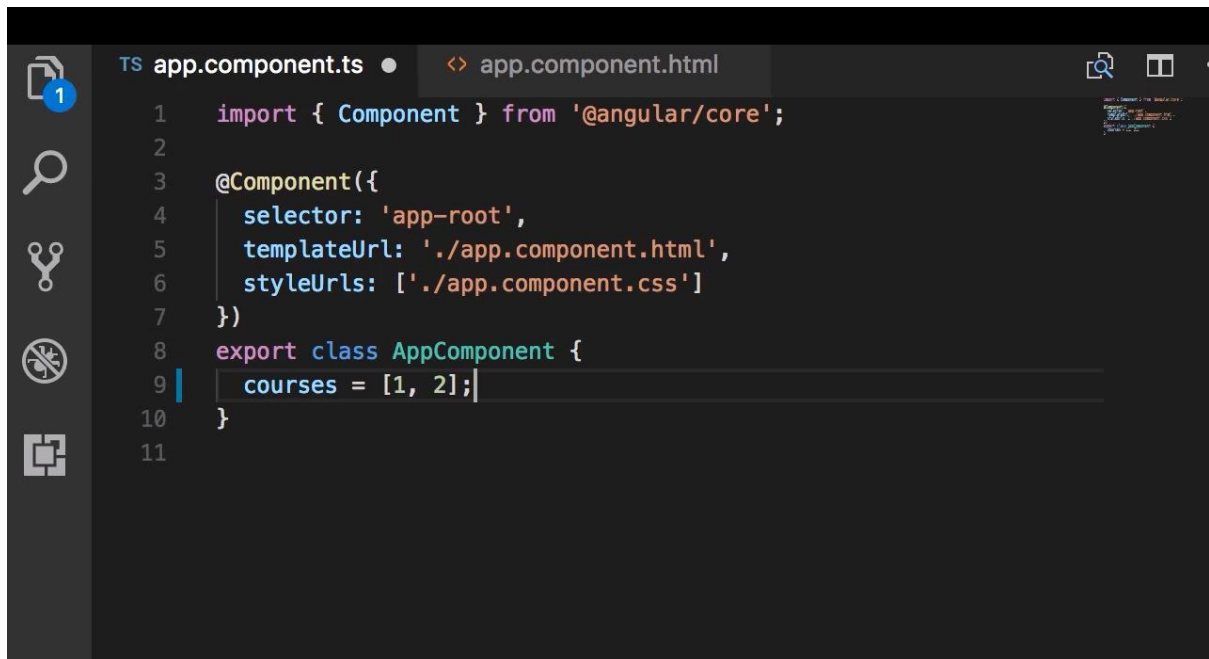


6. Directives

In this section...

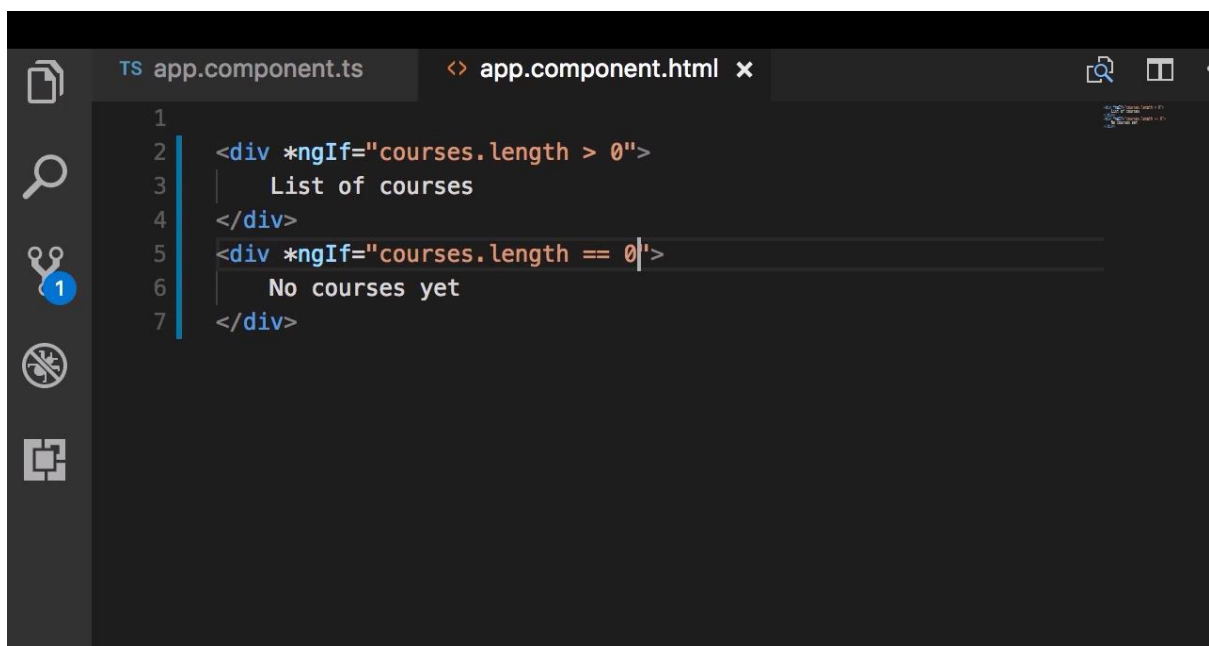
- ngFor
 - ngIf
 - ngSwitchCase
 - ngClass
 - ngStyle
 - Building custom directives
-

