Unit 04: Responsive Design

CONTENTS

Objectives

After studying this unit, you will be able to:

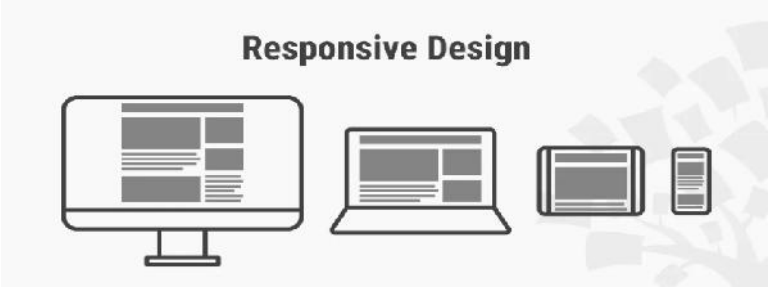
- Understand Responsive Design
- Understand Grid System
- Analyze distinct flex grid in bootstrap

Introduction

A graphic user interface (GUI) design strategy called responsive design is used to produce content that fluidly adapts to different screen sizes. To ensure content consistency across devices, designers apply media queries and size elements in relative units (%) so that their designs can automatically adjust to the browser space.Responsive design allows your site to resize to fit any screen. Rather than designing around a specific layout, or designing a separate mobile and desktop site, your site should scale to fit any device. In the real world, people will use your site in all kinds of different ways; there are small phones, big phones, tablets, tablets in landscape mode, netbooks, laptops, and 5K iMacs.

4.1 Responsive Design

Through code, responsive web design automatically adapts the layout to various screen sizes and resolutions. It is what enables consumers to view a web page smoothly whether they are using a large desktop display or a small mobile device. On a mobile device, for instance, a huge website header with a few columns will be divided into smaller blocks and stacked vertically to make it simpler for the user to read the content and view the visuals.



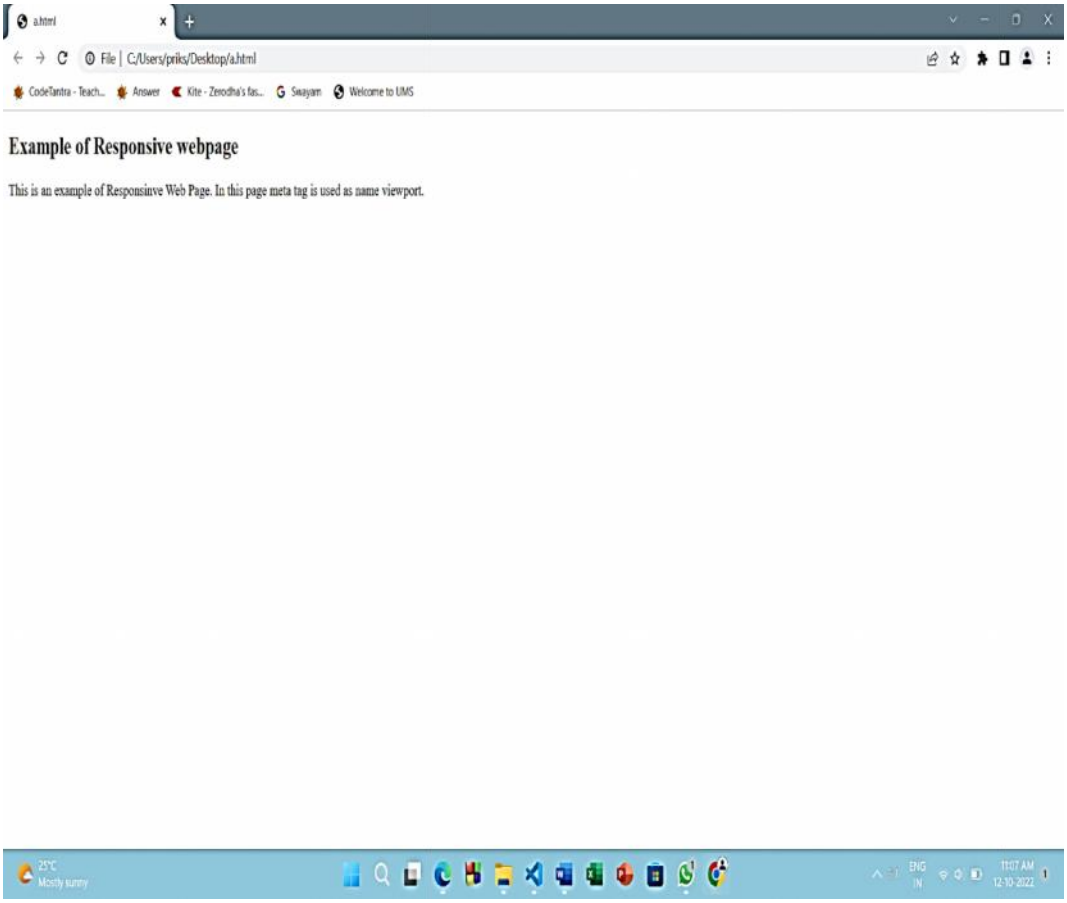
Notes: To create a responsive website, add the following <meta> tag to all your web pages:

