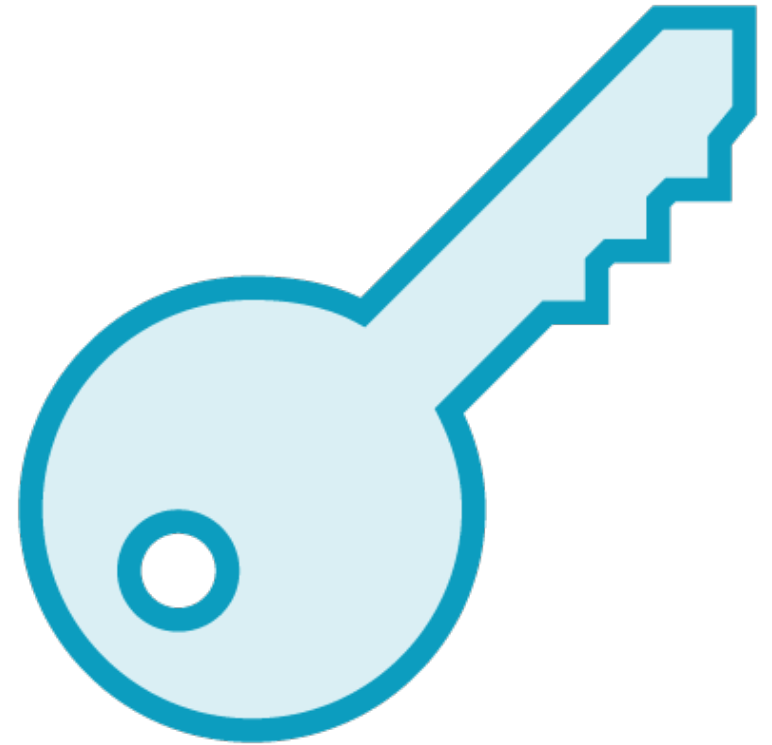


Angular 1 to Angular 2

7 Key Comparisons



1

Controllers to Components



Component

Angular 1

```
<h3 ng-controller="StoryController">
  {{vm.story.name}}
</h3>

(function () {
  angular
    .module('app')
    .controller('StoryController', StoryController);

  function StoryController() {
    var vm = this;
    vm.story = { id: 100, name: 'The Force Awakens' };
  }
})();
```

Angular 2

```
<my-story></my-story>

import { Component } from 'angular2/core';

@Component({
  selector: 'my-story',
  template: '<h3>{{story.name}}</h3>'
})
export class StoryComponent {
  story = { id: 100, name: 'The Force Awakens' };
}
```



2

Bootstrapping Angular

Angular 1

```
<html ng-app="app">
```

Angular 2

```
import { bootstrap } from 'angular2/platform/browser';  
import { AppComponent } from './app.component';  
  
bootstrap(AppComponent);
```



3

Structural Built-In Directives

Angular 1

```
<ul>  
  <li ng-repeat="vehicle in vm.vehicles">  
    {{vehicle.name}}  
  </li>  
</ul>  
<div ng-if="vm.vehicles.length">  
  <h3>You have {{vm.vehicles.length}} vehicles</h3>  
</div>
```

Angular 2

```
<ul>  
  <li *ngFor="#vehicle of vehicles">  
    {{vehicle.name}}  
  </li>  
</ul>  
<div *ngIf="vehicles.length">  
  <h3>You have {{vehicles.length}} vehicles</h3>  
</div>
```



Structural Directive

```
<ul>
  <li *ngFor="#vehicle of vehicles">
    {{vehicle.name}}
  </li>
</ul>
<div *ngIf="vehicles.length">
  <h3>You have {{vehicles.length}} vehicles</h3>
</div>
```

