Angular Notes:

If and else statement in html:

<p \*ngIf="serverCreated; else noServer">Server was created, server name is {{serverName}}</p>

<ng-template #noServer>

<p>

no server is created

</p>

</ng-template>

Interpolation:

{{ variable\_name }}

Two way data binding:

[(ngModel)]="serverName">

Angular cli commands:

Ng g c component\_name

G= generate c = component

Directives example with property binding:

<p [ngStyle]="{backgroundColor: getColor()}" >The {{ 'Server' }} id {{ serverId }} is {{ getServerStatus() }}</p>

<Ng for = “let logItem of log; let i = index”

[ngStyle]=”{backgroundColor: i>= 4? ’blue’ : ’transparent’}”

[ngCLass] = “{‘white-text’: : I >= 5}”>

**[(ngModel)]?**

{{logitem}}

Remove testing class

In ng g ( --skipTests=true)

Add one class type

Ng g class “Classname” –type=model

Adding click events (click) = “functionName()”

**Javascript key value pairs** {type: ‘server’}

Similar to php ‘type’ => ‘server’

**Expose a property in lower components**

@Input(‘diffName’) element: ‘hi’

Import Input from angular/core

@output() an event

Used like (click) even \*can create your own events\* eventAdmitter

@component

Encapsulation: ViewEncapsulation.**none .native**

<inpuy #servernaminput> local reference

Local references through view child

Ng-content

Lifecycle component hooks

Ngonchanges,ngoninit, ngdocheck, ngaftercontentinit,ngaftercontentchecked,ngafterviewinit,ngafterviewchecked,ngondestroy