Codergirl - JavaScript

Class 19

June 30th, 2021

Agenda

- Studio coding activity – end at 6:00 pm
- Lecture – end at 6:30 pm
- Exercise – end at 7:00 pm

Studio – Angular - 1

AngularJS expressions can be written inside double braces: {{ *expression* }}

Directives

1. Components
2. Structure
3. Attribute

ngFor

*ngFor = "let variableName of arrayName"

```
<div class='movies'>
    <h3>Movies to Watch</h3>
    <ol>
        <li>{{movies[0]}}</li>
        <li>{{movies[1]}}</li>
        <li>{{movies[2]}}</li>
        <li>{{movies[3]}}</li>
    </ol>
</div>
```

```
<div class='movies'>
    <h3>Movies to Watch</h3>
    <ol>
        <li *ngFor ="let movie of movies">{{movie}}</li>
    </ol>
</div>
```

ngIf

*ngIf = "condition”

```
<div class='movies' *ngIf ="movies.length > 3">
    <h3>Movies to Watch</h3>
    <ol>
        <li *ngFor ="let movie of movies">{{movie}}</li>
    </ol>
</div>
```

tags not generated if condition evaluates to false.

Logical Operators

Logical AND:

```
<p *ngIf ="conditionA && conditionB">Some text</p>
```

Some text appears on the web page only if **conditionA** and **conditionB** both return **true**.

Logical OR:

```
<p *ngIf ="conditionA || conditionB">Some text</p>
```

Some text appears on the page if either **conditionA** or **conditionB** return **true**.

Logical NOT:

```
<p *ngIf ="arrayName.length !== 0">Some text</p>
```

Some text appears when **arrayName.length** is NOT equal to 0.

if/else

The general syntax for adding an **else** block in Angular is:

```
1  <someTag *ngIf ="condition; else variableName">
2     <!-- HTML tags and content --->
3  </someTag>
4
5  <ng-template #variableName>
6     <!-- Alternate HTML tags and content --->
7  </ng-template>
```

Note that the **#** is required inside the **ng-template** tag.

Events

Angular uses a different approach to listen for events. The event name is placed in parentheses `()` and added inside an HTML tag. This *binds* the event to that element.

The more common events include:

1. `(click)`: Waits for the user to click on the element.
2. `(keyup)`: Waits for the user to release a key.
3. `(keydown)`: Waits for the user to press a key.
4. `(mouseover)`: Waits for the user to move the cursor over the element.

Events

<tag (event) = "statement"></tag>

Examples

1. This code waits for the user to click the "Submit" button and then calls the **addData** function:

```
<button (click) = "addData(arguments)">Submit</button>
```

2. This code waits for the user to move the mouse over the element and then sets the **choice** variable equal to the value of **option**:

```
<p (mouseover) = "choice = option">{{option}}</p>
```

3. This code just waits for any key to be pressed:

```
<div (keydown) = "true">Press Any Key</div>
```

User Input

```
<div class='movies col-4'>
    <h3>Movies to Watch</h3>
    <ol>
        <li *ngFor ="let movie of movies">{{movie}}</li>
    </ol>
    <hr>
    <input #newMovie (keyup.enter)='true' type='text' placeholder="Enter Movie Title Here"/>
    <button (click)="true">Add</button>
    <p>{{newMovie.value}}</p>
</div>      Jim Flores, a year ago • Added input practice files.
```

To accept user input, Angular requires three items:

1. The HTML **input** tag,
2. A variable to store the input, declared as **#variableName**,
3. An event that triggers data collection.

Events calling functions

```
(event) = "functionName(arguments...)"
```

Exercise

Questions?

Studio time!