Local Variable

Structural Directive

Structural Directives

Indicated by the * prefix

Changes the structure



4

Data Binding

Interpolation



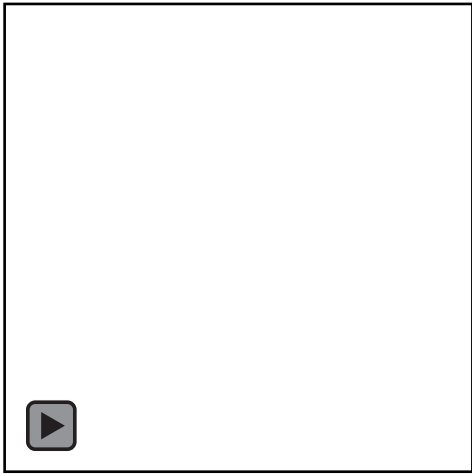
One Way Binding



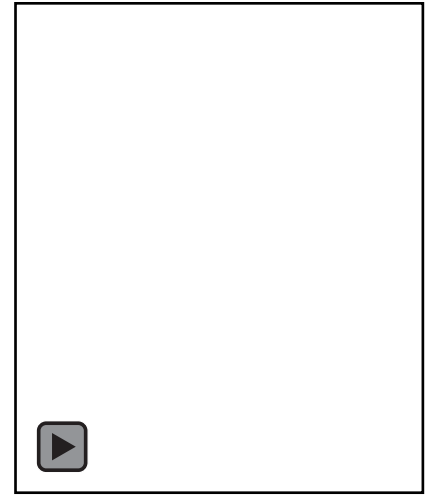
Event Binding



Two Way Binding



DOM



Component



Interpolation

Angular 1

```
<h3>{{vm.story.name}}</h3>
```

Context



Angular 2

```
<h3>{{story.name}}</h3>
```



1 Way Binding

Angular 1

```
<h3 ng-bind="vm.story.name"></h3>
```

Angular 2

```
<h3 [innerText]="story.name"></h3>
```

Any HTML Element Property

```
<div [style.color]="color">{{story.name}}</div>
```



Event Binding

Angular 1

```
<button  
  ng-click="vm.log('click')"  
  ng-blur="vm.log('blur')">OK</button>
```

Angular 2

```
<button  
  (click)="log('click')"  
  (blur)="log('blur')">OK</button>
```



2 Way Binding on an <INPUT>

Angular 1

```
<input ng-model="vm.story.name">
```

Angular 2

```
<input [(ngModel)]="story.name">
```

**Banana in a Box
[()]**



5

Removes the Need for Many Directives

Angular 1

```
<div ng-style=
  "vm.story ?
    {visibility: 'visible'}
    : {visibility: 'hidden'}">

  
  <br/>
  <a ng-href="{{vm.link}}">
    {{vm.story}}
  </a>

</div>
```

Angular 2

```
<div [style.visibility]=
  "story ? 'visible' : 'hidden'">

  <img [src]="imagePath">
  <br/>
  <a [href]="link">{{story}}</a>

</div>
```