<meta name="viewport" content="width=device-width, initial-scale=1.0">

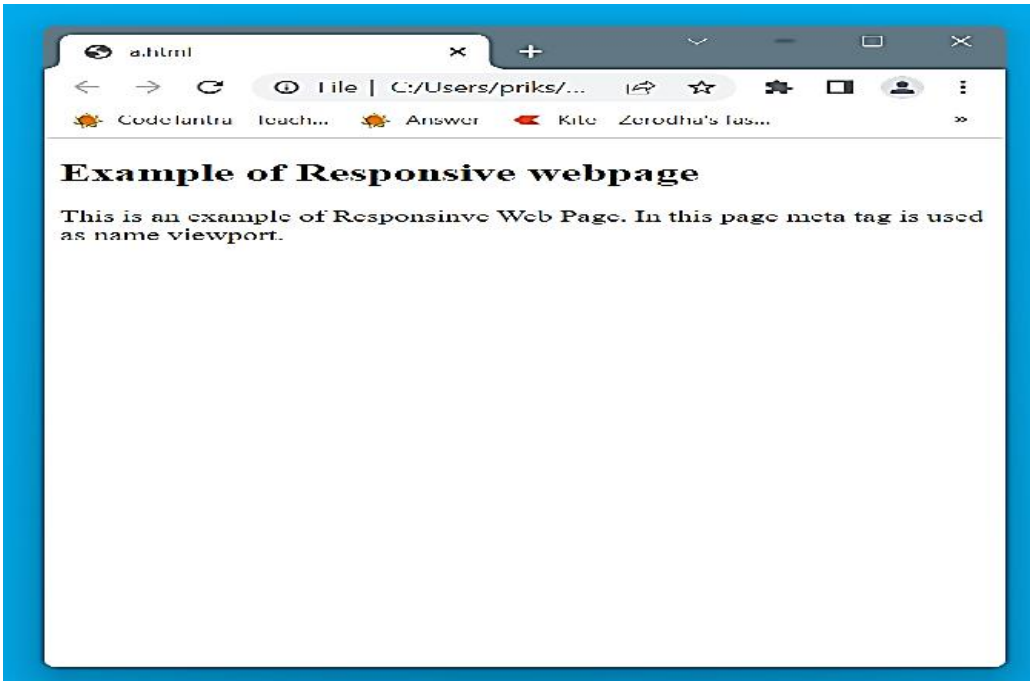
Example: <!DOCTYPE html>

```
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<h2>Example of Responsive webpage</h2>
<p> This is an example of Responsinve Web Page. In this page meta tag is used as name viewport.</p>
</body>
</html>
```

