1. Module Introduction

1. Components & Data Binding

2. Splitting Apps into Components

1. **Command**: ng g c cockpit –spec false

3. Property & Event Binding Overview

1. **Property & Event Binding**: Can be done with native html, custom directive/components.

4. Binding to Custom Properties

1. @Input(“aliasOptionalForOutsideWorld”) element:{type: string, name: string, content: string};

5. Assigning an Alias to Custom Properties

1. @Input(“servElement”) element:{type: string, name: string, content: string};
2. <app-server-element [srvElement]=”serverElement”> where serverElement is object which is passed from parent.

6. Binding to Custom Events

1. **@Output**() **serverCreated** = new EventEmitter<**{serverName: string, serverContent: string}>** 🡸 **Defining Custom Event for this component.**
2. this.**serverCreated**.emit(**{serverName: “someValue”, serverContent: “someOtherValue”});🡸**

7. Assigning an Alias to Custom Events

1. @Output() server:string; 🡸(server)=”handler()” or @Output(“serverAlias”) server:string; 🡸( serverAlias)=”handler()”

8. Custom Property and Event Binding Summary

NOTHING HERE

9. Understanding View Encapsulation

1. **View Encapsulation**: To apply the CSS style to the component’s template only not any other component’s template. Angular adds CSS attribute.

10. More on View Encapsulation

1. @Component({**encapsulation**: **ViewEncapsulation**.Emulated}); **Emulated**(Default), **Native**(Shadow DOM), **None**

11. Using Local References in Templates

1. <input **#templateRefVariable > using it {{ templateRefVariable}} or (click)=”handler(templateRefVariable)” 🡸 contains native HTML ref.**

12. @ViewChild() in Angular 8+

1. For Angular 8+, @ViewChild(“**templateRefVariable**”, {static:Boolean}): true: if you want to access element other than inside ngOnInit(){} otherwise false.  
   Same with @**ContentChild**()

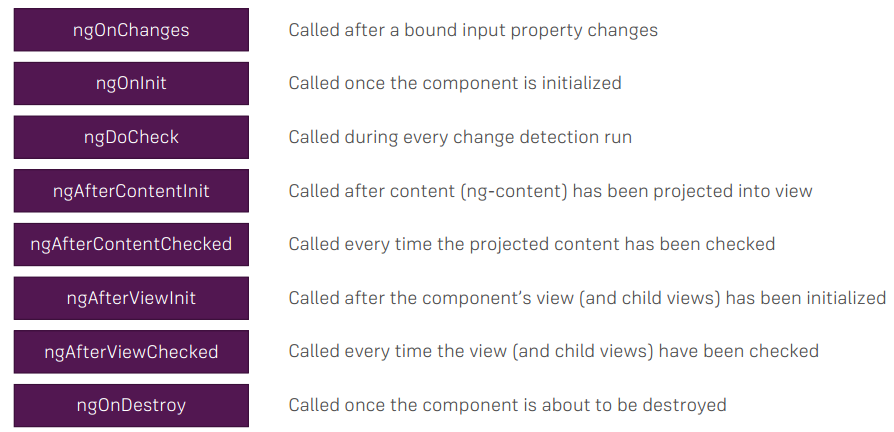
13. Getting Access to the Template & DOM with @ViewChild

1. **@ViewChild(“temlateReferenceVariable”)** propertyName: **ElementRef**;

14. Projecting Content into Components with ng-content

1. To pass HTML template from parent to child. Anything put b/w <server-app>template</server-app> is ignored by default. “template”, you want to pass to server app component so that server app component (child component) to add the received template to its own template.   
   **Solution**: <ng-content>. Wherever you place this tag inside server template, it marks that location to replace it with received template from parent template.
2. You can do the same with property biding but not recommended to prevent **cross-site scripting** even though you can handle that yourself too.

15.1 lifecycle

1. 

16. Seeing Lifecycle Hooks in Action

1. Nothing Here

17. Lifecycle Hooks and Template Access

Nothing Here

18. @ContentChild() in Angular 8+

19. Getting Access to ng-content with @ContentChild

1. For Angular 8+, ContentChild('contentParagraph', {static: true}) paragraph: ElementRef; if you use selected element inside ngOnInit() otherwise false.

19. Getting Access to ng-content with @ContentChild

1. <childComponent> <input #tempateRefVariable> </childComponent>
2. ChildComponent contains 🡺 @ContentChild(“tempateRefVariable”) refHolderProperty: ElementRef;