Angular 2 Template Concepts Remove 40+ Angular 1 Built-In Directives



No Longer Need These Directives Either

Angular 1

`ng-click="saveVehicle(vehicle)"`

`ng-focus="log('focus')"`

`ng-blur="log('blur')"`

`ng-keyup="checkValue()"`

Angular 2

`(click)="saveVehicle(vehicle)"`

`(focus)="log('focus')"`

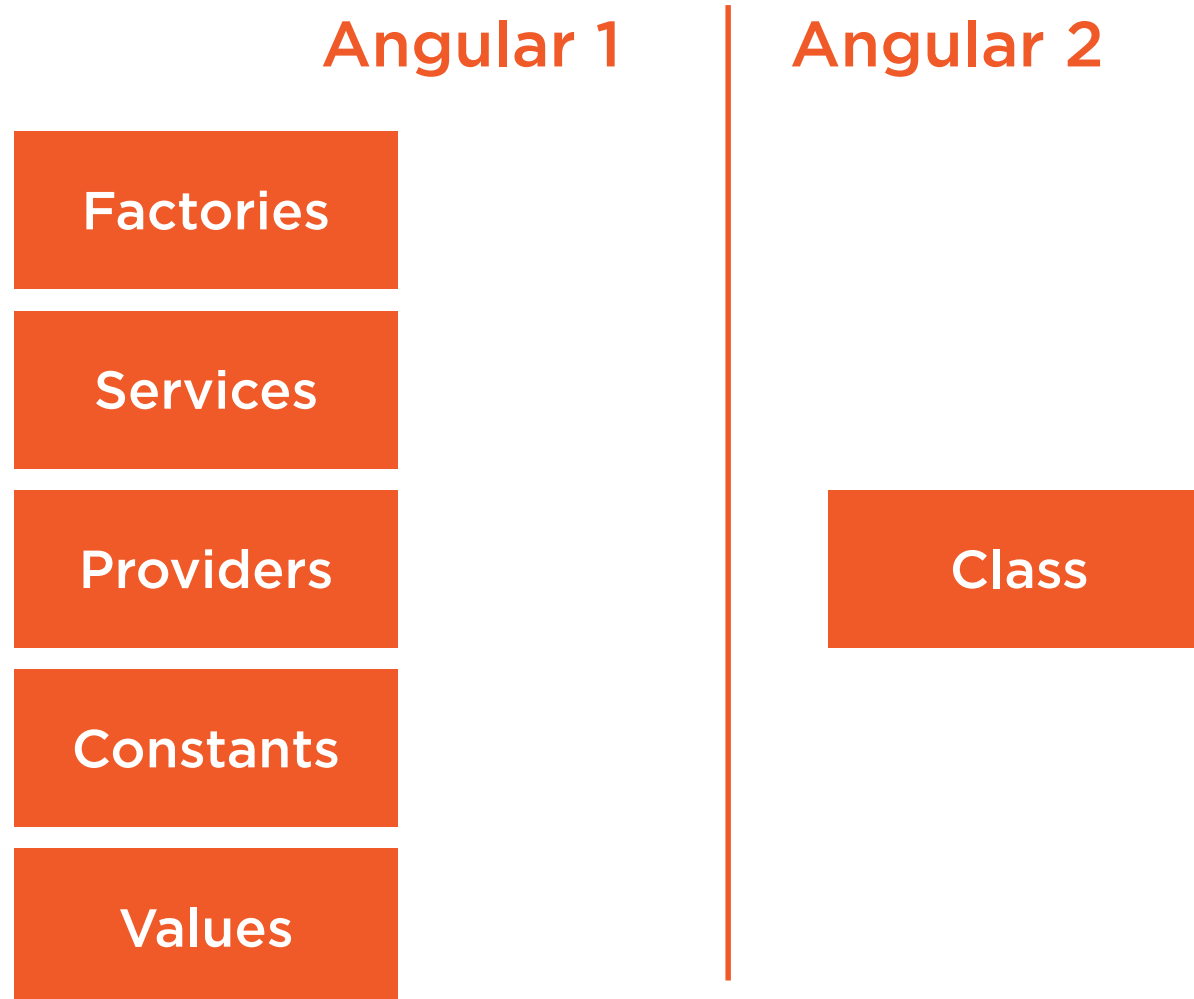
`(blur)="log('blur')"`

`(keyup)="checkValue()"`



6

Services



Creating Services

Angular 1

```
angular
  .module('app')
  .service('VehicleService', VehicleService);

function VehicleService() {
  this.getVehicles = function () {
    return [
      { id: 1, name: 'X-Wing Fighter' },
      { id: 2, name: 'Tie Fighter' },
      { id: 3, name: 'Y-Wing Fighter' }
    ];
  }
}
```

