**Bootstrap 4.5**

1. **Layout:**
   1. .container, which sets a max-width at each responsive breakpoint.
   2. .container-fluid, which is width:100% at all breakpoints
   3. .container-{breakpoint}, which is width:100% until the specified breakpoint
2. **Grid System**
   1. .col- , for xs | .col-sm- | .col-md- |.col-lg- | .col-xl-
   2. 30px gutter width (15px each side for each col)
   3. .w-100 used where you want the columns to break to a new line in equal width columns.
   4. Break to new line at specific breakpoint using .w-100 .d-none .d-md-block
   5. Use .col-{breakpoint}-auto classes to size columns based on the natural width of their content
   6. .px-5, .px-{breakpoint}-5 padding for x and .py-5, .py-{breakpoint}-5 padding for y
   7. .mx-5, .mx-{breakpoint}-5 margin for x and .my-5, .my-{breakpoint}-5 margin for y
   8. .border for div border
   9. .row-cols-\* or .row-cols-{breakpoint}-\* to quickly set the number of columns for a row.
   10. .align-items-start for top, .align-items-center for center and .align-items-center for bottom used to align item inside row vertically
   11. .align-self-start, .align-self-center and .align-self-end used in column for alignment
   12. .justify-content-start, .justify-content-center, .justify-content-end, .justify-content-around and .justify-content-between used in row to position content
   13. .no-gutters used in row to remove margin in row and padding in columns.
   14. .order-\* ( 1 to 12) used in columns for column ordering.
   15. .order-first and .order-last for first and last column.
   16. .offset-\* or .offset-{breakpoint}-\* used to move \* columns.
   17. .ml-auto or .ml-{breakpoint}-auto to move from left column
   18. .mr-auto or .mr-{breakpoint}-auto to move from right column

**Variable:**

$grid-columns: 12;

$grid-gutter-width:30px;

$grid-breakpoints: (

// Extra small screen / phone

xs: 0,

// Small screen / phone

sm: 576px,

// Medium screen / tablet

md: 768px,

// Large screen / desktop

lg: 992px,

// Extra large screen / wide desktop

xl: 1200px

);

$container-max-widths: (

sm: 540px,

md: 720px,

lg: 960px,

xl: 1140px

);

**Mixins:**

.example-container {

@include make-container();

// Make sure to define this width after the mixin to override

// `width: 100%` generated by `make-container()`

width: 800px;

}

.example-row {

@include make-row();

}

.example-content-main {

@include make-col-ready();

@include media-breakpoint-up(sm) {

@include make-col(6);

}

@include media-breakpoint-up(lg) {

@include make-col(8);

}

}

.example-content-secondary {

@include make-col-ready();

@include media-breakpoint-up(sm) {

@include make-col(6);

}

@include media-breakpoint-up(lg) {

@include make-col(4);

}

}

e.g:

<div class="example-container">

<div class="example-row">

<div class="example-content-main">Main content</div>

<div class="example-content-secondary">Secondary content</div>

</div>

</div>

For 3 breakpoint in container:

$grid-breakpoints: (

xs: 0,

sm: 480px,

md: 768px,

lg: 1024px

);

$container-max-widths: (

sm: 420px,

md: 720px,

lg: 960px

);

