**Bootstrap Columns system**

If we take a bootstrap coloumn col-md-6 then we want to devide the column inside it. Then inside of the column will also considered as 12 columns.

<**div** *class=***"col-md-4 offset-md-1"**>  
 //each row will have again 12 columns

<**div** *class=***"row"**>  
 <**div** *class=***"col-md-6"**>  
 <**label** *for=***""**>Name</**label**>  
 <**input** *type=***"text"** *placeholder=***"your name \*"**>  
 </**div**>  
  
 <**div** *class=***"col-md-6"**>  
 <**label** *for=***""**>Email</**label**>  
 <**input** *type=***"email"** *placeholder=***"your email \*"**>  
 </**div**>  
 </**div**>  
  
  
 <**div** *class=***"row"**>  
 <**div** *class=***"col-md-6"**>  
 <**label** *for=***""**>Date</**label**>  
 <**input** *type=***"date"** *placeholder=***"date \*"**>  
 </**div**>  
  
 <**div** *class=***"col-md-6"**>  
 <**label** *for=***""**>Party number</**label**>  
 <**select** *name=***""** *id=***""**>  
 <**option** *value=***""**>Party number</**option**>  
 <**option** *value=***""**>Party id</**option**>  
 </**select**>  
 </**div**>  
  
 </**div**>  
  
</**div**>

**Offsetting Columns**

Move columns to the right using .col-md-offset-\* classes. These classes increase the left margin of a column by \* columns.

.col-md-offset-\* to leave a particular number of virtual Bootstrap columns to the left of any column (kind of like invisible place holders). It is used for giving space from margin-left.

<div class="col-md-12">

<div class="col-md-2 col-md-offset-1"> col-md-2 with offset 1</div>

<div class="col-md-2 col-md-offset-2"> col-md-2 with offset 2</div>

</div>

The first div will leave 1 bootstrap column on its left then start and

second div will leave 2 bootstrap column on its left then start.

<https://stackoverflow.com/questions/38500828/whats-exactly-the-purpose-of-col-md-offset>

<https://www.w3schools.com/bootstrap/bootstrap_grid_examples.asp>

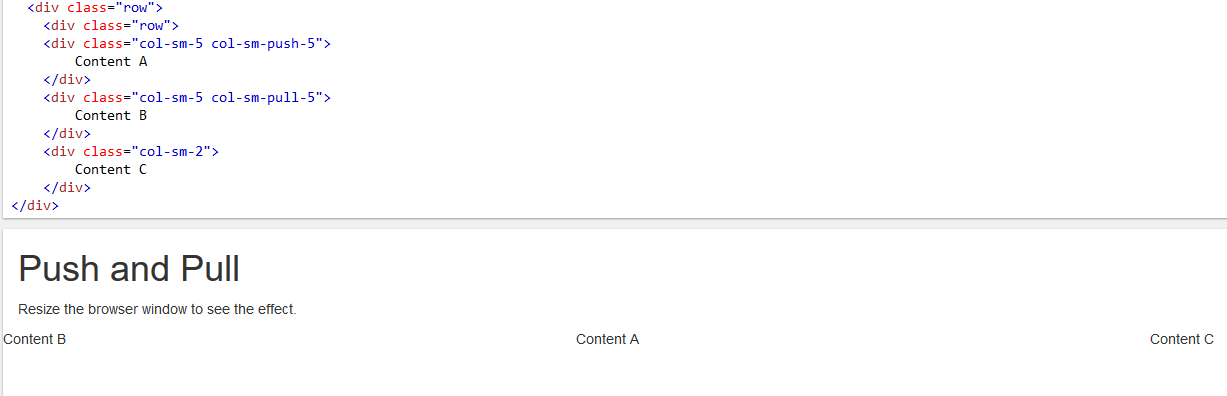
## Push And Pull - Change Column Ordering

