
Session 12

Mohamed Emary

April 18, 2024

1 Bootstrap Cont

Bootstrap contains:

- **Components** - Predefined components that can be used in the web page like buttons, forms, etc.
- **Utility classes** - Classes that can be used to apply some styles to the elements.
- **Grid system** - A system that helps to create responsive web pages.

1.1 Screen Sizes

Bootstrap has the following screen sizes:

- Screen Sizes:
 - Extra small $< 576\text{px}$
 - Small **sm** $\geq 576\text{px}$
 - Medium **md** $\geq 768\text{px}$
 - Large **lg** $\geq 992\text{px}$
 - Extra large **xl** $\geq 1200\text{px}$
 - Extra extra large **xxl** $\geq 1400\text{px}$

1.2 Containers

- **container** - Container class which has a **max-width** depending on the screen size, it also has a right and left padding.
- **container-fluid** - Container class which has a **width** of 100% in all screen sizes, it still has a right and left padding.
- **container-{SCREEN-SIZE}** - Container class which has a **max-width** depending on the screen size, it also has a right and left padding. ex: **container-sm**, **container-md**, **container-lg**, **container-xl**, **container-xxl**.

- `container-lg` - has a 100% width in extra small, small, and medium screens, and a fixed width in large and above.

2 Flex Display

- `d-flex` - Display flex
- `row` - It has a flex display with `flex-wrap` set to `wrap`, it also has a negative margin to counter the padding of the container and if you remove the container you will notice a horizontal scroll bar.

If you want to test the negative margin try the following code:

```
1 | <div class="negMargin">
2 | </div>
1 | .negMargin {
2 |   height: 100px;
3 |   background-color: teal;
4 |   margin: 0 -15px;
5 | }
```

You will see a horizontal scroll bar no matter how wide your screen is. Try to remove the margin or make it positive and it will disappear.

Now try putting it inside a container that has a padding like the following:

```
1 | <div class="container">
2 |   <div class="negMargin"></div>
3 | </div>
1 | .container {
2 |   background-color: gold;
3 |   padding: 15px;
4 | }
```

And give the container a padding:

The horizontal scroll bar will disappear and you will see the padding of the container only from upper and lower sides since the negative margin is only on the left and right sides.

It will look like this:



Figure 1: Negative Margin

And if you give the container a padding larger than the negative margin (for example 25px) it will look like this:



Figure 2: Side Padding

`row` also allows us to specify the number of items in the row using the classes `col-{NUMBER-OF-ROWS}` followed by the number of columns you want to divide the row into. You can also specify them based on each screen size, for example, `col-sm-6` will make the item take half the width of the row in small screens.

Bootstrap grid system has 12 columns so you can divide the row into 12 columns.

By default if you specify the number of columns an element inside a container spans in a specific screen size, the larger screen sizes will also have the same number of columns and smaller screen sizes will have 12 columns.

For example if your element has the class `col-lg-4` it will span 4 columns in large, extra large, and extra extra large screens, and 12 columns in medium, small, and extra small screens.

The `row` must be the direct parent of the items that have the `col-` class.

`row` has a default right and left padding (gutter) for all its direct children. If you want to change it you can use `gx-{NUMBER}` The number should be between 0 and 5.

With `row` `gy-` is used for vertical **margin** not padding.

So in summary:

- Gutter is used with `row`.
- `gx-{NUMBER}` is used for horizontal **padding** (it's not margin because margin would have caused the elements to go to next line and `box-sizing: border-box` will not be able to fix it).
- `gy-{NUMBER}` is used for vertical **margin**.
- `g-{NUMBER}` allows you to change both `gx-` and `gy-` at the same time.

When using `row` you can make all the elements inside the row have the same width by using the class `col` without specifying the number of columns. You can also specify at which screen size you want the element to have the same width by adding the screen size to the class like `col-lg` so this will make the element have the same width in large screens and above then you can use `col-sm-6` to make the element take half the width in small and medium screens, and the extra small screen will have 12 columns.

2.1 Offset

`offset` is used to give the elements a `margin-left` that is equal to a number of columns you specify. For example, `offset-2` will give the element a margin-left equal to the width of 2 columns.

`offset` is used when you have extra columns space in the row.

You can also make that offset appear in a specific screen size by adding the screen size to the class like `offset-md-2` so this will make the offset appear only in medium screens or larger.

2.2 Min and Max Width

`min-width` is the minimum width that an element can have, the element can be larger than it but not smaller.

`max-width` is the maximum width that an element can have, the element can be smaller than it but not larger.

3 Summary

- **Containers:**
 - `container` - has a `max-width` depending on the screen size.
 - `container-fluid` - has a `width` of 100% in all screen sizes.
 - `container-{SCREEN-SIZE}` - has a `max-width` in that screen size or larger and 100% in smaller screen sizes.
- **row & Flex Display:**
 - `d-flex` - Display flex.
 - `row` - It has a flex display with `flex-wrap` set to `wrap`.
 - `row` has a negative margin to counter the padding of the container.
 - `col-{NUMBER-OF-COLUMNS}` - Specify the number of columns an element will span in a row. **-Gutter**
 - `gx-{NUMBER}` - Change the horizontal padding of items inside the row.
 - `gy-{NUMBER}` - Change the vertical margin of items inside the row.
 - `g-{NUMBER}` - Change both the horizontal padding and vertical margin of items inside the row.
- **col**
 - `col` - Make all the elements inside the row have the same width.
 - `col-{SCREEN-SIZE}` - Make all the elements inside the row have the same width in that screen size or larger.
- **offset**
 - `offset-{NUMBER-OF-COLUMNS}` - Give the element a `margin-left` equal to the number of columns you specify.
 - `offset-{SCREEN-SIZE}-{NUMBER-OF-COLUMNS}` - Give the element a `margin-left` equal to the number of columns you specify in that screen size or larger.