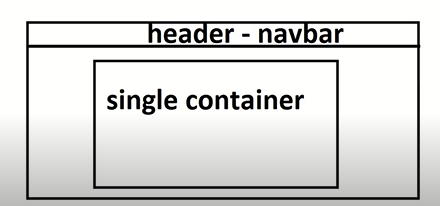
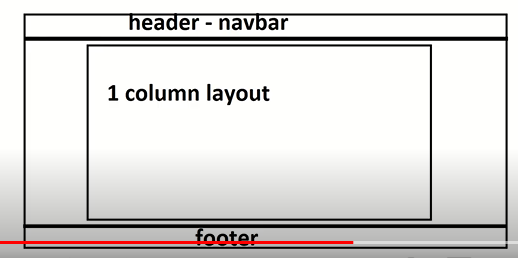
1. Angular Component
   1. Layout:

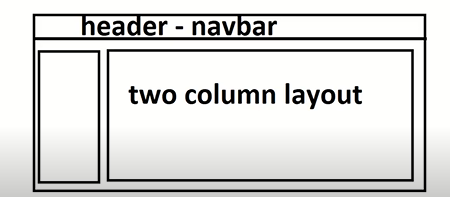
Single header/single container/one column template



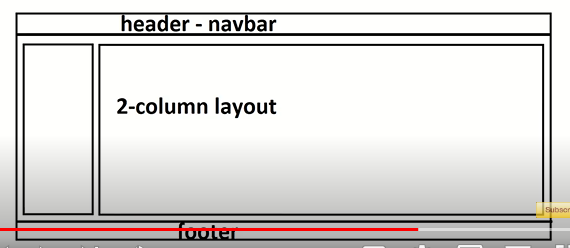
One column layout and footer



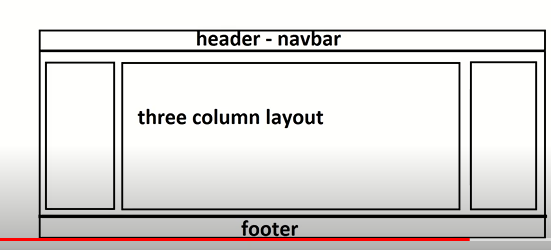
Two column layout: header, sidebar, main container with two column layout



Two column layout and footer



Header, three column layout, footer



2. Bootstrap classes -

- Container

- Margin

- Padding

3. Bootstrap Grid - col-6 (bootstrap layoout)

3.1. Bootstrap Grid

Bootstrap Grid allows us to design responsive layouts easily

The grid is designed in a way to adapt to different screan resolutions

The bootstrap grid is by default 12 column on any screen size: containers, row, column

The bootstrap grid supports-large, medium, small and extra small resolutions

We refer by using col-6 or col-md-4,…

4. Bootstrap Navbar

- navbar-brand

- navbar-item

- navbar-link

- navbar-item dropdown

Bootstrap navbar demo with real time project coding

<https://www.youtube.com/watch?v=TIEhI6RZzvY&t=34s>

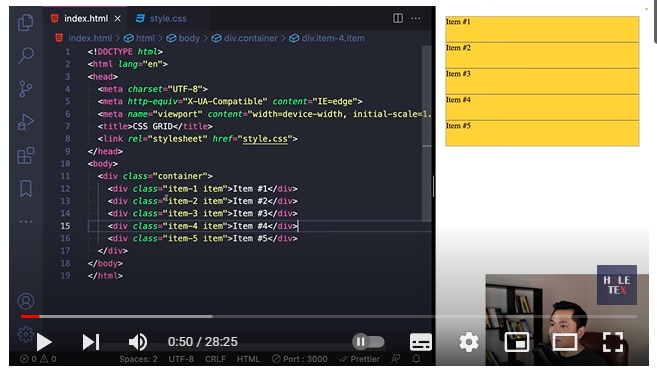
1. **CÁC Class của hệ thống lưới trong bootstrap (bao gồm 5 classes)**