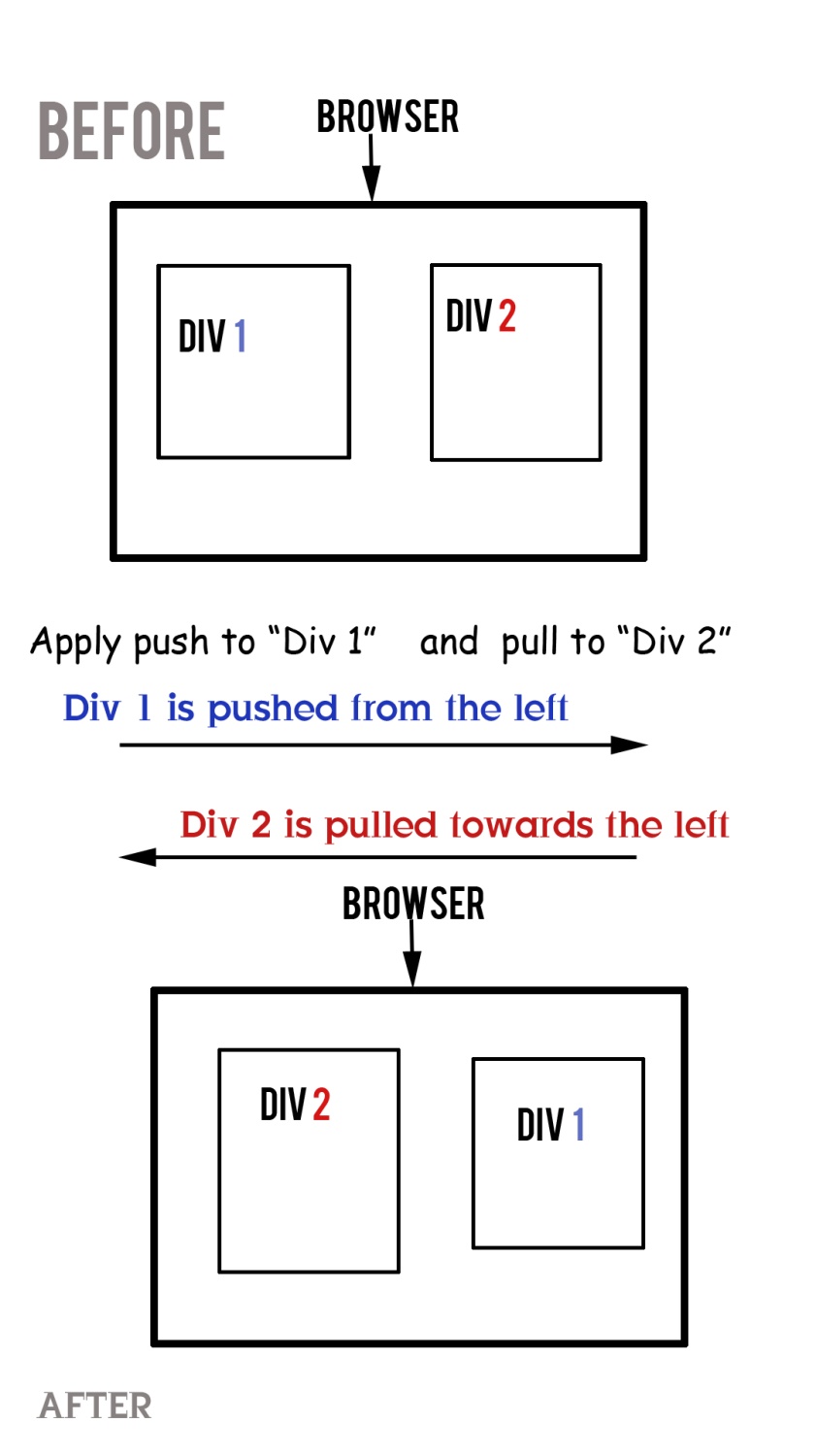
Change the order of the grid columns with .col-md-push-\* and .col-md-pull-\* classes

* col-vp-push-x = push the column to the right by **x** number of columns, starting from where the column would normally render -> position: relative, on a **vp** or larger view-port.
* col-vp-pull-x = pull the column to the left by **x** number of columns, starting from where the column would normally render -> position: relative, on a **vp** or larger view-port.