Output 1 – Full Browser



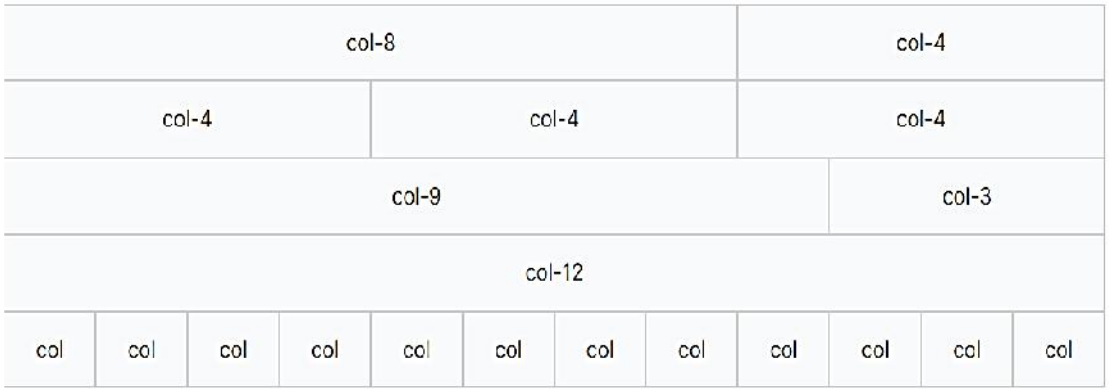
Output 2 – Output resized according to browser width



4.2 Grid System

The grid system in Bootstrap 4 is created with CSS Flexbox and divided into a maximum of twelve column-like segments on a webpage. But there are also situations where a webpage does not require a 12-column structure, then column grouping, that is, a grid structure, can be created in many other ways.

Here's an example that shows the different multi-column grid systems that can be created in Bootstrap 4:



Notes: Bootstrap 4 grid is a 2D structure of intersecting lines, which can be both vertical and horizontal, and is used to structure the content.

Bootstrap 4 classes of Grid

Class Name	Description
col-	It is for other small devices where the screen size is less than 576px.
col-sm-	It is for small devices with a screen size equal to or greater than 576px.
col-md-	It is for medium devices with a screen width equal to or greater than 768px.
col-lg-	It is for large devices with a screen width equal to or greater than 992px.
col-xl-	It is for extra-large devices with a screen width equal to or greater than 1200px.

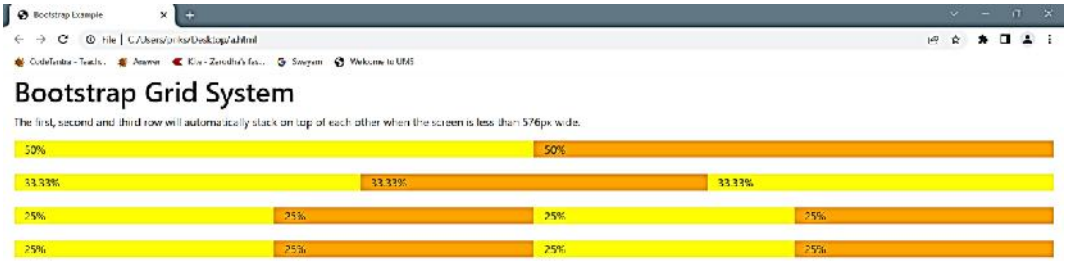
Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
```

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<div class="container-fluid">
<h1>Bootstrap Grid System</h1>

<p>The first, second and third row will automatically stack on top of each other when the screen is less than 576px wide.</p>

<div class="container-fluid">

<div class="row">
<div class="col-sm-6" style="background-color:yellow;">50%</div>
<div class="col-sm-6" style="background-color:orange;">50%</div>
</div>
<br>

<div class="row">
<div class="col-sm-4" style="background-color:yellow;">33.33%</div>
<div class="col-sm-4" style="background-color:orange;">33.33%</div>
<div class="col-sm-4" style="background-color:yellow;">33.33%</div>
</div>
<br>

<!-- Or let Bootstrap automatically handle the layout -->
<div class="row">
<div class="col-sm" style="background-color:yellow;">25%</div>
<div class="col-sm" style="background-color:orange;">25%</div>
<div class="col-sm" style="background-color:yellow;">25%</div>
<div class="col-sm" style="background-color:orange;">25%</div>
</div>
<br>
<div class="row">
```

```
<div class="col" style="background-color:yellow;">25%</div>
<div class="col" style="background-color:orange;">25%</div>
<div class="col" style="background-color:yellow;">25%</div>
<div class="col" style="background-color:orange;">25%</div>
</div>
</div>
</div>

</body>
</html>
```

