Commands:

ng g component file\_name --spec false;

ng g class ingredient-model --skipTests

**Angular Features:**

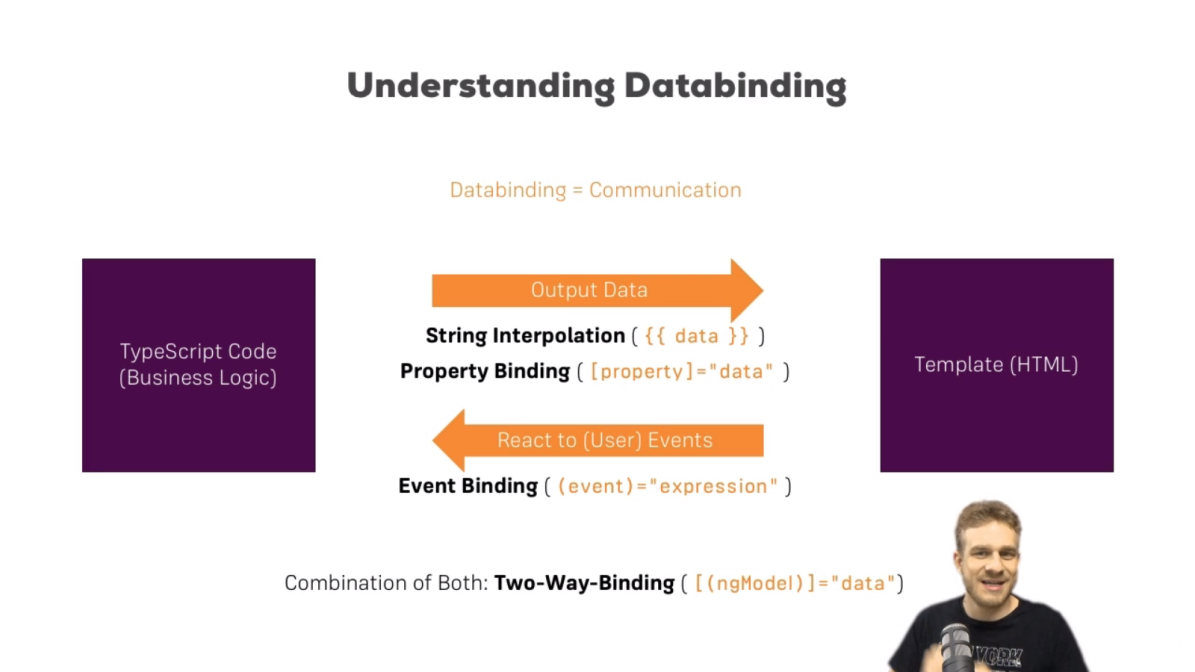
**Compiled to**

JavaScript

TypeScript

**By Angular CLI**

* **Angular is javascript framework which allows to create single page application.**
* **Index.html file is served as single page application.**
* **Index.html is being served in the browser when your application run on localhost:4200, to confirm right click view page source**
* <script type="text/javascript" src="[runtime.js](http://localhost:4200/runtime.js)"></script>
* <script type="text/javascript" src="[es2015-polyfills.js](http://localhost:4200/es2015-polyfills.js)" nomodule></script>
* <script type="text/javascript" src="[polyfills.js](http://localhost:4200/polyfills.js)"></script>
* <script type="text/javascript" src="[styles.js](http://localhost:4200/styles.js)"></script>
* <script type="text/javascript" src="[vendor.js](http://localhost:4200/vendor.js)"></script>
* <script type="text/javascript" src="[main.js](http://localhost:4200/main.js)"></script>
* **All these scripts are injected by the Angular CLI automatically.**
* **You will not able to see in index.html but whenever you do ng serve , it basically creates Javascript bundles and then it adds into index.html.**

****

**How to declare variables in typescript:**

studentName**:string='santa banta'; rollno:number=24;**

**1.1 String Interpolation:**

**Q- Can number can be use in place of string interpolation?**

**Ans: Yes**

<p>{{studentName | uppercase}} who has Roll No: {{rollno}} is {{ getRank() }} holder and {{ 'University Topper' }}</p>

1.2 Property Binding: [disabled]="customDisabled" or [disabled]="! customDisabled"

**Default Disabled**: It is disable by using default disable keyword

<button type="button" class="btn btn-primary" disabled>Add here </button><br>

**Condition Disabled**: It is disable by some condition disable

<button type="button" class="btn btn-primary" [disabled]="customDisabled">Add here </button><br>

setTimeout(()=>{this.customDisabled=false},1000);

1.3 Event Binding:

**TypeEvent**: whatever letter you will type in text box it will trigger typeEvent($event)

<input type="text" class="form-control" placeholder="typeEvent($event) is being called on typing" (input)="typeEvent($event)">

<p>{{propertyBinding}}</p>

In ts file.

typeEvent(event:any){

    this.propertyBinding=event.target.value;

  }

Example: Combination of both Property binding and Event binding

In the below example there is text field User Name, if user name is empty then Type Enable will be diabled.

If you click Type Enable, it will empty the text box .

<input type="text" class="form-control" [(ngModel)]="username"><br>

<div class="row">

    <div class="col-md-4">

        <button class="btn btn-primary" [disabled]="username ===''" (click)="username=''">Type Enable</button>

      </div>

</div><br>

String interpolation can be replaced by property binding:

<p>{{ message }}</p>

<p [innerText]="message"></p>

Note: both output will look same

If Else condition using directive \*ngIf

<p \*ngIf="username ===''; else noServer">If condition in single paragraph</p>

<ng-template #noServer>

  <p>Else Condition in single paragraph</p>

</ng-template>

Ternary Operator in Angular

this.status =Math.random() > 0.5 ? 'Online' : 'Offline' ;

ngStyle directive using Ternary Operator for property binding

* ngStyle is used for dynamically changing
* CSS

{ key : value }

<p [ngStyle]="{ backgroundColor: getColor() }">I am Online </p>

 getColor():string{

   return this.username==='' ? 'greeen' : 'red';

  }

ngClass directive : If condition is false , ngClass directive will not be in DOM,

ngFor directive: <div \*ngFor="let item of arrayList">{{item}}</div>

Custom Tag URl :

if selector: 'app-servers' then <app-servers></app-servers>

if selector: '[app-servers] then <div app-servers></div>

if selector: '.app-servers' then <div class="app-servers"></div>