1. **Reboot**
   1. Page Defaults: The <html> and <body> elements are updated to provide better page-wide defaults:
      1. The box-sizing is globally set on every element—including \*::before and \*::after, to border-box. This ensures that the declared width of element is never exceeded due to padding or border.
      2. No base font-size is declared on the <html>, but 16px is assumed (the browser default). font-size: 1rem is applied on the <body> for easy responsive type-scaling via media queries while respecting user preferences and ensuring a more accessible approach.
      3. The <body> also sets a global font-family, line-height, and text-align. This is inherited later by some form elements to prevent font inconsistencies.
      4. For safety, the <body> has a declared background-color, defaulting to #fff.
   2. Native font stack:
      1. This font-family is applied to the <body> and automatically inherited globally throughout Bootstrap. To switch the global font-family, update $font-family-base and recompile Bootstrap.
   3. Headings and Paragraphs
      1. All heading elements—e.g., <h1>—and <p> are reset to have their margin-top removed. Headings have margin-bottom: .5rem added and paragraphs margin-bottom: 1rem for easy spacing.
   4. Lists
      1. All lists—<ul>, <ol>, and <dl>—have their margin-top removed and a margin-bottom: 1rem. Nested lists have no margin-bottom.
      2. For simpler styling, clear hierarchy, and better spacing, description lists have updated margins. <dd>s reset margin-left to 0 and add margin-bottom: .5rem. <dt>s are **bolded(font-weight:700)**.
   5. Preformatted text
      1. The <pre> element is reset to remove its margin-top and use rem units for its margin-bottom.
   6. Tables
      1. Tables are slightly adjusted to style <caption>s, collapse borders, and ensure consistent text-align throughout. Additional changes for borders, padding, and more come with [the .table class](https://getbootstrap.com/docs/4.5/content/tables/).
   7. Forms: Various form elements have been rebooted for simpler base styles.
      1. <fieldset>s have no borders, padding, or margin so they can be easily used as wrappers for individual inputs or groups of inputs.
      2. <legend>s, like fieldsets, have also been restyled to be displayed as a heading of sorts.
      3. <label>s are set to display: inline-block to allow margin to be applied.
      4. <input>s, <select>s, <textarea>s, and <button>s are mostly addressed by Normalize, but Reboot removes their margin and sets line-height: inherit, too.
      5. <textarea>s are modified to only be resizable vertically as horizontal resizing often “breaks” page layout.
      6. <button>s and <input> button elements have cursor: pointer when :not(:disabled).
   8. Pointers on buttons
      1. Reboot includes an enhancement for role="button" to change the default cursor to pointer. Add this attribute to elements to help indicate elements are interactive. This role isn’t necessary for <button> elements, which get their own cursor change.
   9. Misc elements
      1. Address
         1. The <address> element is updated to reset the browser default font-style from italic to normal. line-height is also now inherited, and margin-bottom: 1rem has been added. <address>s are for presenting contact information for the nearest ancestor (or an entire body of work). Preserve formatting by ending lines with <br>.
      2. Blockquote
         1. The default margin on blockquotes is 1em 40px, so we reset that to 0 0 1rem for something more consistent with other elements.
      3. Inline elements
         1. The <abbr> element receives basic styling to make it stand out amongst paragraph text.
      4. Summary
         1. The default cursor on summary is text, so we reset that to pointer to convey that the element can be interacted with by clicking on it.
2. **Typography**
   1. Global Settings
      1. Use a [native font stack](https://getbootstrap.com/docs/4.5/content/reboot/#native-font-stack) that selects the best font-family for each OS and device.
      2. For a more inclusive and accessible type scale, we assume the browser default root font-size (typically 16px) so visitors can customize their browser defaults as needed.
      3. Use the $font-family-base, $font-size-base, and $line-height-base attributes as our typographic base applied to the <body>.
      4. Set the global link color via $link-color and apply link underlines only on :hover.
      5. Use $body-bg to set a background-color on the <body> (#fff by default).
      6. These styles can be found within \_reboot.scss, and the global variables are defined in \_variables.scss. Make sure to set $font-size-base in rem.
   2. Headings
      1. .h1 through .h6 classes are also available, for when you want to match the font styling of a heading but cannot use the associated HTML element.
      2. Customizing headings under h tags we can use <small> tag with .text-muted class
   3. Display Headings