Ng-if



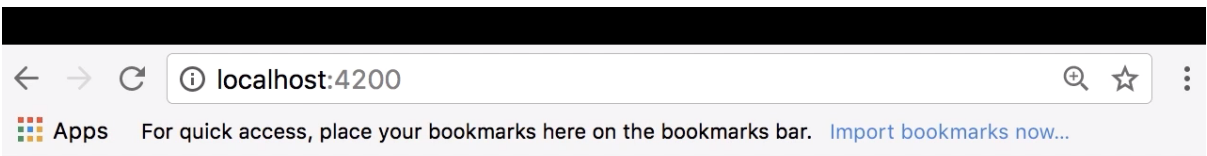
The screenshot shows the VS Code editor with the file 'app.component.ts' open. The code defines an Angular component named 'AppComponent'. It imports the 'Component' class from '@angular/core'. The '@Component' decorator is configured with 'selector: 'app-root'', 'templateUrl: './app.component.html'', and 'styleUrls: ['./app.component.css']'. The 'AppComponent' class is exported and contains a property 'courses' with the value [1, 2].

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   courses = [1, 2];
10 }
11
```



The screenshot shows the VS Code editor with the file 'app.component.html' open. The HTML template uses the 'NgIf' directive to conditionally render content. It displays 'List of courses' if the 'courses' array has a length greater than 0, and 'No courses yet' if the length is 0.

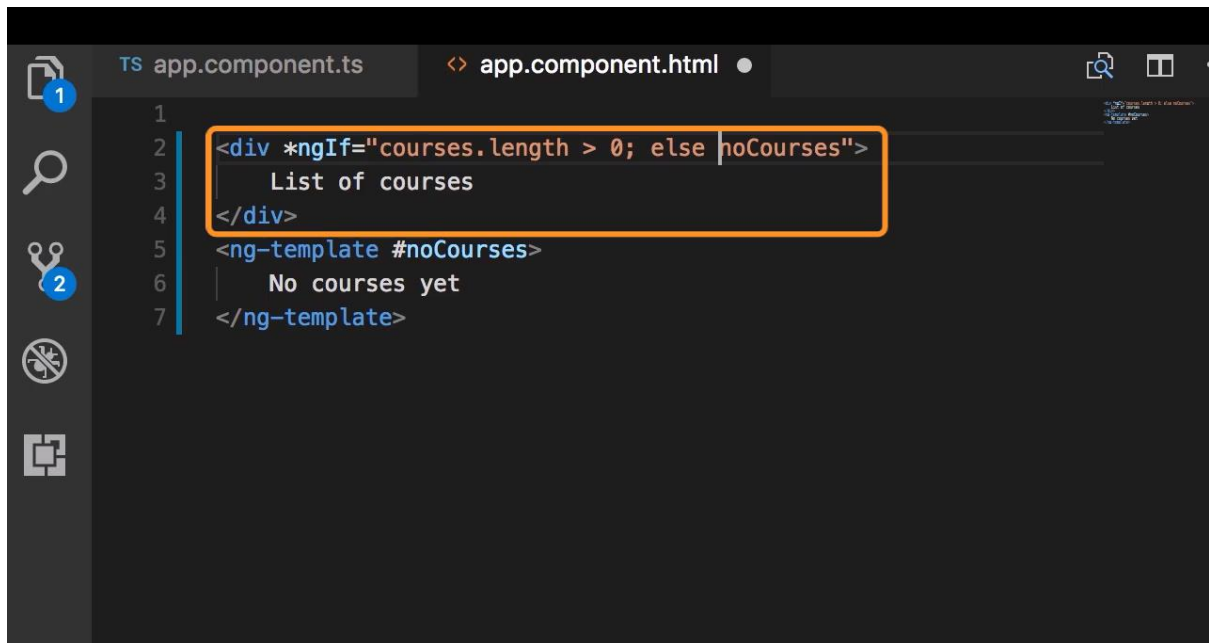
```
1