**vp** = xs, sm, md, or lg

**x** = 1 thru 12





**CSS**

.social-links a { /\*Means this style will be applied to all the <a/> tags under "social-links" class \*/  
 ***color***: #db9e10;  
 ***padding***: 0**px** 5**px**;  
 ***display***: **inline-block**;  
}

.btn-filled.btn-bookTable { /\* here we wrote 2 classes it means if both class are used in one element only then this style properties will apply. Notice there is no space between the class names\*/  
 ***position***: **relative**;  
 ***z-index***: 1;  
 ***color***: #E2E2E2;  
 ***display***: **inline-block**;  
 ***padding***: 10**px** 30**px**;  
 ***text-transform***: **uppercase**;  
 ***margin***: 0**px** 13**px**;  
}

.btn-filled.btn-bookTable:after { /this after pseudo style will be applied to the element who have these two classes/  
 ***content***: **''**; /\*This is must for sudo element\*/  
 ***z-index***: -1;  
 ***background***: #252A2C;  
 ***position***: **absolute**;  
 ***left***: 0;  
 ***top***: 0;  
 ***width***: 100%;  
 ***height***: 100%;  
 ***opacity***: 0.8;  
}

.nav-item.nav-link.my-nav-itm.active:before{  
 ***content***: **''**; /\*This is must for sudo element\*/  
 ***position***: **absolute**;  
 ***top***: -8**px**;  
 ***width***: 100%;  
 ***height***: 3**px**;  
 ***background***: **white**;  
 ***left***: -3**px**;  
}

.ingredients-text:before, .testimonial-list:before { /Here we defined styles for 2 psuedo classes/  
 ***content***: **''**;  
 ***position***: **absolute**;  
 ***z-index***: -1;  
 ***top***: 0**px**;  
 ***left***: 0**px**;  
 ***width***: 100%;  
 ***height***: 100%;  
 ***background***: **black**;  
 ***opacity***: .67;  
}

.is-sticky .header-area:before { /\*When a element will have ".is-sticky" as a parent element’s class and ".header-area" as own class then this style will be applied. Notice the space between the name of two classes\*/  
 ***content***: **""**;  
 ***background***: **black**;  
 ***opacity***: 0.8;  
 ***position***:**absolute**;  
 ***top***: 0;  
 ***left***: 0;  
 ***width***: 100%;  
 ***height***: 100%;  
 ***z-index***: 2;  
}

.is-sticky, .header-area:before { /\*It means these same set of style will be applied in these 2 classes. Elements using any of these class will get the same style. Notice a “,” comma between 2 class names \*/  
 ***content***: **""**;  
 ***background***: **black**;  
 ***opacity***: 0.8;  
  
}

/\*All the text in this template coming undet these all header tags will have the following properties\*/  
h1, h2, h3, h4, h5, h6 {  
 ***margin***: 0 0 15**px**;  
 ***font-weight***: 700;  
 ***font-family***: **'Yeseva One'**, **cursive**;  
}

.mainmanu {  
 ***display***: **inline-block**;  
 ***float***: **none**;  
 ***background-color***: **transparent !important**; /\* we need “!important” to overwrite the background-color property of the default parent \*/  
 ***text-align***: **right**;  
 ***position***: **relative**;  
 ***left***: 75**px**;  
}