Output –



4.3 Using Flex

With a complete set of responsive flexbox tools, you can easily control the layout, alignment, and scaling of grid columns, navigation, components, and more. Custom CSS can be required for implementations that are more complicated.

Apply display utilities to create a flexbox container and transform direct children elements into flex items. Flex containers and items are able to be modified further with additional flex properties.

Example:

```
<div class="d-flex p-2">I'm a flexbox container! </div>
```

Responsive variations also exist for .d-flex and .d-inline-flex.

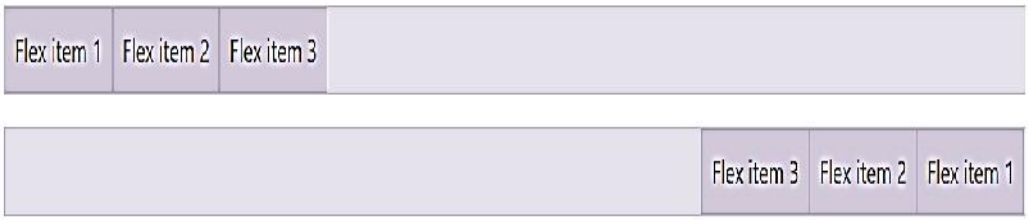
- .d-flex
- .d-inline-flex
- .d-sm-flex
- .d-sm-inline-flex
- .d-md-flex
- .d-md-inline-flex
- .d-lg-flex
- .d-lg-inline-flex
- .d-xl-flex
- .d-xl-inline-flex

Direction

Set the direction of flex items in a flex container with direction utilities. In most cases you can omit the horizontal class here as the browser default is row. However, you may encounter situations where you needed to explicitly set this value (like responsive layouts).

Use .flex-row to set a horizontal direction (the browser default), or .flex-row-reverse to start the horizontal direction from the opposite side.

Example




```
<div class="d-flex flex-row">
<div class="p-2">Flex item 1</div>
<div class="p-2">Flex item 2</div>
<div class="p-2">Flex item 3</div>
</div>
<div class="d-flex flex-row-reverse">
<div class="p-2">Flex item 1</div>
<div class="p-2">Flex item 2</div>
<div class="p-2">Flex item 3</div>
</div>
```

Responsive variations also exist for flex-direction.

