**@yield**

@yield is just a shorthand for a @section. For @yield, it always gets the section from the child layout. That means it always overrides the @yield part, even if it has a default defined as @yield('section', 'Default Content')

So when the Laravel executes your blade file, it first checks if you have extended a master layout, if you have extended one, then it shifts to the master layout and starts executing and if found @yield then it shifts to the child layout and again it comes back to the master. And of course, finally, it parses the blade format to raw php and gives the output in html

**For example:**

At Parent:

@yield('content')

At Child:

@extends('website.layout.master')

@section('content')

//html ….

@stop

**@include**

@include is similar to php's include() function, but it still can have @sections inside. The main difference between @yield and @include is: @include defines view to inject into parent. @yield defines section to be injected. @yield works only if your view @extends the parent view.

**For example:**

At parent:

@include('website.inc.main\_menu')

At another directory:

<nav>….</nav>

**@show and @parent**

@show is a blade directive that is used with @section, blade parse if you want to show this as section that have a default content.

If you already have a @section defined in the master layout, it will be overridden unless you specify @parent inside the child layout's @section.

Using just @yield() you wont be able to have a default content. Using @section()..@show you'll be able to have a default content and you can do stuff like.

**Example 01:**

At Parent:

@section('rightSidebar')

<div class="well">

<h3>Right Sidebar</h3>

<p>This is the sidebar</p>

@show

</div>

At view:

@section('rightSidebar')

@parent

<p>This is apended text</p>

@stop

**Example 02:**

At parent:

<title>

@section('title')

Laravel

@show

</title>

At view:

@section('title')

@parent

| Some page

@stop

Output:

<title>

Laravel | Some Page

</title>