/\*If we need a transparent gackground for a element such as this Button then we should do it using "class:after" but not by editing the background of the class\*/  
.btn-filled.btn-bookTable {   
 ***position***: **relative**;  
 ***z-index***: 1;  
 ***color***: #E2E2E2;  
 ***display***: **inline-block**;  
 ***padding***: 10**px** 30**px**;  
 ***text-transform***: **uppercase**;  
 ***margin***: 0**px** 13**px**;  
}  
.btn-filled.btn-bookTable:after {  
 ***content***: **''**;  
 ***z-index***: -1;  
 ***background***: #252A2C;  
 ***position***: **absolute**;  
 ***left***: 0;  
 ***top***: 0;  
 ***width***: 100%;  
 ***height***: 100%;  
 ***opacity***: 0.8;  
}

.section-title {  
 ***color***: #CC9900; /\*Dont need to give font-family because we have already defined all the font for text under <h2> tag before\*/  
 ***background***: **url**(**"images/about\_section\_below\_title\_line.png"**);  
 ***background-repeat***: **no-repeat**; /\*the image will not repeat\*/  
 ***background-position***: **center bottom**; /\*the background image will be placed in the bottom of the center of the container/class\*/  
 ***background-attachment***: **scroll**; /\*the image will scroll with the text it will not be fixed in one position\*/

***background-position***: **top**; /\*Scale the background image to be as large as possible\*/

***padding-bottom***: 35**px**;  
 ***margin-bottom***: 30**px**;  
}

<**div** *class=***"single-testimonial-item"**>  
 <**p**>If you’ve been to one of our restaurants, you’ve seen – and tasted – what keeps our customers coming back for more. Perfect materials and freshly baked food, delicious Lambda cakes, muffins, and gourmet coffees make us hard to resist! Stop in today and check us out!</**p**>  
 <**p** *class=***"testimonial-author"**>- food inc, New York</**p**>  
</**div**>

If we want to change the style of these both <p> tags under the class “single-testimonial-item”

.single-testimonial-item {  
 ***font-size***: 25**px**;  
 ***font-family***: **'Gentium Book Basic'**, **serif**;  
}

testimonial-author {  
 ***font-size***: 18**px**;  
 ***font-family***: **'Lato'**, **sans-serif**;  
}

the styles will not work and the default style we set for all <p> tag at the beginning will be applied to these 2 <p> tags.

So we need to do this:

For styling any <p> tags under the “single-testimonial-item” :

.single-testimonial-item p{ /Write the names like this/  
 ***font-size***: 25**px**;  
 ***font-family***: **'Gentium Book Basic'**, **serif**;  
}

And for any <p> tag with its own class (the 2nd <p> tag):

p.testimonial-author { /Write the names like this/  
 ***font-size***: 18**px**;  
 ***font-family***: **'Lato'**, **sans-serif**;  
}

.contact-form input[type=**"text"**], input[type=**"email"**], input[type=**"date"**], select { /\*to select multiple input type elements\*/  
 ***width***: 100%;  
 ***border***: 1**px solid** #C8C6C6;  
 ***padding***: 5**px**;  
 ***margin-bottom***: 10**px**;  
}  
.contact-form input[type=**"text"**]:hover, input[type=**"email"**]:hover, input[type=**"date"**]:hover, select:hover {  
 ***border***: 1**px solid** #CC9900;  
}

Important Note about **z-index**

Z-index is a “**position: absolute**” property because it always used to make an element come up or go below another element/parent element.

So if the parent is “**position: relative**” and have “”**z-index: 999**” and it’s child in “**position: absolute**” have **no z-index** or **z-index: 0** then the child will still be above the parent.

Like:

**<div class='parent'> <h1>The z-index Property</h1>**

**<img class=”child” src="w3css.gif" width="100" height="140">**

**</div>**

**.parent {**

**position: relative;**

**z-index: 10000;**

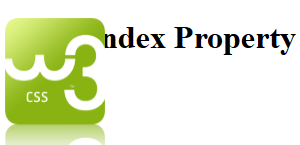
**}**

**.child {**

**position: absolute;**

**z-index: 0;**

**}**

****

In this case if we still want to take the child element below the parent then we need to set the

“**z-index: -1**” for the child. Like:

**.parent {**

**position: relative;**

**z-index: 10000;**

**}**

**.child {**

**position: absolute;**

**z-index: -1;**

**}**

****

Note: “z-index: 1” of a “position:fixed” element will get more priority than “z-index: 1” of a “position:relative” element.

