variables and mixins, and dozens of predefined classes



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

## How it works

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. Below is an example and an in-depth look at how the grid comes together.

New to or unfamiliar with flexbox? Read this [CSS Tricks flexbox guide](#) for background, terminology, guidelines, and code snippets.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

One of three columns

One of three columns

One of three columns

Copy

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

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

Breaking it down, here's how it works:

Containers provide a means 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 (called a gutter) 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.

# Grid System

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

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 automatically be 25% wide from the small breakpoint and up. See the auto-layout columns section for more examples. Column classes indicate the number of columns you'd like to use out of the possible 12 per row. So, if you want three equal-width columns across, you can use `.col-4`.

Column widths are set in percentages, so they're always fluid and sized relative 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`.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

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, meaning 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 semantic markup. Be aware of the limitations and bugs around flexbox, like the inability to use some HTML elements as flex containers.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*



# Grid System

## Grid options

While Bootstrap uses ems or rems for defining most sizes, pxs are used for grid breakpoints and container widths. This is because the viewport width is in pixels and does not change with the font size.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				



# Grid System

## 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 xl. Add any number of unit-less classes for each breakpoint you need and every column will be the same width.

## 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.

# Grid System

Variable width content

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

## Equal-width multi-row

Create equal-width columns that span multiple rows by inserting a `.w-100` where you want the columns to break to a new line. Make the breaks responsive by mixing the `.w-100` with some responsive display utilities.

## Responsive classes

Bootstrap's grid includes five 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`.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

## Stacked to horizontal

Using a single set of `.col-sm-*` classes, you can create a basic grid system that starts out stacked before becoming horizontal with at the small breakpoint (sm).

## Mix and match

Don't want your columns to simply stack in some grid tiers? Use a combination of different classes for each tier as needed. See the example below for a better idea of how it all works.

## Alignment

Use flexbox alignment utilities to vertically and horizontally align columns.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

## No gutters

The gutters between columns in our predefined grid classes can be removed with `.no-gutters`. This removes the negative margins from `.row` and the horizontal padding from all immediate children columns.

Here's the source code for creating these styles. Note that column overrides are scoped to only the first children columns and are targeted via attribute selector. While this generates a more specific selector, column padding can still be further customized with spacing utilities.

Need an edge-to-edge design? Drop the parent `.container` or `.container-fluid`.

# Grid System

```
.no-gutters {  
  margin-right: 0;  
  margin-left: 0;  
  
  > .col,  
  > [class*="col-"] {  
    padding-right: 0;  
    padding-left: 0;  
  }  
}
```



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

## 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. Normally this is accomplished with multiple .rows, but not every implementation method can account for this.

## 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). Includes support for 1 through 12 across all five grid tiers.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*



# Grid System

## Offsetting columns

You can offset grid columns in two ways: our responsive `.offset-` grid classes and our margin utilities. Grid classes are sized to match columns while margins are more useful for quick layouts where the width of the offset is variable.

## Offset classes

Move columns to the right using `.offset-md-*` classes. These classes increase the left margin of a column by `*` columns. For example, `.offset-md-4` moves `.col-md-4` over four columns.

## Margin utilities

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



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*



# Grid System

## 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)

## Sass mixins

When using Bootstrap's source Sass files, you have the option of using Sass variables and mixins to create custom, semantic, and responsive page layouts. Our predefined grid classes use these same variables and mixins to provide a whole suite of ready-to-use classes for fast responsive layouts.

## Variables

Variables and maps determine the number of columns, the gutter width, and the media query point at which to begin floating columns. We use these to generate the predefined grid classes documented above, as well as for the custom mixins listed below.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

## Customizing the grid

Using our built-in grid Sass variables and maps, it's possible to completely customize the predefined grid classes. Change the number of tiers, the media query dimensions, and the container widths—then recompile.

## Columns and gutters

The number of grid columns can be modified via Sass variables. `$grid-columns` is used to generate the widths (in percent) of each individual column while `$grid-gutter-width` allows breakpoint-specific widths that are divided evenly across padding-left and padding-right for the column gutters.

`$grid-columns: 12 !default;`

`$grid-gutter-width: 30px !default;`



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

## Grid tiers

Moving beyond the columns themselves, you may also customize the number of grid tiers. If you wanted just four grid tiers, you'd update the `$grid-breakpoints` and `$container-max-widths` to something like this:

```
$grid-breakpoints: (  
  xs: 0,  
  sm: 480px,  
  md: 768px,  
  lg: 1024px  
);
```

```
$container-max-widths: (  
  sm: 420px,  
  md: 720px,  
  lg: 960px  
);
```

When making any changes to the Sass variables or maps, you'll need to save your changes and recompile. Doing so will output a brand new set of predefined grid classes for column widths, offsets, and ordering. Responsive visibility utilities will also be updated to use the custom breakpoints. Make sure to set grid values in px (not rem, em, or %).



# Media Object

Documentation and examples for Bootstrap's media object to construct highly repetitive components like blog comments, tweets, and the like.

## Example

The media object helps build complex and repetitive components where some media is positioned alongside content that doesn't wrap around said media. Plus, it does this with only two required classes thanks to flexbox.

Below is an example of a single media object. Only two classes are required—the wrapping `.media` and the `.media-body` around your content. Optional padding and margin can be controlled through spacing utilities.