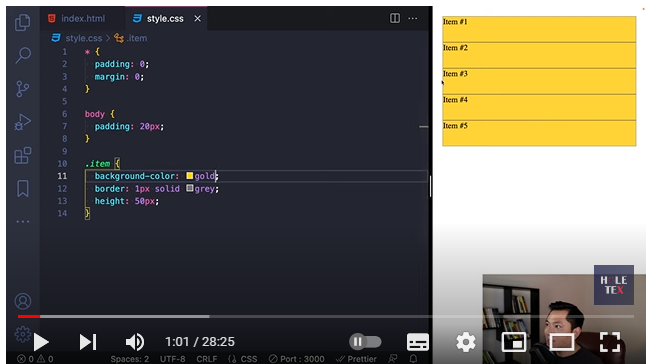
* Col-: dành cho thiết bị cực nhỏ (extra small) có hộ rộng màn hình nhỏ hơn 576px.
* Col-sm-: dành cho các thiết bị nhỏ (small) có độ rộng màn hình bằng hoặc lớn hơn 576px.
* Col-md-: dành cho các thiết bị trung bình (medium) có độ rộng màn hình bằng hoặc lớn hơn 768px.
* Col-lg-: dành cho các thiết bị lớn (large) có độ rộng màn hình bằng hoặc lớn hơn 992px.
* Col-xl-: dành cho các thiết bị cực lớn (extra large) độ rộng màn hình bằng hoặc lớn hơn 1200px.

1. **Các quy tắc khi sử dụng hệ thống lưới trong bootstrap**

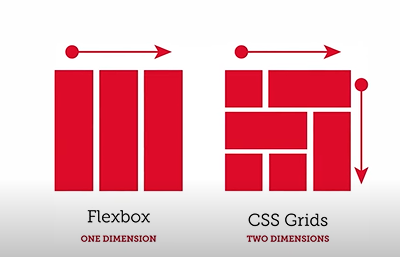
* Để căn chỉnh và padding cho phù hợp thì các hành phải được đặt trong một **.contanier** or **container-liquid**.
* Sử dụng các hàng để tạo nhóm cột ngang (class .row)
* Các cột là con của các hàng, nội dung nên đặt trong các cột.
* Các lớp .row và col-sm-4 phải có sẵn để tạo bố cục lưới nhanh chóng.
* Tạo khoảng cách giữa nội dung cột thông quan padding.
* Cột lưới được tạo ra bằng cách chỉ định số lượng cột trong tổng số 12 cột mà ta muốn tạo.
* Độ rộng cột được tính theo tỷ lệ % nên khá linh hoạt, và kích thước của chúng sẽ tương ứng với phần tử cha.
* Bootstrap 4 được xây dựng với flexbox, nên các cột lưới khi không được chỉ định chiều rộng sẽ tự động bố cục như là các cột có chiều rộng và cao bằng nhau.

**CSS GRID (https://www.youtube.com/watch?v=hJHQVpv6-Z8)**



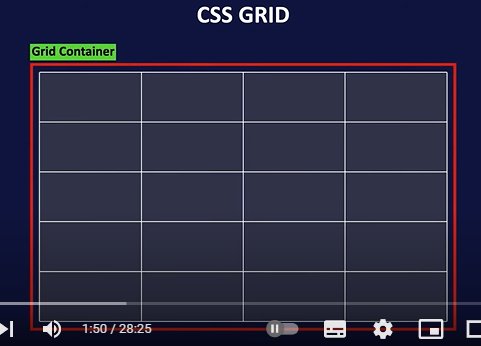


Mô tả css:

* Body có padding = 20px là khoảng trắng trống xung quanh thẻ container (thẻ div này với màn hình).
* Borrder: 1px solid grey; tạo đường viền giữa các thẻ div
* Height: 50px. /nmin-height: 50px (chiều cao sẽ tăng theo nội dung bên trong của thẻ)
* 
* 

CSS grid sẽ bao gồm các thành phần:

* Grid container: khi chúng ta sét 1 thuộc tính là **"display: grid”** cho bất kỳ element nào thì nó sẽ trở thành 1 grid container

****

* Grid area là 1 hình chữ nhật, nó sẽ chiếm bao nhiêu ô trong grid container
* Css cho grid container:

.container{

**Display: grid ;**

**Grid-template-column: 300px 100px 50px/1fr 2fr 1fr/repeat (3, 1fr)** (chia theo tỷ lệ số fr); (quy định bao nhiêu hàng bao nhiêu cột và bao nhiêu hàng ví dụ dụ có 2 cột, cột 1 có 300px, cột 2 100px).

