

Angular 2.0

Brighter future?

Eugene Zharkov

*Angular momentum image

JavaScript Learning Curves



jQuery



Node.js



AngularJS

2.0 source code



Angular 1.3 support
~2 years after 2.0 release

Why I hate Angular?

- Routing
- Directives
- Binding
- Scope
- Scope
- RootScope
- Continue...

AtScript

AtScript

- Superset of ES6
- TypeScript is my friend
- Type, Field, Metadata Annotations

Type Annotations

AtScript

```
class MyClass {  
  methodA(name:string):int {  
    var length:int = name.length;  
    return length;  
  }  
}
```

ES6

```
class MyClass {  
  methodA(name) {  
    var length = name.length;  
    return length;  
  }  
}
```


Generics

AtScript

```
class MyClass {
    methodA(names:List<string>):List<int> {
        var sizes = [];
        for(var i = 0; i < names.length; i++) {
            sizes[i] = names[i].length;
        }
        return sizes;
    }
}
```

ES6

```
import * as rtts from 'rtts';

class MyClass {
    methodA(names) {
        rtts.types(names, Array.of(rtts.string));
        var sizes = [];
        for(var i = 0; i < names.length; i++) {
            sizes[i] = names[i].length;
        }
        return rtts.returnType(
            sizes,
            Array.of(rtts.int));
    }
}
```

Type Introspection

AtScript

```
@Component()
class MyApp {
  server:Server;
  @Bind('name') name:string;
  @Event('foo') fooFn:Function;

  @Inject()
  constructor(@parent server:Server) {}

  greet():string {}
}
```

ES5

```
function MyApp() {}

MyApp.properties = {
  'server': { is: Server },
  'name': { is:string,
            annotate: [new Bind('name')] },
  'fooFn': { is:Function,
              annotate:[new Event('foo')] }
}

MyApp.annotate = [
  new Component(),
  new Inject()
];

MyApp.parameters = [
  {is:Server, annotate:[parent]}
];

MyApp.prototype.greet = function() {}
MyApp.prototype.greet.returns = string;
```

You can skip AtScript and
use CoffeeScript,
TypeScript, ES6, ES5

DI

Annotations

```
MyComponent.parameters = [{is:Server}];
```

Instance Scope

@TransientScope

```
export class MyClass { ... }
```

DI will always create a new instance of a class, every time you ask for one

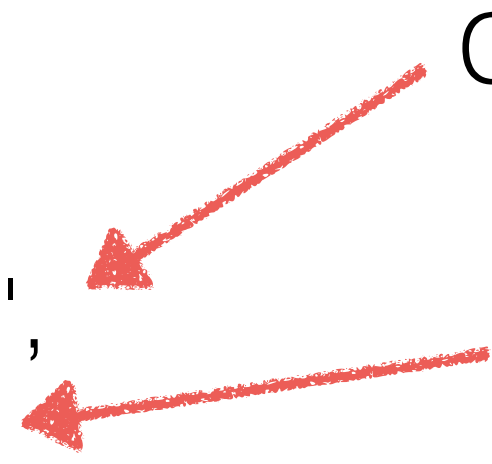
Child Injectors

A child injector inherits from its parent all of its parent's services, but it has the ability to override them at the child level.

Directives

- Component Directive
- Decorator Directive (ng-show)
- Template Directive (ng-if || ng-repeat)


```
@ComponentDirective({  
  selector:'tab-container',  
  directives:[NgRepeat]  
})  
export class TabContainer {  
  constructor(panes:Query<Pane>) {  
    this.panes = panes;  
  }  
  
  select(selectedPane:Pane) { ... }  
}
```



