## Looping

Our goal in this lecture is to display a *list* of jokes instead of just one.

 To add some visual jazz to our application we are going to be using the twitter bootstrap ui framework and specifically the *card* style for our JokeComponent.

**Learning Outcomes**

* Using Arrays in TypeScript.
* Using the NgFor directive to repeat an element.

**JokeListComponent**

We will create a new component called JokeListComponent with the following listing:

Note: jokes: Object[] – colon and word syntax means that we are giving property and its type.

And [] – means its array type.

And the perfect way to write the above line is

Jokes: Array<Object>;

Array is generic type. We will cover more on generic types in next section.

1:40

@Component({

selector: 'joke-list',

template: `

<

div class="card card-block"

\*ngFor="let joke of jokes">

<h4 class="card-title">{{joke.setup}}</h4>

<p class="card-text">{{joke.punchline}}</p>

<

/div

>

`

})

class JokeListComponent {

jokes: Object[];

constructor() {

this.jokes = [

{

setup: "What did the cheese say when it looked in the mirror?",

punchline: "Hello-Me (Halloumi)"

},

{

setup: "What kind of cheese do you use to disguise a small horse?",

punchline: "Mask-a-pony (Mascarpone)"

},

{

setup: "A kid threw a lump of cheddar at me",

punchline: "I thought

‘

That

’

s not very mature

’

"

},

];

}

}

### Arrays

The first change you’ll notice is that we have a property called jokes and the type is Object[].

The [] syntax in the type means *list of* or *Array*, so the jokes property holds a list of Objects.

Another perfectly legal way to write this would be Array<Object> but I prefer

Object[] since for me it’s easier to see the [] characters at a glance.

In the constructor we initialise this array with some hilarious cheese jokes.

### Card Element

You might notice in the template we are using some classes called card, card-block etc… this is from twitter bootstrap and it’s a style called a *card* which displays a rectangle with a border.

The basic HTML structure for a twitter bootstrap *card element* is like so:

<div class="card card-block">

<h4 class="card-title"></h4>

<p class="card-text"></p>

</div>

### NgFor

We want to repeat this card element for each joke in our array of jokes.

So we add a special syntax called an NgFor on the card element, like so:

<

div class="card card-block"

\*ngFor="let joke of jokes">

<h4 class="card-title"></h4>

<p class="card-text"></p>

<

/div

>

\*ngFor="let joke of jokes" will create a new HTML element, using the div element it’s attached to as a template, for every joke in the jokes array.

It will also make available to the element a variable called joke which is the item in the joke array it’s currently looping over.

The syntax translates to let <name-i-want-to-call-each-item> of <array-property-on-component>

This is what we call in Angular a *Directive*. Specifically it’s a *structural directive*

since it changes the structure of the DOM. We’ll be going through more built-in directives later on and also you’ll learn how to create your own.

So now we can display properties of this joke object in the HTML using {{joke.setup}} and {{joke.punchline}}, like so:

<div class="card card-block"

\*ngFor="let joke of jokes">

<h4 class="card-title">{{joke.setup}}</h4>

<p class="card-text">{{joke.punchline}}</p>

</div>

If you’ve worked with Angular 1 before, you probably used the ng-repeat directive.

 NgFor is the analogous directive In Angular. Its syntax is slightly different but they have the same purpose.

**Configuring**

In order to use our JokeListComponent we need to add it to the declarations on our NgModule and also mark it as the component we want to bootstrap the page with.

@NgModule({

imports:[BrowserModule],

declarations: [JokeComponent, JokeListComponent],

bootstrap: [JokeListComponent]

})

Since we are now bootstrapping JokeListComponent and it’s selector is joke-list we also need to change the root tag in our index.html, like so:

<

body class="container m-t-1"

