Module 1: Introduction to Bootstrap

* What is Bootstrap?
* Advantages of using Bootstrap
* Downloading and setting up Bootstrap in a project
* Exploring the Bootstrap documentation

Module 2: Basic Bootstrap Components

* Grid system: Containers, rows, and columns
* Typography: Headings, paragraphs, and text styling
* Buttons: Creating different types of buttons
* Forms: Building forms with Bootstrap styles
* Tables: Styling and enhancing tables with Bootstrap

Module 3: Bootstrap CSS Components

* Navigation bars: Creating responsive navigation menus
* Alerts: Displaying informational or error messages
* Badges: Adding badges to elements
* Cards: Building flexible and responsive card layouts
* Modals: Creating pop-up modal dialogs
* Carousel: Building image sliders

Module 4: Responsive Design with Bootstrap

* Responsive breakpoints and classes
* Hidden and visible elements at different screen sizes
* Responsive images and videos
* Creating responsive layouts using the grid system
* Mobile-first design principles

Module 5: Bootstrap JavaScript Components

* Bootstrap's JavaScript plugins and their usage
* Dropdowns: Adding dropdown menus to your website
* Tabs: Creating tabbed content
* Accordion: Building collapsible content sections
* Tooltips and popovers: Adding interactive tooltips and popovers
* Scrollspy: Highlighting navigation based on scroll position

Module 6: Customization and Extending Bootstrap

* Customizing Bootstrap with Sass or CSS variables
* Overriding Bootstrap's default styles
* Creating custom Bootstrap themes
* Exploring third-party Bootstrap themes and templates
* Using Bootstrap with other front-end frameworks or libraries

Module 7: Best Practices and Advanced Topics

* Optimizing Bootstrap for performance
* Accessibility considerations with Bootstrap
* Bootstrap and responsive web design patterns
* Integrating Bootstrap with a back-end framework (e.g., Ruby on Rails, Django)
* Building a complete Bootstrap-based project

1.1 What is Bootstrap?

Bootstrap is a popular front-end framework that provides a collection of CSS and JavaScript components for building responsive and mobile-first websites. It offers a range of pre-built design elements and utilities that simplify the development process and ensure consistency across different browsers and devices.

1.2 Advantages of using Bootstrap

Bootstrap offers several advantages for web development:

Time-saving: Bootstrap provides ready-to-use components and styles, reducing the need for custom coding.

Responsive design: Bootstrap's grid system and responsive classes make it easy to create websites that adapt to different screen sizes.

Cross-browser compatibility: Bootstrap is designed to work well across various browsers, ensuring consistent appearance and functionality.

Customization options: Bootstrap allows developers to customize its components, styles, and themes according to their project requirements.

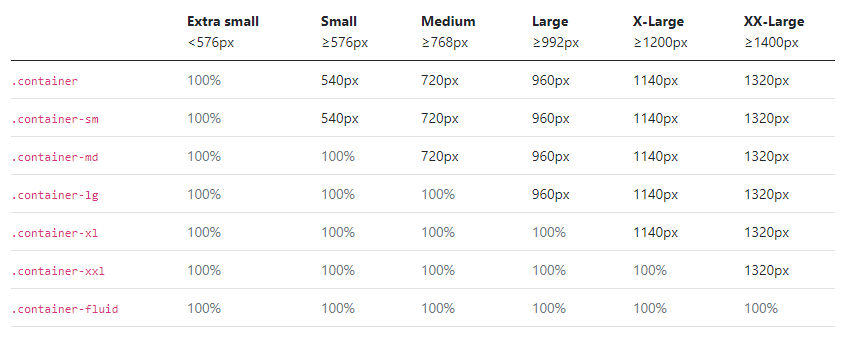
Community and support: Bootstrap has a large user community and extensive documentation, making it easier to find help and resources.

1.3 Downloading and setting up Bootstrap in a project

To use Bootstrap, you can download the CSS and JavaScript files from the official website (getbootstrap.com) or include them via a content delivery network (CDN). The Bootstrap files can be added to your project's directory structure, and you can link to them in your HTML files to start utilizing Bootstrap's features.

Grid system: Containers, rows, and columns

The Bootstrap grid system is a fundamental part of building responsive layouts. It uses a 12-column grid layout, allowing you to divide your page into responsive columns. Containers provide a fixed-width container for your content, while rows are used to create horizontal groups of columns. Columns are the building blocks where you place your content and adjust their widths based on the desired layout.