2 <div *ngIf="courses.length > 0">
3   List of courses
4 </div>
5 <div *ngIf="courses.length == 0">
6   No courses yet
7 </div>
```



List of courses



Another way

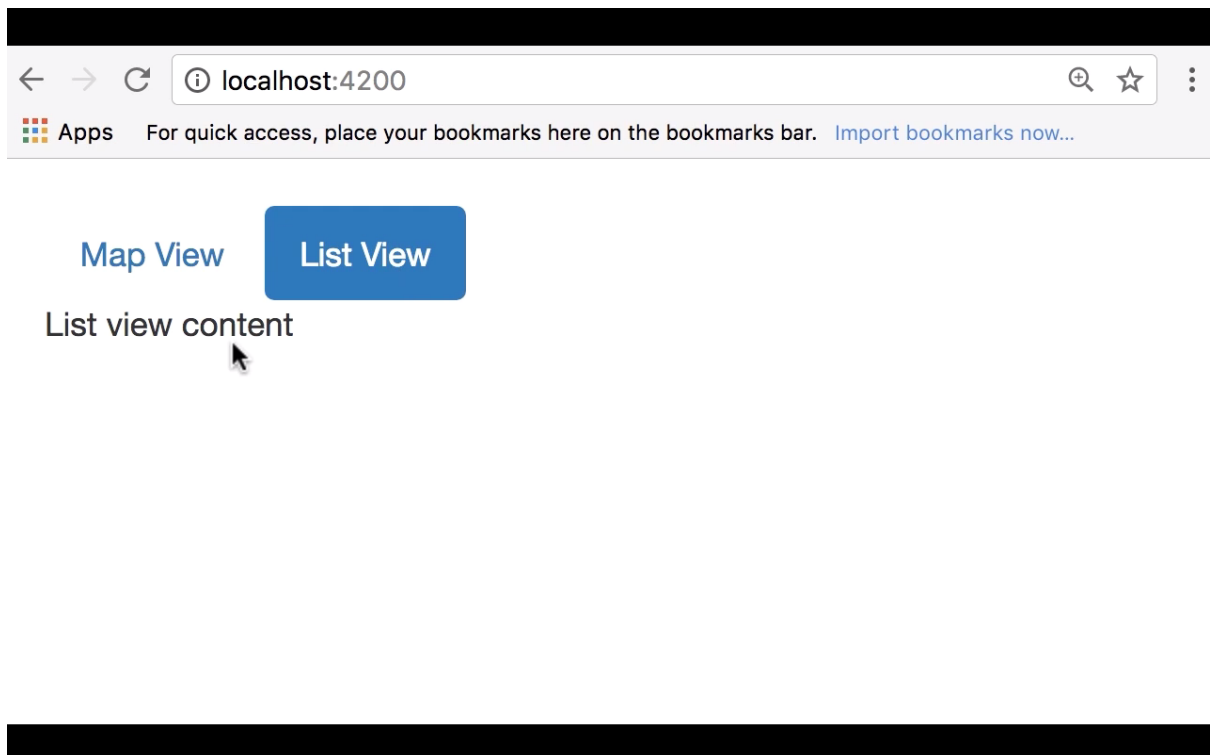
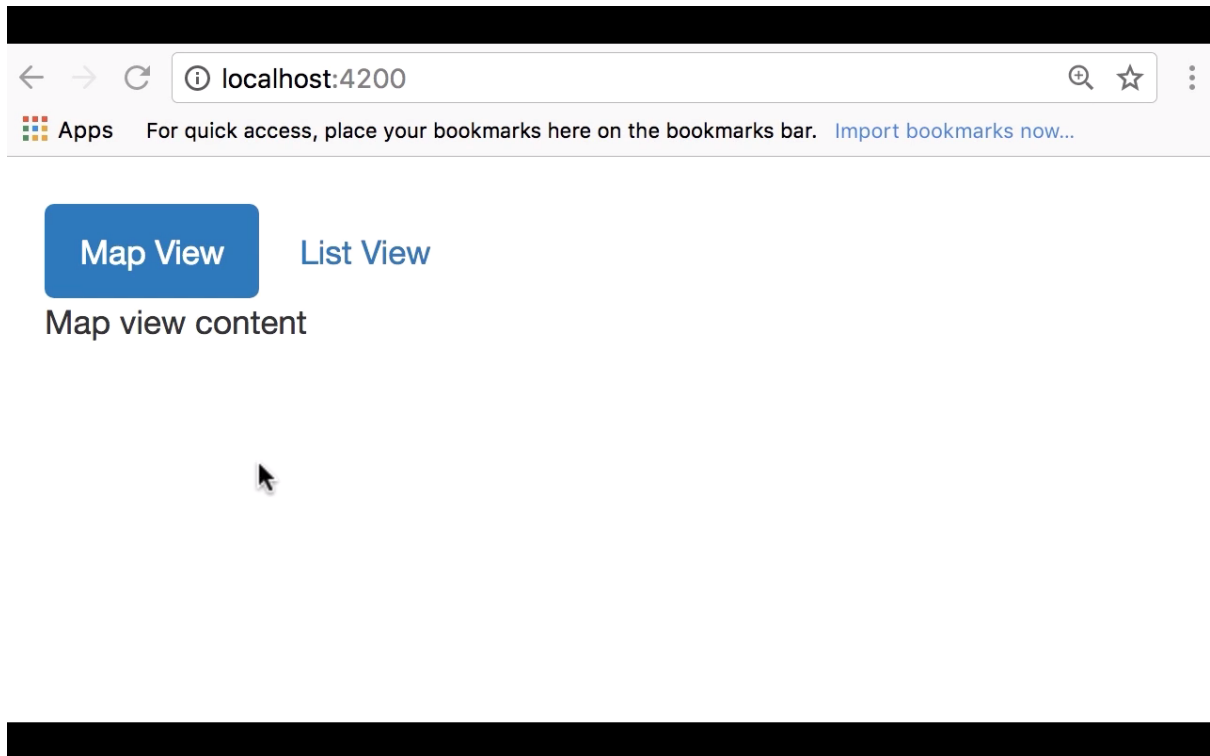


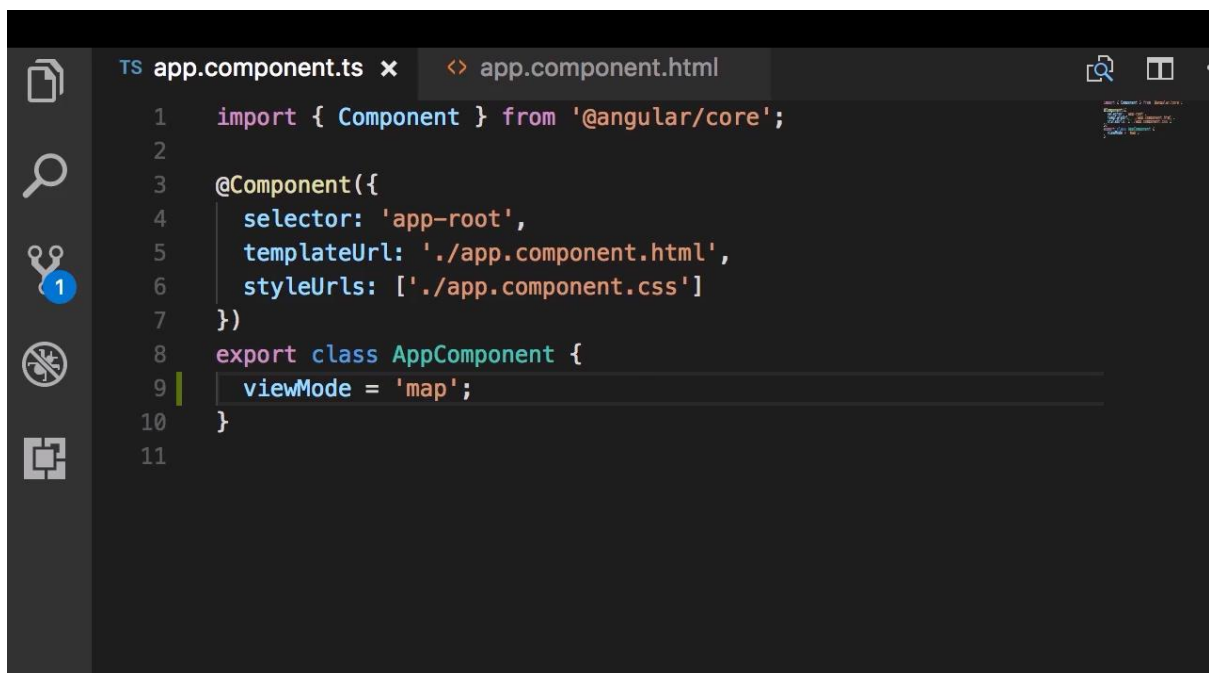
The screenshot shows a code editor with two tabs: 'app.component.ts' (TypeScript) and 'app.component.html' (HTML). The HTML tab is active, displaying the following code:

