**Compile function**:If we want some of the code execution at the part of compilation phase. It happens only once.It loads and traverse the template.Same template never compiled more then once.Compile function returns a link function or an object which performs the pre , post . No scope is present as a part of compilation process. We can not access the scope during the compilation process.

No instance or the clone of the template created yet. This is the part of linking process. No data is maintained or updated during the compilation process. This is because we do not have any scope here. Suppose we applied the border color then same view is available for all the clones for the directive template that we have managed during the compilation process.So that same DOM can be cloned.

Note : Again saying we can not play with the data/events.No events or event handles during this phase.

compile(tElement,tAttributes){}

Note : <div message text=”first”>..</div> complete refers to the tElement.

text=”first” this refers to the tAttributes

**Link** : It is the next phase after the compilation process.It has many phases.

As a part of link first whole **<div message text=”first”>..</div>**item gets cloned into the memory. This instance or clone is created just after the compilation process.Linking then proceed with this instance that got created into the memory.

Linking has the 3 major steps.

a)**Controller** : If it is available it is executed first.Every directive has it’s own controller associated with it.Once the template has been cloned the controller gets kicking in. In this phase the scope gets created. Here we can modify the data. Can any type of variables , properties , method to that scope. We can also access the DOM of the template. But it is not good to change the css of template.

b)**Pre Link** : It is the just next step of the controller. It also gets executed on the same template instance. The DOM template is available during the pre-link phase. The scope that get available at the controller phase still available in the pre link phase. User can access the scope related data here.The scope generated during the controller and that is available in pre link is not bound completely with the DOM template.Child directives are not processed at this moment. Can not access any of the child directive. Also not save to change the DOM template.

c)**Post Link :** It works on the same instance . It is the last phase before the render. Once the post link gets completed , it is going to render finally. Here DOM gets prepared completely . Means completely bound to the scope. Here we can manipulate the DOM.All the child directives are ready. We can access the scope but not required in this phase . We can attach any type of the listener for dom elements here. Post link is best place for applying the click event for the button. No save to save the data for the child elements. Data for the child element is best provided during the Pre link.

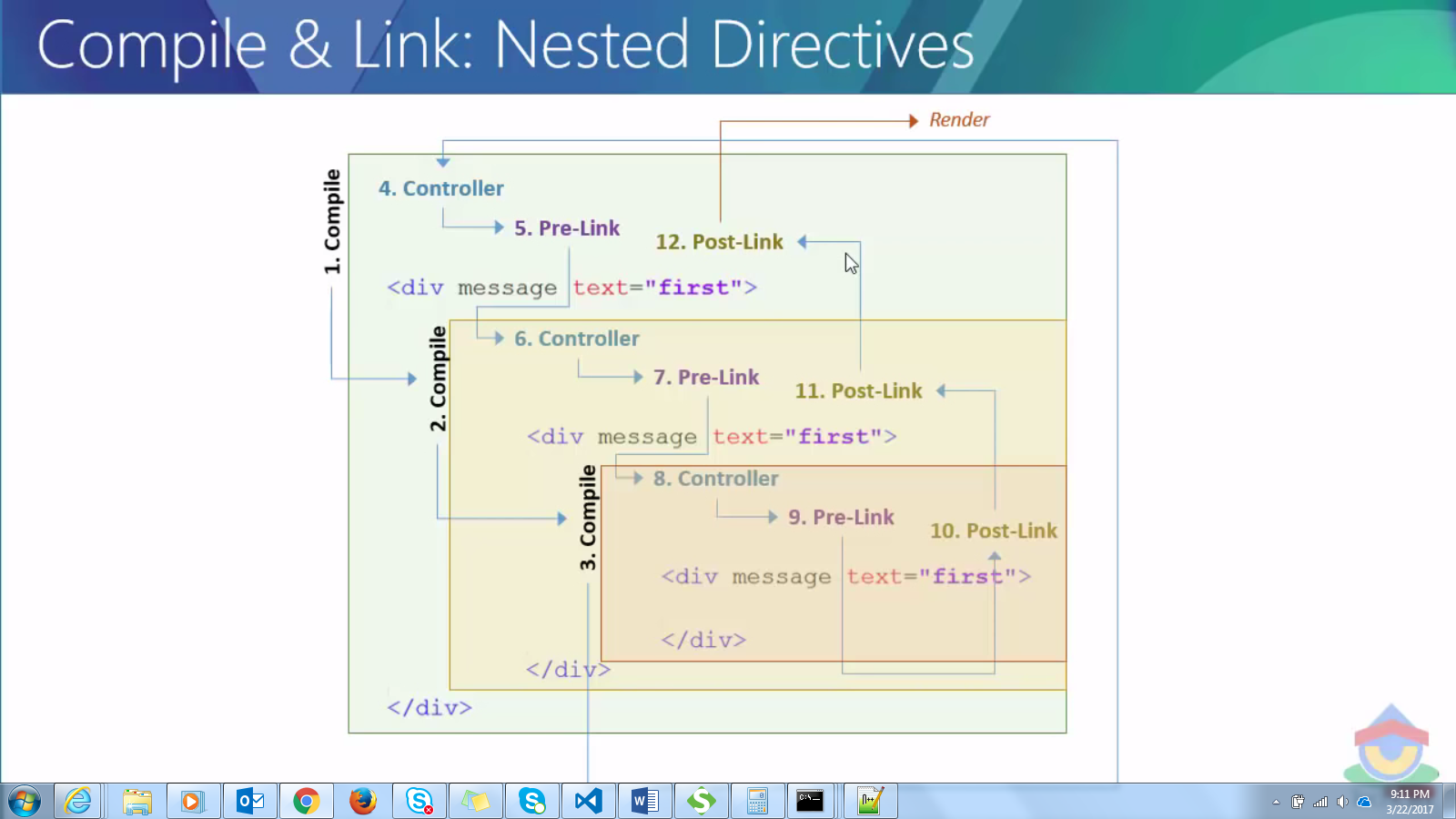
All the 3 steps are performed against the instance or the clone created into the memory.

Note : Once the linking gets completed the angularjs render that clone created on html view.

The execution of compile and the link:

**Nested directive :**

How the compile and link will work for the nested directive.



First the compilation of Our directive , then inner directives takes place. All the compilation for all directives needed to complete first before going to the other phase.

After this the control goes to the linking process. Controller invoked first of them.