      1. When you need a heading to stand out, consider using a **display heading**—a larger, slightly more opinionated heading style. Keep in mind these headings are not responsive by default, but it’s possible to enable [responsive font sizes](https://getbootstrap.com/docs/4.5/content/typography/#responsive-font-sizes).
      2. .display-1, .display-2, .display-3, .display-4 for <h1,h2,h3,h4> tags
   4. Lead
      1. Make a paragraph stand out by adding .lead.
   5. Inline text elements
      1. .mark and .small classes are also available to apply the same styles as <mark> and <small> while avoiding any unwanted semantic implications that the tags would bring.
   6. Abbreviations
      1. Add .initialism to an abbreviation for a slightly smaller font-size.
      2. <abbr></abbr>
   7. Blockquotes
      1. For quoting blocks of content from another source within your document. Wrap <blockquote class="blockquote"> around any HTML as the quote.
      2. Naming a source
         1. Add a <footer class="blockquote-footer"> for identifying the source. Wrap the name of the source work in <cite>.
   8. Lists
      1. Unstyled(.list-unstyled)
         1. Remove the default list-style and left margin on list items (immediate children only). **This only applies to immediate children list items**, meaning you will need to add the class for any nested lists as well.
      2. Inline
         1. Remove a list’s bullets and apply some light margin with a combination of two classes, .list-inline and .list-inline-item.
      3. Description list alignment
         1. Align terms and descriptions horizontally by using our grid system’s predefined classes(.row and .col-\*) (or semantic mixins). For longer terms, you can optionally add a .text-truncate class to truncate the text with an ellipsis.
      4. Responsive font sizes
         1. Bootstrap v4.3 ships with the option to enable responsive font sizes, allowing text to scale more naturally across device and viewport sizes. RFS can be enabled by changing the $enable-responsive-font-sizes Sass variable to true and recompiling Bootstrap.
3. **Code**
   1. Inline Code
      1. Wrap inline snippets of code with <code>. Be sure to escape HTML angle brackets.
   2. Code Block
      1. Use <pre>s for multiple lines of code. Once again, be sure to escape any angle brackets in the code for proper rendering. You may optionally add the .pre-scrollable class, which will set a max-height of 340px and provide a y-axis scrollbar.
   3. Variables
      1. For indicating variables use the <var> tag.
   4. User Input
      1. Use the <kbd> to indicate input that is typically entered via keyboard.
   5. Sample Output
      1. For indicating sample output from a program use the <samp> tag.
4. **Images**
   1. Responsive Images
      1. Images in Bootstrap are made responsive with .img-fluid. max-width: 100%; and height: auto; are applied to the image so that it scales with the parent element.
   2. Image Thumbnails
      1. In addition to our [border-radius utilities](https://getbootstrap.com/docs/4.5/utilities/borders/), you can use .img-thumbnail to give an image a rounded 1px border appearance.
   3. Aligning Images
      1. Align images with the [helper float classes](https://getbootstrap.com/docs/4.5/utilities/float/) or [text alignment classes](https://getbootstrap.com/docs/4.5/utilities/text/#text-alignment).
      2. block-level images can be centered using [the .mx-auto margin utility class](https://getbootstrap.com/docs/4.5/utilities/spacing/#horizontal-centering).
      3. .rounded class for making border round
   4. Picture
      1. If you are using the <picture> element to specify multiple <source> elements for a specific <img>, make sure to add the .img-\* classes to the <img> and not to the <picture> tag.
5. **Tables**
   1. Using the most basic table markup, here’s how .table-based tables look in Bootstrap.
   2. You can also invert the colors—with light text on dark backgrounds—with .table-dark.
   3. Similar to tables and dark tables, use the modifier classes .thead-light or .thead-dark to make <thead>s appear light or dark gray.
   4. Use .table-striped to add zebra-striping to any table row within the <tbody>.
   5. Add .table-bordered for borders on all sides of the table and cells.
   6. Add .table-borderless for a table without borders.
   7. Add .table-hover to enable a hover state on table rows within a <tbody>.
   8. Add .table-sm to make tables more compact by cutting cell padding in half.
   9. Use contextual classes(.table-\*) to color table rows or individual cells.
   10. A <caption> functions like a heading for a table. It helps users with screen readers to find a table and understand what it’s about and decide if they want to read it.
   11. Responsive tables allow tables to be scrolled horizontally with ease. Make any table responsive across all viewports by wrapping a .table with .table-responsive. Or, pick a maximum breakpoint with which to have a responsive table up to by using .table-responsive{-sm|-md|-lg|-xl}.
   12. Scope=”col” for <th> and scope=”row” for first column row used in <td>