Bootstrap Grid

A screenshot of a computer

Description automatically generated with medium confidence

**Useful Classes**

.container

.container-fluid

.row

.col-sm-1, .col-md-1, .col-lg-1, .col-xl-1 (.col-sm-1 to .col-sm-12)

.bg-primary, .bg-secondary, .bg-warning, .bg-success and so on

.ps-0, .ps-1, .ps-2, .ps-3, .ps-4, .ps-5 (pe, pt, pl, px, py) (mx, my, mt, mb, ms, me)

.h1 to .h6 and .display-1 to .display-6

.table, .table-responsive, .table-dark, .table-striped, .table-hover

.btn, .btn-primary, .btn-outline-primary, .btn-sm, .btn-lg

.alert, .alert-success, .alert-danger ....

.badge, .rounded-pill

.card, .card-header, .card-img-top, .card-body, .card-title, .card-subtitle, .card-text, .card-footer

.form, .form-group, .form-control, .form-control-file

.modal, .modal-dialog, .modal-content, .modal-header, .modal-body, .modal-footer

.navbar, .navbar-light, .navbar-brand, .navbar-collapse, .navbar nav, .nav-item, .nav-link

.dropdown, .dropdown-menu, .dropdown-item

.rounded, .rounded-top, .rounded-end, .rounded-bottom, .rounded-start, .rounded-circle, .rounded-pill, .rounded-circle,

.border, .border-primary, border-1 ...

.text-primary, .text-secondary, .text-success …

.text-white, .text-dark, .text-muted

.text-start, .text-justify, .text-end, .text-center

.text-uppercase, .text-lowercase, .text-capitalize

.d-flex, .flex-row, .flex-column, .flex-nowrap, .flex-wrap

.justify-content-start, .justify-content-end, .justify-content-center, .justify-content-around, .justify-content-between, .justify-content-evenly (similar classes available for align-content)

.align-items-center, .align-items-start, .align-items-end

.img-fluid, .img-thumbnail

.shadow-sm, .shadow, .shadow-lg

.list-unstyled

Bootstrap Unit:

A screenshot of a computer

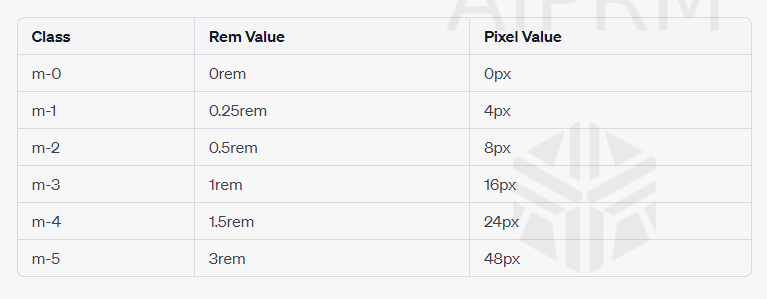
Description automatically generated with medium confidence

As you can see, Bootstrap 1 is equivalent to 1px. This is the default unit for Bootstrap, and it is used for most margin and padding settings. Other Bootstrap units, such as rem and em, are used for more complex layouts.

Here is a brief explanation of each unit:

* px - This is the most basic unit, and it represents a single pixel.
* rem - This unit is relative to the font size of the element. For example, if the font size of an element is 16px, then 1rem will be equal to 16px.
* em - This unit is also relative to the font size of the element, but it is based on the current font size of the element, rather than the default font size. For example, if the font size of an element is 24px, then 1em will be equal to 24px.

**Bootstrap unit with rem and pixel**



**Color in bootstrap**

A picture containing text, screenshot, rectangle, font

Description automatically generated

**Bootstrap Flex  
  
align-items-baseline:** This value aligns the flex items based on their baselines within the container. The baseline is an imaginary line upon which the text of the flex items sits. When align-items-baseline is applied, the flex items will be aligned so that their baselines are at the same level. This means that if the flex items contain text or inline elements with varying heights, the items will be aligned based on the baseline of their content.

**align-items-start:** This value aligns the flex items at the top of the container. When align-items-start is applied, the flex items will be aligned vertically at the top edge of the container. The items will be positioned at their natural height, without any vertical stretching or alignment based on baselines.

In summary, the main difference between align-items-baseline and align-items-start is:

**align-items-baseline** aligns flex items based on their baselines, which is useful when dealing with text or inline elements with varying heights.

**align-items-start aligns** flex items at the top of the container without any vertical stretching or alignment based on baselines.

You can choose the appropriate value based on your specific layout requirements. If you want the flex items to align based on their content's baselines, use align-items-baseline. If you want the flex items to be aligned at the top of the container, use align-items-start.