# How TO - Parallax Scrolling

<https://www.w3schools.com/howto/howto_css_parallax.asp>

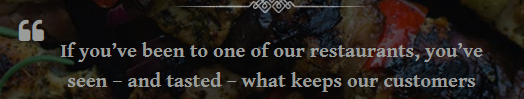
.parallax {   
  /\* The image used \*/  
  background-image: url("img\_parallax.jpg");  
  
  /\* Set a specific height \*/  
  height: 500px;   
  
  /\* Create the parallax scrolling effect \*/  
  background-attachment: fixed;  
  background-position: center;  
  background-repeat: no-repeat;  
  background-size: cover;  
}

Or

body, html {  
  height: 100%;  
}  
  
.parallax {   
  /\* The image used \*/  
  background-image: url("img\_parallax.jpg");  
  
  /\* Full height \*/  
  height: 100%;   
  
  /\* Create the parallax scrolling effect \*/  
  background-attachment: fixed;  
  background-position: center;  
  background-repeat: no-repeat;  
  background-size: cover;  
}

**Font Awsome**

To add queation icons from font awesome like this:

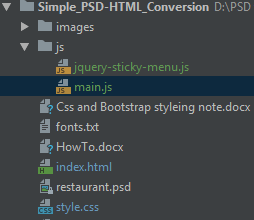


We need to do this:

#testimonial-carousel:before {  
 ***font-family***: **"FontAwesome"**;  
 ***content***: **"\f10d"**; //unicone id of the icon  
 ***position***: **absolute**;  
 ***top***: -31**px**;  
 ***left***: -25**px**;  
 ***color***: #9A9D9A;  
 ***font-size***: 34**px**;  
}

**To make the Navigation bar stick to the top of the page**

Open a folder name “js”



Go to: <http://stickyjs.com/>

In the main.js file:

 $(document).ready(function(){  
    $(".the\_warpper\_class\_name\_of \_the\_menu").sticky({topSpacing:0});  
  });

And in the “**jquery-sticky-menu.js**” put the code from (download the plugin zip file from the page> take the code from “jquery.sticky.js”)

Then in “index.html”:

<**head**>  
 /other imports…/  
 <**script** *src=***"js/jquery-sticky-menu.js"**></**script**>  
 <**script** *src=***"js/main.js"**></**script**>  
</**head**>

The whole element containing the menu which will be sticky on scroll must be “position:fixed !important” and “z-index:999 !important” to be able to operate properly like:

<**div** *class=***"header-area"**>  
 <**div** *class=***"container"**>  
 <**div** *class=***"row"**>

<**div** *class=***"col-md-2"**>  
 <**div** *class=***"logo"**>  
 <**img** *src=***"images/logo.png"**>  
 </**div**>  
 </**div**>

<**div** *class=***"col-md-10"**>  
  
 <**div** *class=***"mainmanu"**>  
 <**nav** *class=***"navbar navbar-expand-lg bg-transparent"**>  
  
 <**div** *class=***"collapse navbar-collapse"** *id=***"**   
   
 </**div**>  
  
 <**div** *class=***"social-links"**>  
   
 </**div**>  
 </**nav**>

</div>

</div>

</div>  
 </**div**>  
 </**div**>

.header-area {  
 ***width***: 100%;  
 ***z-index***: 999;  
 ***position***: **relative**; /\*relative\*/  
}  
.is-sticky .header-area:after { /\*When a element will have ".is-sticky" as parent class and ".header-area" as own class then this style will be applied. Notice the space between the name of two classes\*/  
 ***content***: **""**;  
 ***background***: **black**;  
 ***opacity***: 0.8;  
 ***position***: **absolute**;  
 ***top***: 0;  
 ***left***: 0;  
 ***width***: 100%;  
 ***height***: 100%;  
 ***z-index***: -1;/\*z-index: -1;\*/  
}  
.is-sticky .header-area{ /\*When on scroll the class “.is-sticky” gets added to the parent of “.header-area” this is a must to make the sticky header work and clickable. Or else it will inherit wrong properties and will not work\*/   
 ***position***: **fixed !important**;  
 ***z-index***: 999 **!important**;  
}

