**@yield**

@yield is just a shorthand for a @section. The @yield is mainly used to define a section in a layout. You define a area (@yield) within the layout. It can have a default content but it will be replaced by a view section.

**@parent**

**If you already have a @section (@show) defined in the master layout, it will be overriden unless you specify @parent inside the child layout's @section (@stop)**.

But 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 getting the @sections that you have overridden and the main content i.e the @yield from the child layout. And of course, finally, it parses the blade format to raw php and gives the output in html/css/js

**In Master Layout @section()…...@show:**

The @section()….@show and @yield()….@show are both blade directives, blade parse them you want to show this as section to every pages that have a default content but it will not be replaced by a view section. And @yield()….@show has no default value if have it replaced by view section. The main difference are:

1. Using just @yield()….@show you wont be able to have a default content.

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

**Examples:**

i.e 01,

@section('sidebar')

This is the master sidebar.

@show

**And in the child page @section() ….@stop and @parent**

@section(‘sidebar’)

@parent

//Appended new content

@stop

i.e 02,

**in parent:**

<title>

@section('title')

**Laravel** **(Default content)**

@show

</title>

**in view:**

@section('title')

@parent

**| Some page**

@stop

**And output:**

<title>

**Laravel** **(Default content)** **| Some Page**

</title>

[**N.B**, Notice the @parent you can position it before or after the content and it will inherit your default content, if you don't use @parent the default will be completely replaced]

**@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.