```
1
2 <div *ngIf="courses.length > 0; else noCourses">
3   List of courses
4 </div>
5 <ng-template #noCourses>
6   No courses yet
7 </ng-template>
```

An orange rectangular box highlights the opening tag of the first `<div>` element, which includes the `*ngIf` directive: `<div *ngIf="courses.length > 0; else noCourses">`. The code is written in a dark-themed editor with a sidebar on the left containing icons for Explorer, Search, Source Control, Run and Debug, and Extensions.

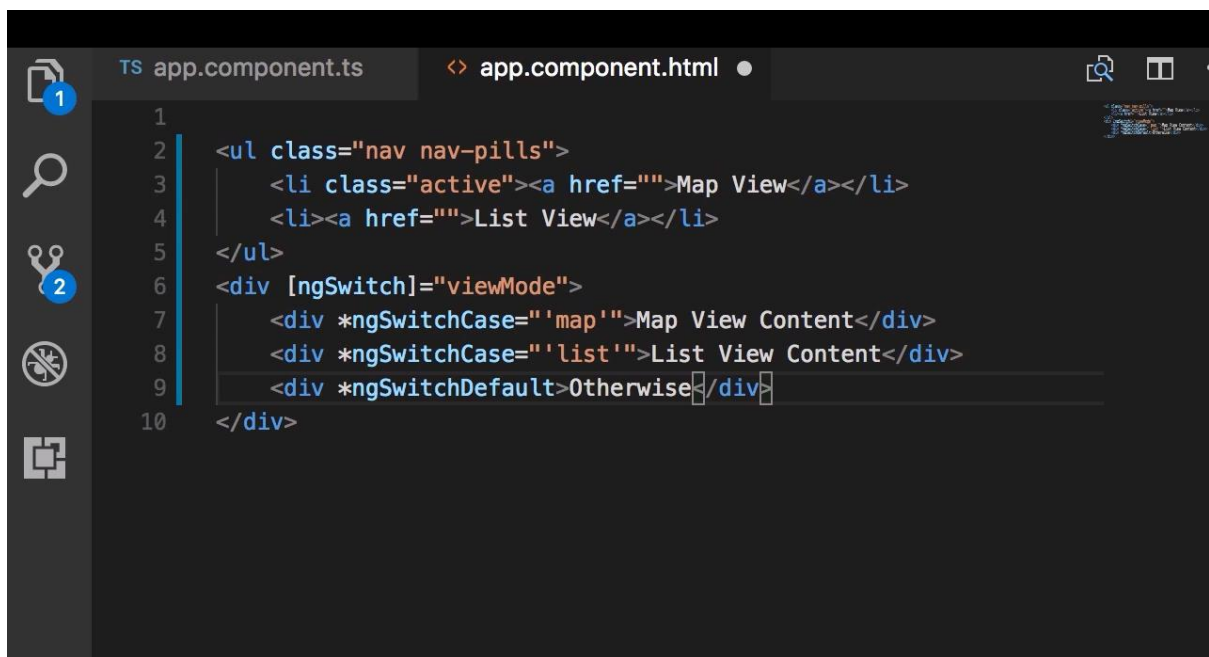
ngSwitchCase





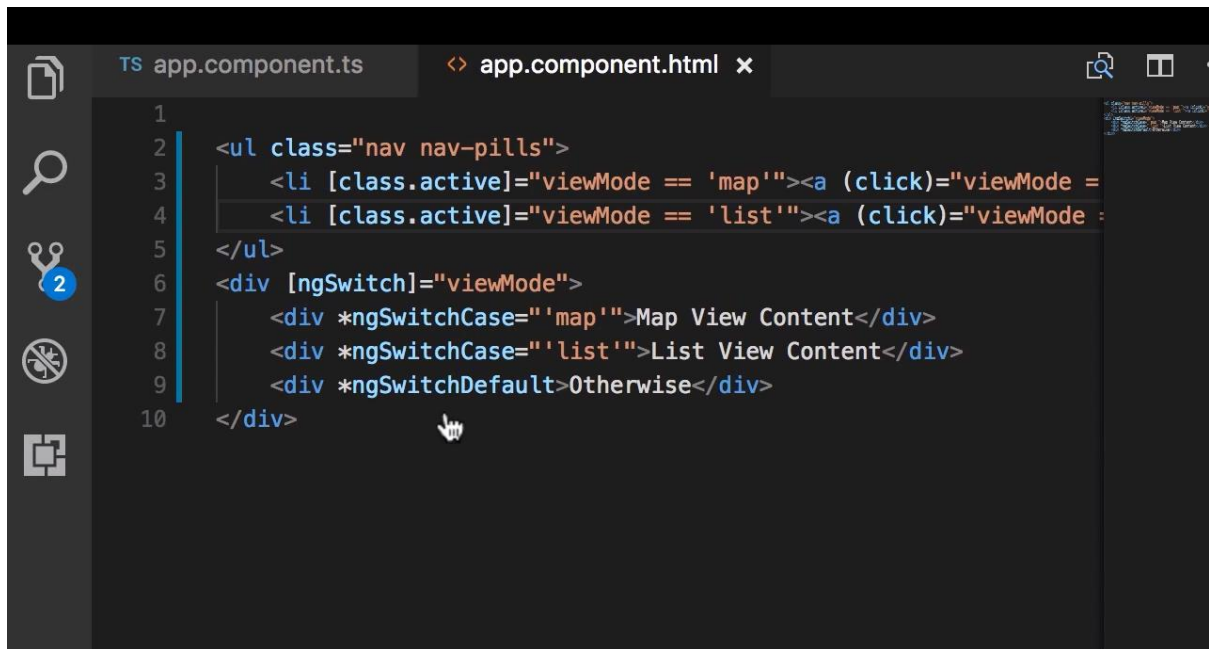
The image shows a VS Code editor window with two tabs: 'app.component.ts' (selected) and 'app.component.html'. The left sidebar contains icons for Explorer, Search, Source Control (with a '1' badge), Run and Debug, and Extensions. The main editor area displays the following TypeScript code:

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   viewModel = 'map';
10 }
11
```