**Grid-template-rows: 100px; 50px;**

}

Note: ngoài độ dài tuyệt đối (vd 50px) cũng có độ dài tương đối: fr (vd: 1fr)

.item-5{

   grid-column-start: 2;

   grid-column-end: 4;

   grid-row-start: 2;

   grid-row-end: 4;

}

Phân bổ chia layout cho item (ví dụ: thẻ div(item-5) bắt đầu từ cộ thứ 2 và kết thúc ở cột thứ 4

Hoặc viết gắn gọn:

.item-5{

   grid-column: 2/4;

grid-row: 2/4;

}

<!DOCTYPE html>

<html lang="en">

<head>

</head>

<body>

<div class="container">

    <div class="item-2 item">sidebar #2</div>

    <div class="item-3 item">footer #3</div>

    <div class="item-4 item">content #4 </div>

    <div class="item-5 item">header #5</div>

</div>

</body>

<footer></footer>

</html>

\* {

    padding: 0;

    margin: 0;

}

body{

    padding: 372px;

}

.item{

    background-color: rgb(200, 200, 106);

    border: 1px solid grey;

    min-height: 50px;

    width: 100%;

}

.container{

    display: grid;

    /\* grid-template-columns: 300px 100px 100px;

    grid-template-rows: 100px 200px 50px; \*/

    grid-auto-rows: minmax(60px,auto);

    grid-template-areas:

    "header header header"

    "sidebar content content"

    "sidebar content content"

    "footer footer footer";

    min-height: 50vh;

}

.item-2{

    grid-area: sidebar;

}

.item-5{

   /\* grid-column-start: 2;

   grid-column-end: 4;

   grid-row-start: 2;

   grid-row-end: 4; \*/

   grid-area: header;

}

.item-4{

    grid-area: content;

}

.item-3{

    grid-area: footer;

}

FLEX BOX

<!DOCTYPE html>

<html lang="en">

<head>

</head>

<body>

   <div class="container">

    <div class="box box-1">

        <p>Box 1</p>

        <p>Flex item</p>

    </div>

    <div class="box box-2">

        <p>Box 2</p>

        <p>Flex item</p>

    </div>

    <div class="box box-3">

        <p>Box 3</p>

        <p>Flex item</p>

    </div>

   </div>

</body>

</html>

\*{

    padding: 0;

    margin: 0;

    box-sizing: border-box;

}

.container{

    width:100vw ;

    height: 800px;

    border: 5px solid grey;

}

.box{

    height: 200px;

    padding: 10px;

}

.box p {

    font-size: 20px;

    color: white;

}

.box-1{

    background-color: red;

}

.box-2{

    background-color: green;

}

.box-3{

    background-color: yellow;

}

Flex box xử lý theo 1 chiều: chiều ngang theo row or chiều học theo column

* Chiều ngang là trục main
* Chiều dọc là trục cross

2 trục này sẽ được quyết định bởi thuộc tính direction (mặc định là cross)

.container:

Flex-driection: row; các element sẽ nằm theo trục từ trái qua phải teo chiều nằm ngang

Flex-direction: column: xếp theo thứ tự từ trên xuống dưới

Flex-direction: column-reverse: xếp theo thứ tự từ dưới lên trên

Flex-wrap: wrap;

Flex-flow: row wrap (thay thế cho flex-direction và flex-wrap)

justify-content: ; căn chỉnh element theo trục main

align-items:; căn chỉnh element theo trục cross

.box:

flex-grow: ; tăng kích thước của flex item theo trục main (có thể sét ở box-item)

flex-shrink: ; flex item nằm tràn ra ngoài main thì dùng

flex-basis: ; xác định kích thước của 1 flex item

.flex-item:

Order để set vị trí cho flex box item