>

<joke-list></joke-list>

<

/body

>



The classes

container

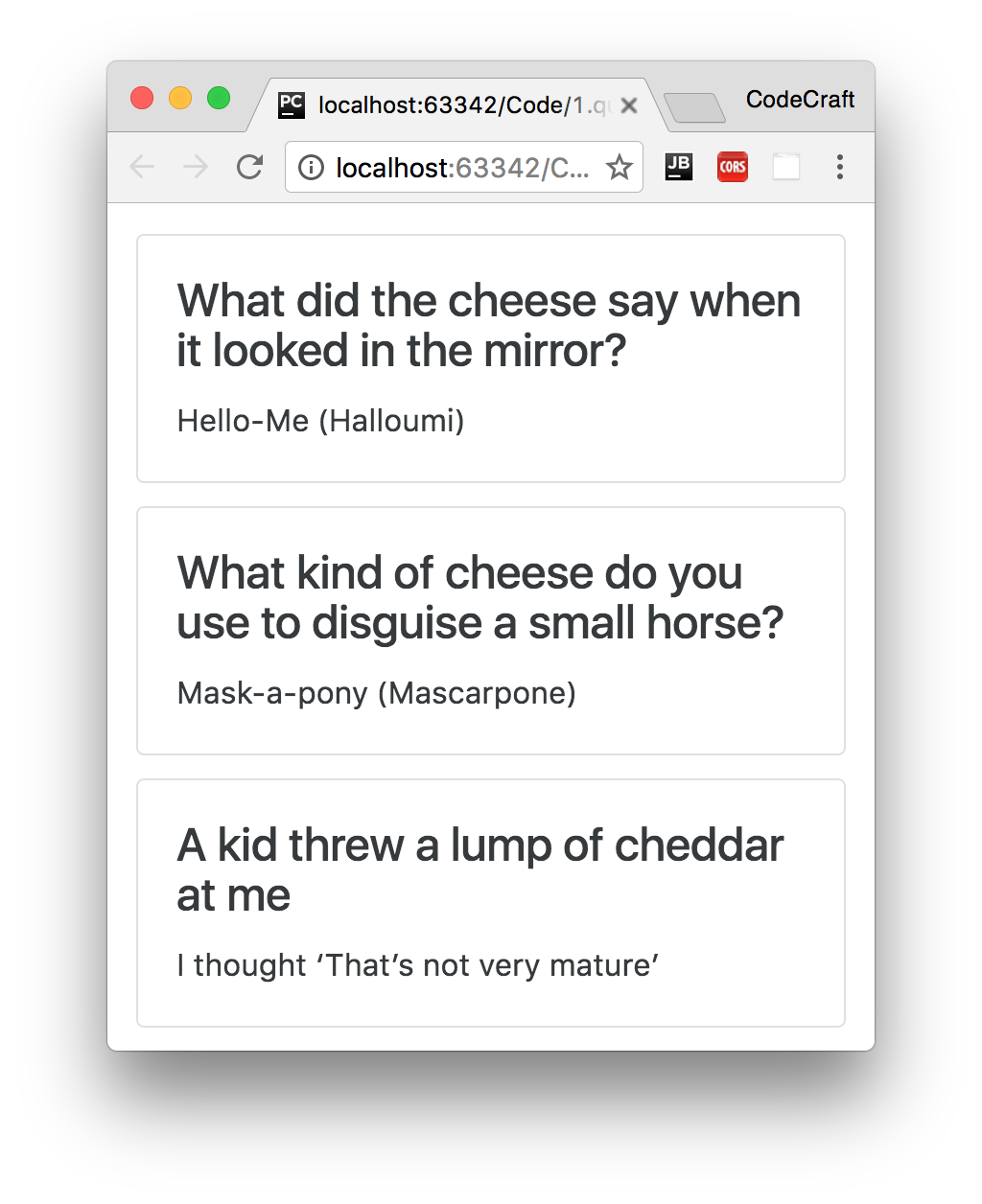
and

m-t-1

are from twitter bootstrap and add some nice

visual padding to the page.

Now if we run the application we see multiple jokes printed to the screen, like so:



**Summary**

When we declare an array in TypeScript we also tell it what *Type* of thing the array holds using Type[] or the Array<Type> syntax.

We can repeat the same element multiple times in Angular using the NgFor directive

**Listing**

<http://plnkr.co/edit/6BGJzWKFuPLFeBdKZ6z2?p=preview>

 Since we are now using the JokeListComponent as our root component, our rootcomponents tag has changed from <joke></joke> to <joke-list></joke-list>

*index.html*

<!

DOCTYPE html

>

<!--

suppress ALL

-->

<

html

>

<

head

>

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-

alpha.4/css/bootstrap.min.css">

<script src="https://unpkg.com/core-js/client/shim.min.js"></script>

<script src="https://unpkg.com/zone.js@0.7.4?main=browser"></script>

<script src="https://unpkg.com/systemjs@0.19.39/dist/system.src.js"></script>

<script src="systemjs.config.js"></script>

<script>

System.import('script.ts').catch(function (err) {

console.error(err);

});

</script>

<

/head

>

<

body class="container m-t-1"

>

<

joke-list></joke-list

>

<

>

/body

<

/html

>

*script.ts*

import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';

import {NgModule} from '@angular/core';

import {BrowserModule} from '@angular/platform-browser';

import {Component} from '@angular/core';

@Component({

selector: 'joke-list',

template: `

<

div class="card card-block"

\*ngFor="let joke of jokes">

<h4 class="card-title">{{joke.setup}}</h4>

<p class="card-text">{{joke.punchline}}</p>

<

/div

>

`

})

class JokeListComponent {

jokes: Object[];

constructor() {

this.jokes = [

{

setup: "What did the cheese say when it looked in the mirror?",

punchline: "Hello-Me (Halloumi)"

},

{

setup: "What kind of cheese do you use to disguise a small horse?",

punchline: "Mask-a-pony (Mascarpone)"

},

{

setup: "A kid threw a lump of cheddar at me",

punchline: "I thought

‘

That

’

s not very mature

’

"

},

];

}

}

@NgModule({

imports: [BrowserModule],

declarations: [JokeListComponent],

bootstrap: [JokeListComponent]

})

export class AppModule {

}

platformBrowserDynamic().bootstrapModule(AppModule);