- .flex-row
- .flex-row-reverse
- .flex-column

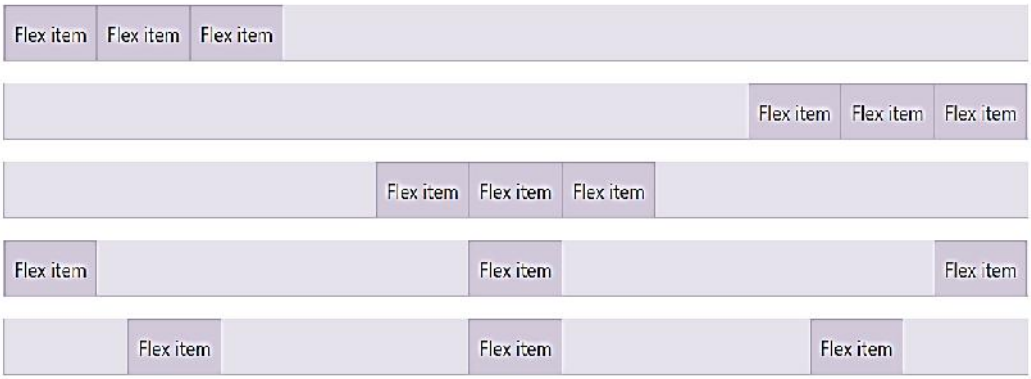
- .flex-column-reverse
- .flex-sm-row
- .flex-sm-row-reverse
- .flex-sm-column
- .flex-sm-column-reverse
- .flex-md-row
- .flex-md-row-reverse
- .flex-md-column
- .flex-md-column-reverse
- .flex-lg-row
- .flex-lg-row-reverse
- .flex-lg-column
- .flex-lg-column-reverse
- .flex-xl-row
- .flex-xl-row-reverse
- .flex-xl-column
- .flex-xl-column-reverse

Justify content

Use justify-content utilities on flexbox containers to change the alignment of flex items on the main axis (the x-axis to start, y-axis if flex-direction: column). Choose from start (browser default), end, center, between, or around.

Example:

```
<div class="d-flex justify-content-start">...</div>
<div class="d-flex justify-content-end">...</div>
<div class="d-flex justify-content-center">...</div>
<div class="d-flex justify-content-between">...</div>
<div class="d-flex justify-content-around">...</div>
```



Responsive variations also exist for justify content.

- .justify-content-start
- .justify-content-end

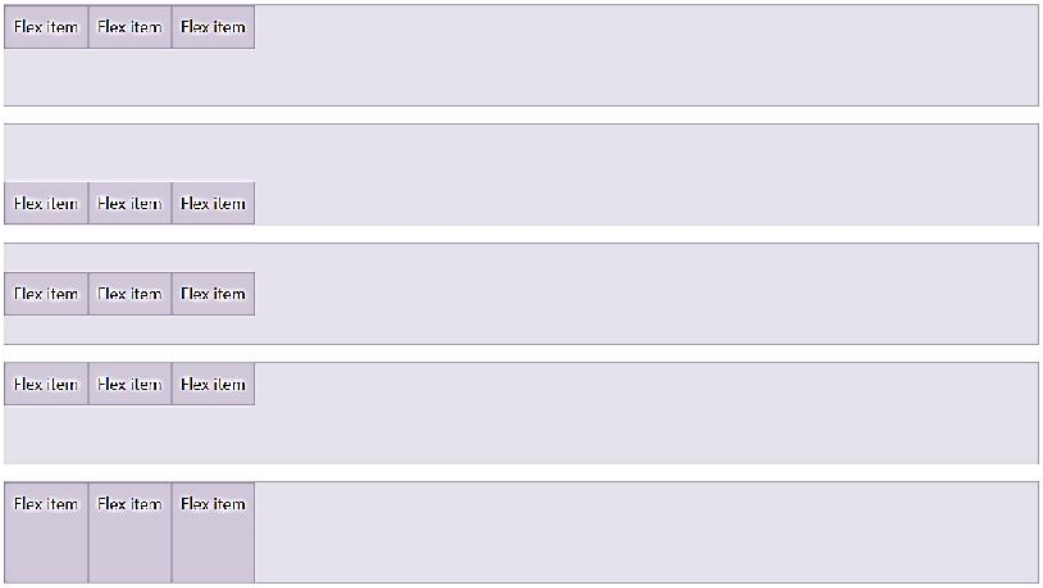
- .justify-content-center
- .justify-content-between
- .justify-content-around
- .justify-content-sm-start
- .justify-content-sm-end
- .justify-content-sm-center
- .justify-content-sm-between
- .justify-content-sm-around
- .justify-content-md-start
- .justify-content-md-end
- .justify-content-md-center
- .justify-content-md-between
- .justify-content-md-around
- .justify-content-lg-start
- .justify-content-lg-end
- .justify-content-lg-center
- .justify-content-lg-between
- .justify-content-lg-around
- .justify-content-xl-start
- .justify-content-xl-end
- .justify-content-xl-center
- .justify-content-xl-between
- .justify-content-xl-around

Align items

Use align-items utilities on flexbox containers to change the alignment of flex items on the cross axis (the y-axis to start, x-axis if flex-direction: column). Choose from start, end, center, baseline, or stretch (browser default).