Angular 2

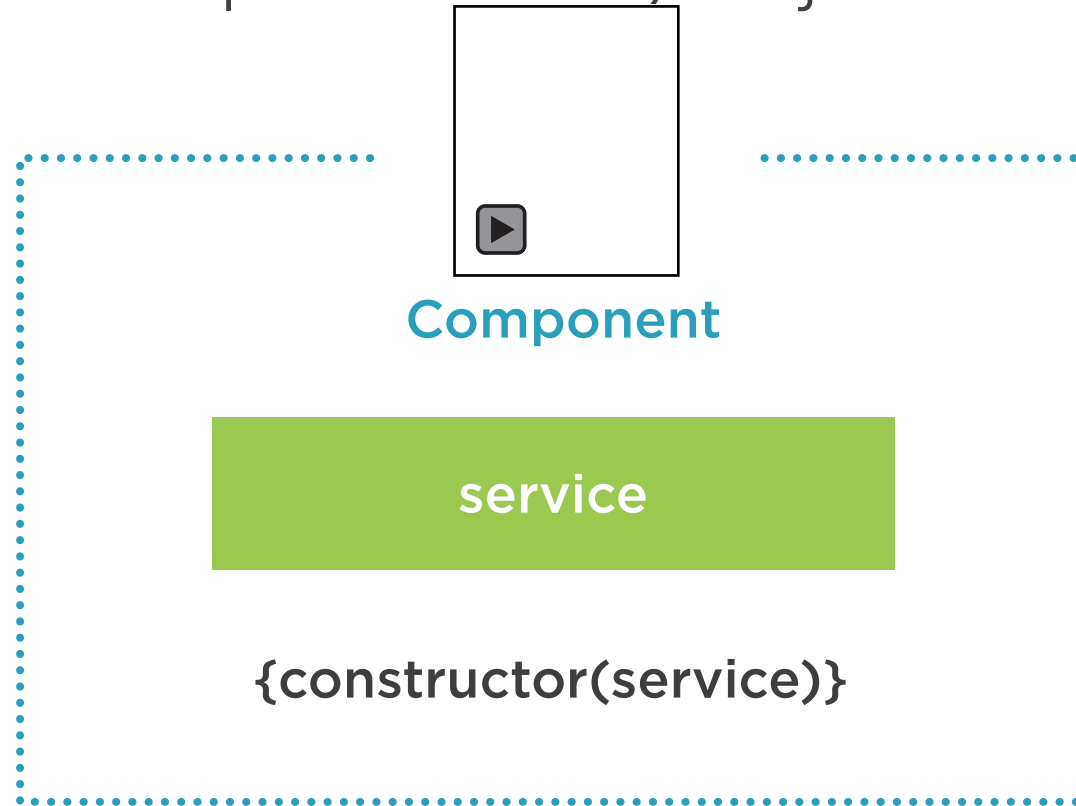
```
import {Injectable} from 'angular2/core';

@Injectable()
export class VehicleService {
  getVehicles = () => [
    { id: 1, name: 'X-Wing Fighter' },
    { id: 2, name: 'Tie Fighter' },
    { id: 3, name: 'Y-Wing Fighter' }
  ];
}
```



7

Dependency Injection



Registering Services with the Injector

Registration

Angular 1

```
angular
  .module('app')
  .service('VehicleService', VehicleService);

function VehicleService() {
  this.getVehicles = function () {
    return [
      { id: 1, name: 'X-Wing Fighter' },
      { id: 2, name: 'Tie Fighter' },
      { id: 3, name: 'Y-Wing Fighter' }
    ];
  }
}
```

Registration

Angular 2

```
import { VehicleService } from './vehicle.service';

@Component({
  selector: 'my-vehicles',
  templateUrl: 'app/vehicles.component.html',
  providers: [VehicleService]
})
export class VehiclesComponent {
  constructor(
    private _vehicleService: VehicleService) { }
  vehicles = this._vehicleService.getVehicles();
}
```

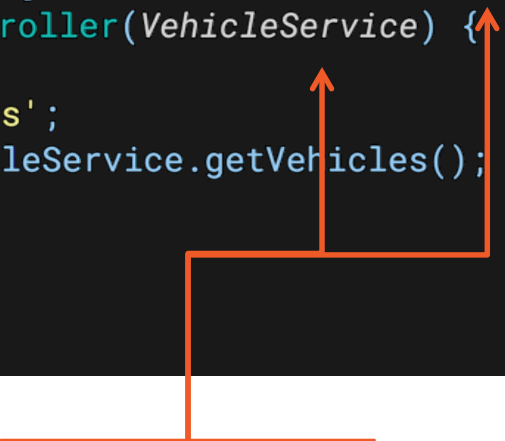


Dependency Injection

Angular 1

```
angular
  .module('app')
  .controller('VehiclesController', VehiclesController);

VehiclesController.$inject = ['VehicleService'];
function VehiclesController(VehicleService) {
  var vm = this;
  vm.title = 'Services';
  vm.vehicles = VehicleService.getVehicles();
}
```

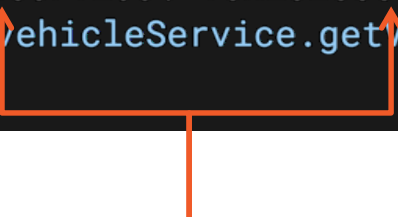


Injection

Angular 2

```
import { VehicleService } from './vehicle.service';

@Component({
  selector: 'my-vehicles',
  templateUrl: 'app/vehicles.component.html',
  providers: [VehicleService]
})
export class VehiclesComponent {
  constructor(
    private _vehicleService: VehicleService) { }
  vehicles = this._vehicleService.getVehicles();
}
```



Injection





Removal of
Many
Directives

Components

Structural
Directives

Bootstrapping

Services

Data Binding

Dependency
Injection

Angular 1 to Angular 2

7 Key Comparisons