# Media Object

```
<div class="media">  
    
  <div class="media-body">  
    <h5 class="mt-0">Media heading</h5>  
    Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin.  
    Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc  
    ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.  
  </div>  
</div>
```

64x64

## Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Media Object

## Nesting

Media objects can be infinitely nested, though we suggest you stop at some point. Place nested `.media` within the `.media-body` of a parent media object.

64x64

### Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

64x64

### Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Media Object

```
<div class="media">  
    
  <div class="media-body">  
    <h5 class="mt-0">Media heading</h5>  
    Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate  
    at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.
```

```
  <div class="media mt-3">  
    <a class="pr-3" href="#">  
        
    </a>  
    <div class="media-body">  
      <h5 class="mt-0">Media heading</h5>  
      Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate  
      at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.  
    </div>  
  </div>  
</div>  
</div>  
</div>
```



# Media Object

## Alignment

Media in a media object can be aligned with flexbox utilities to the top (default), middle, or end of your .media-body content.



64x64

### Top-aligned media

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*



# Media Object

```
<div class="media">  
    
  <div class="media-body">  
    <h5 class="mt-0">Top-aligned media</h5>  
    <p>Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante  
sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce  
condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.</p>  
    <p>Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis  
natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.</p>  
  </div>  
</div>
```



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Media Object

## Order

Change the order of content in media objects by modifying the HTML itself, or by adding some custom flexbox CSS to set the order property (to an integer of your choosing).

### Media object

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

64x64



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Media Object

```
<div class="media">
```

```
  <div class="media-body">
```

```
    <h5 class="mt-0 mb-1">Media object</h5>
```

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

```
  </div>
```

```
  
```

```
</div>
```

# Media Object

## Media list

Because the media object has so few structural requirements, you can also use these classes on list HTML elements. On your `<ul>` or `<ol>`, add the `.list-unstyled` to remove any browser default list styles, and then apply `.media` to your `<li>`s. As always, use spacing utilities wherever needed to fine tune.



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Media Object

```
<ul class="list-unstyled">
  <li class="media">
    
    <div class="media-body">
      <h5 class="mt-0 mb-1">List-based media
object</h5>
      Cras sit amet nibh libero, in gravida nulla. Nulla
vel metus scelerisque ante sollicitudin. Cras purus
odio, vestibulum in vulputate at, tempus viverra
turpis. Fusce condimentum nunc ac nisi vulputate
fringilla. Donec lacinia congue felis in faucibus.
    </div>
  </li>
  <li class="media my-4">
    
    <div class="media-body">
      <h5 class="mt-0 mb-1">List-based media
object</h5>
```

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

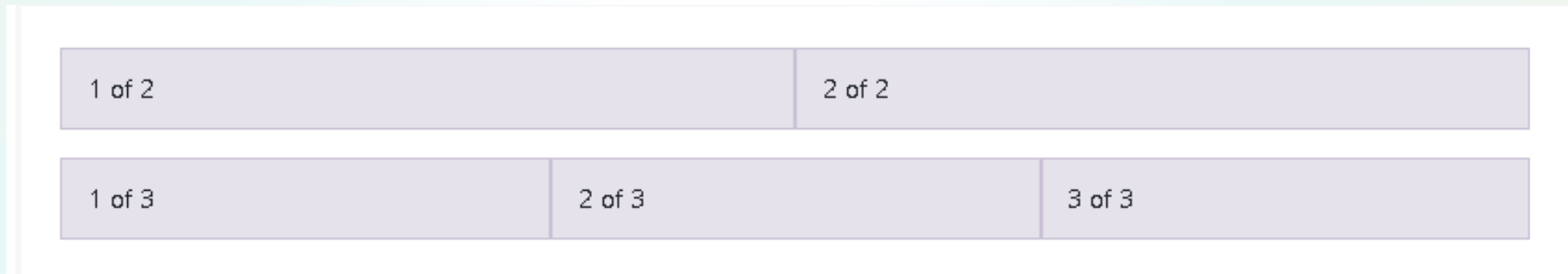
```
</div>
</li>
<li class="media">
  
  <div class="media-body">
    <h5 class="mt-0 mb-1">List-based media
object</h5>
    Cras sit amet nibh libero, in gravida nulla. Nulla
vel metus scelerisque ante sollicitudin. Cras purus
odio, vestibulum in vulputate at, tempus viverra
turpis. Fusce condimentum nunc ac nisi vulputate
fringilla. Donec lacinia congue felis in faucibus.
  </div>
</li>
</ul>
```



# Grid System

## Equal-width

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



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

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

*Technology Driven by Innovation*



FEU ALABANG



FEU DILIMAN



FEU TECH

# Grid System

## 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.

1 of 3	2 of 3 (wider)	3 of 3
1 of 3	2 of 3 (wider)	3 of 3





# Grid System

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

*Technology Driven by Innovation*



FEU ALABANG



FEU DILIMAN



FEU TECH

# Grid System

## Variable width content

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

1 of 3	Variable width content	3 of 3
1 of 3		Variable width content 3 of 3



# Grid System

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



FEU ALABANG



FEU DILIMAN



FEU TECH

*Technology Driven by Innovation*

# Grid System

## Equal-width multi-row

Create equal-width columns that span multiple rows by inserting a `.w-100` where you want the columns to break to a new line. Make the breaks responsive by mixing the `.w-100` with some responsive display utilities.

col	col
col	col

```
<div class="row">
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="w-100"></div>
  <div class="col">col</div>
  <div class="col">col</div>
</div>
```