 **Example:**

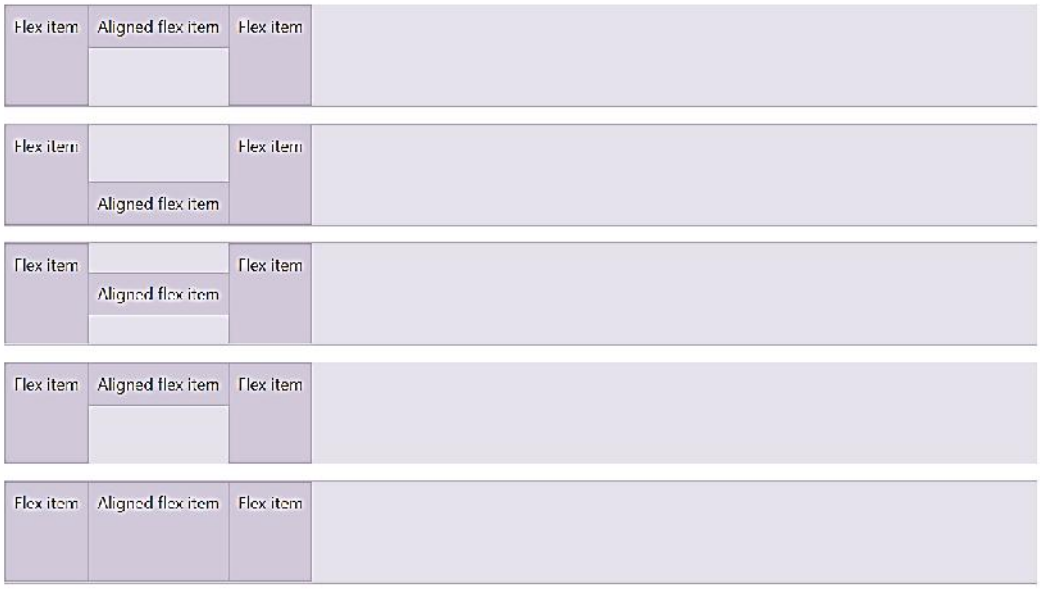
```
<div class="d-flex align-items-start">...</div>
<div class="d-flex align-items-end">...</div>
<div class="d-flex align-items-center">...</div>
<div class="d-flex align-items-baseline">...</div>
<div class="d-flex align-items-stretch">...</div>
```



Align self

Use align-self utilities on flexbox items to individually change their alignment on the cross axis (the y-axis to start, x-axis if flex-direction: column). Choose from the same options as align-items: start, end, center, baseline, or stretch (browser default).

Example




```
<div class="align-self-start">Aligned flex item</div>
<div class="align-self-end">Aligned flex item</div>
<div class="align-self-center">Aligned flex item</div>
<div class="align-self-baseline">Aligned flex item</div>
<div class="align-self-stretch">Aligned flex item</div>
```

Responsive variations also exist for align-self.

- .align-self-start
- .align-self-end
- .align-self-center
- .align-self-baseline
- .align-self-stretch
- .align-self-sm-start
- .align-self-sm-end
- .align-self-sm-center
- .align-self-sm-baseline
- .align-self-sm-stretch
- .align-self-md-start
- .align-self-md-end
- .align-self-md-center
- .align-self-md-baseline
- .align-self-md-stretch
- .align-self-lg-start
- .align-self-lg-end
- .align-self-lg-center

Summary

- Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing a responsive and mobile-friendly website.
- It is absolutely free to download and use.
- It is a front-end framework used for easier and faster web development.
- It facilitates you to create responsive designs.