For smooth transition between pages add the following jQuery:

Add the class “**.smooth-menu**” to each navigation menu item like:

<**div** *class=***"navbar-nav"**>  
 <**a** *class=***"nav-item nav-link my-nav-itm smooth-menu active"** *href=***"#home"**>**Home** <**span** *class=***"sr-only"**>**(current)**</**span**></**a**>  
 <**a** *class=***"nav-item nav-link my-nav-itm smooth-menu"** *href=***"#about"**>**About**</**a**>  
  
</**div**>

Then add the following jQuery in the main.js inside the parent $(document) function

$(**".smooth-menu"**).on(**'click'**, function(*event*) {  
  
 // Make sure this.hash has a value before overriding default behavior  
 if (this.**hash** !== **""**) {  
 // Prevent default anchor click behavior  
 *event*.preventDefault();  
  
 // Store hash  
 var hash = this.**hash**;  
 var h=**'70' //how far the page should scroll from top** // Using jQuery's animate() method to add smooth page scroll  
 // The optional number (800) specifies the number of milliseconds it takes to scroll to the specified area  
 $(**'html, body'**).animate({  
 **scrollTop**: $(hash).**offset**().**top** - h + **'px'** }, 800, function(){  
  
 // Add hash (#) to URL when done scrolling (default click behavior)  
  
 });  
 } // End if  
});

**ScrollSpy:**

If we want to make the navigation menu items automatically selected upon on visiting or scrolling them then add the following bootstrap

<**body** *data-spy=***"scroll"** *data-target=***".navbar"**>

Where the “.navbar” is the class of navigation items.

Or add this jQuery in main.js inside the $(document) function:

$(**"body"**).scrollspy({  
 **target**: **'.navbar'**,  
 **offset**: 10  
});

<**a** *class=***"nav-item nav-link my-nav-itm smooth-menu active"** *href=***"#home"**>**Home** <**span** *class=***"sr-only"**>**(current)**</**span**></**a**>  
<**a** *class=***"nav-item nav-link my-nav-itm smooth-menu"** *href=***"#about"**>**About**</**a**>  
<**a** *class=***"nav-item nav-link my-nav-itm smooth-menu"** *href=***"#ingredients"**>**Ingredients**</**a**>  
<**a** *class=***"nav-item nav-link my-nav-itm smooth-menu"** *href=***"#food-menu"**>**Menu**</**a**>  
<**a** *class=***"nav-item nav-link my-nav-itm smooth-menu"** *href=***"#reviews"**>**Reviews**</**a**>  
<**a** *class=***"nav-item nav-link my-nav-itm smooth-menu"** *href=***"#reservations"**>**Reservations**</**a**>

I am trying to import React.js codes from from a .js file and use it in my index.html with react cdn imports. I needed to to very small task so I didn't wanted to use "create react app".

I there any possible way to use React.js just like we can use jQuery/Vue.js using only cdn imports and importing them from a separate .js file?

What if I want to use React on a existing project?

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>

<script src="https://unpkg.com/babel-standalone@6/babel.min.js"> </script>

<script src="js/my\_react.js"></script>

</head>

<body>

<div id="root"></div>

</body>

</html>

class Greeting extends React.Component {

state={

name: 'Deutschland'

}

render() {

return (<h1>Hello {this.state.name}</h1>);

}

}

ReactDOM.render(<Greeting />, document.getElementById('root') );