Breaking Changes

By: Mosh Hamedani

As I explained before, most changes from beta to final release are syntactical and basic setup code. The only exception is a new concept called “Modules”. Let’s start with the simple syntactical changes and then over the next videos, I’ll introduce you to the module system in Angular 2.

Module names

All module names have been renamed from **angular2** to **@angular:**

Beta	`import {Component} from ‘angular2/core’;`
Final	`import {Component} from ‘@angular/core’;`

ngFor syntax

You use should use the **let** keyword to declare a loop variable:

Beta	`*ngFor=“#course of courses”`
Final	`*ngFor=“let course of courses”`

ngSwitch syntax

Beta	`<template [ngSwitchWhen]=“…”>`
Final	`<template *ngSwitchCase=“…”>`

Breaking Changes

By: Mosh Hamedani

Component metadata

In beta version, if we used custom directives and pipes in a component, we had to register them in the component metadata:

```
@Component({
    directives: […],
    pipes: […]
})
```

The problem with this approach was that if we used these custom directives and pipes in many places, we had to repeat the registration in each component, and this was tedious.

Now, we have a new concept called modules. Each application has at least one module (called **AppModule)**. As the application grows in size, we can divide AppModule into smaller, more focused modules, with each module focusing on one area of functionality.

With the module system, we register all components, directives and pipes that belong to a module in one place. So, **directives** and **pipes** have been removed from component metadata. You'll learn about modules in details over the next few videos.

Custom Pipes

In Angular Beta, the second parameter of the transform method used to be a **string[]**. We had to manually access an element in this array and convert it to a different type.

```
export class SummaryPipe implements PipeTransform {
    transform(value: string, args: string[]) {
        var maxWords = parseInt(args[0]);
    }
}
```

In Angular Final, this has been simplified. Instead of a **string[]**, we add one or more parameters using their correct type.

```
transform(value: string, maxWords: Number) {
}
```