Keywords

Responsive Website:A website is called a responsive website which can automatically adjust itself to look good on all devices, from smartphones to desktops etc.

Scaffolding: Bootstrap provides a basic structure with Grid System, link styles, and background.

CSS: Bootstrap comes with the feature of global CSS settings, fundamental HTML elements style and an advanced grid system.

Components: Bootstrap contains a lot of reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.

JavaScript Plugins: Bootstrap also contains a lot of custom jQuery plugins. You can easily include them all, or one by one.

SelfAssessment

1. Which of the following class in Bootstrap is used to provide a responsive fixed width container?

A. .container-fixed

B. .container-fluid

C. .container

D. All of the above

2. Bootstrap includes a responsive, mobile-first fluid grid system that appropriately scales up to _____ as the device or viewport size increases.

A. 9 columns

B. 10 columns

C. 11 columns

D. 12 columns

3. Which of the following is correct about the Bootstrap Grid System?

A. Rows must be placed within a .container class for proper alignment and padding.
B. Use rows to create horizontal groups of columns.
C. Content should be placed within the columns, and only columns may be the immediate children of rows.
D. All of the above.

4. Which of the following is correct about the Bootstrap Grid System?

A. Predefined grid classes like .row and .col-xs-4 are available for quickly making grid layouts. Fewer mixins can also be used for more semantic layouts.
B. Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and the last column via negative margin on .rows.
C. Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three .col-xs-4.
D. All of the above.

5. Responsive variations also exist for align-self.

A. True

B. False

6. What are the features of bootstrap?

A. Mobile first approach

B. Easy to get started

C. Browser Support

D. All of the above

7. Bootstrap package includes?

A. Scaffolding

B. Components

C. JavaScript Plugins

D. All of the above

8. Which of these elements in HTML can be used for making a text bold?

A. <a>

B. <pre>

C.

D. <b>

9. Which tag is used in HTML5 for the initialization of the document type?

A. <Doctype HTML>

B. <!DOCTYPE html>

C. <Doctype>

D. <\Doctype html>

10. Which of the following statements is/are correct about Bootstrap 4?

A. It provides a free front - end framework

B. It is a completely open - source product

C. It gives the ability to create responsive designs

D. All of the above

11. What does the following meta tag do?

```
<META NAME="VIEWPORT" CONTENT="WIDTH=DEVICE-WIDTH, INITIAL-SCALE=1">
```

A. Tells the browser to look for media queries in the CSS file

B. Makes the web page responsive automatically

C. Prevents browsers from scaling desktop websites on mobile

D. None of the above options are correct

12. Which of the following is not included in the framework?

A. Forms

B. Font Size

C. Button Style

D. Css

13. Bootstrap was first introduced in _____.

A. Google

B. Facebook

C. Twitter

D. Snapchat

14. Which is not a tool to build responsive websites?

A. skeleton

B. bookmarklets

C. code editors

D. web browsers

15. How many classes bootstrap 4 grid system has?

A. 1

B. 3

C. 5

D. 7

Answers for SelfAssessment

l.	C	2.	D	3.	D	4.	D	5.	A
6.	D	7.	D	8.	D	9.	B	10.	D
11.	C	12.	A	13.	C	14.	A	15.	C

Review Questions

1. Create a responsive web page that include following HTML tags.
    - <P>
    - <Table>
2. What do you mean by flex system in bootstrap? Explain using example.
3. When we need responsive design?
4. Design a web page that demonstrate working of grid system
5. Develop a web page using Align self-flex grid.

Further Readings

Bootstrap by Example, Silvio Morato

Learning Bootstrap 4, Matt Lambart

Bootstrap in 24 Hours, By Sams

Jump Start Bootstrap,

Web Links

https://getbootstrap.com/docs/4.1/layout/overview/

https://getbootstrap.com/docs/4.1/layout/grid/