The image shows the same VS Code editor window, but now the 'app.component.html' tab is selected. The left sidebar shows the Source Control icon with a '2' badge. The main editor area displays the following HTML code:

```
1
2 <ul class="nav nav-pills">
3   <li class="active"><a href="">Map View</a></li>
4   <li><a href="">List View</a></li>
5 </ul>
6 <div [ngSwitch]="viewModel">
7   <div *ngSwitchCase="'map'">Map View Content</div>
8   <div *ngSwitchCase="'list'">List View Content</div>
9   <div *ngSwitchDefault>Otherwise</div>
10 </div>
```



The screenshot shows a code editor with two tabs: 'app.component.ts' and 'app.component.html'. The 'app.component.html' tab is active, displaying the following HTML code:

```
1
2 <ul class="nav nav-pills">
3   <li [class.active]="viewModel == 'map'"><a (click)="viewModel =
4   <li [class.active]="viewModel == 'list'"><a (click)="viewModel =
5 </ul>
6 <div [ngSwitch]="viewModel">
7   <div *ngSwitchCase="'map'">Map View Content</div>
8   <div *ngSwitchCase="'list'">List View Content</div>
9   <div *ngSwitchDefault>Otherwise</div>
10 </div>
```

The code implements a navigation bar with two links, 'map' and 'list', and a corresponding view switcher using Angular's `ngSwitch` directive. The `class.active` attribute is used to highlight the selected view. The `ngSwitchDefault` case is labeled 'Otherwise'.

Follow the detailed steps in assignment for full implementation