CSS selector

Dependencies

such wow

no \$scope

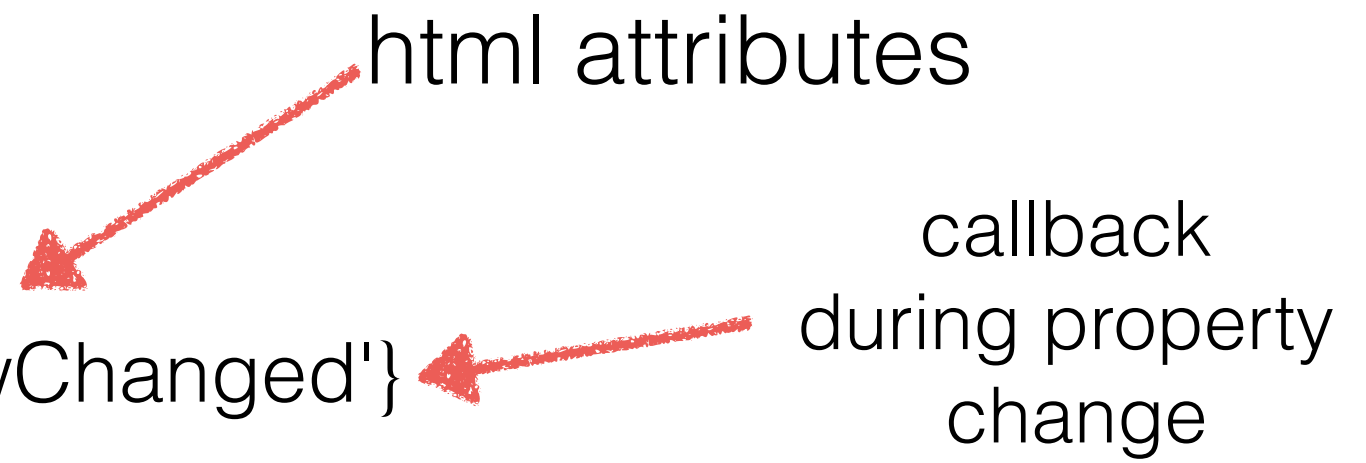
direct access in the template

```
@DecoratorDirective({
  selector:'[ng-show]',
  bind: { 'ngShow': 'ngShow' },
  observe: {'ngShow': 'ngShowChanged'}
})
export class NgShow {
  constructor(element:Element) {
    this.element = element;
  }

  ngShowChanged(newValue){
    if(newValue){
      this.element.style.display = 'block';
    }else{
      this.element.style.display = 'none';
    }
  }
}
```

html attributes

callback during property change



```
@TemplateDirective({
  selector: '[ng-if]',
  bind: {'ngIf': 'ngIf'},
  observe: {'ngIf': 'ngIfChanged'}
})
export class NgIf {
  constructor(viewFactory:BoundViewFactory, viewPort:ViewPort) {
    this.viewFactory = viewFactory;
    this.viewPort = viewPort;
    this.view = null;
  }

  ngIfChanged(value) {
    if (!value && this.view) {
      this.view.remove();
      this.view = null;
    }

    if (value) {
      this.view = this.viewFactory.createView();
      this.view.appendTo(this.viewPort);
    }
  }
}
```

CSS selector

html attributes

callback during property change

represents the location in the DOM

instantiates the template itself

```
@ComponentDirective({
  selector:'tab-container',
  directives:[NgRepeat]
})
export class TabContainer {
  constructor(panes:Query<Pane>) {
    this.panes = panes;
  }

  select(selectedPane:Pane) { ... }
}
```

```
<template>
  <div class="border">
    <div class="tabs">
      <div [ng-repeat|pane]="panes" class="tab" (^click)="select(pane)">
        <img [src]="pane.icon"><span>${pane.name}</span>
      </div>
    </div>
    <content></content>
  </div>
</template>
```

pane - variable name

(^click) - event handler

```
<template>
  <div class="border">
    <div class="tabs">
      <div [ng-repeat|pane]="panes" class="tab" (^click)="select(pane)">
        <img [src]="pane.icon"><span>${pane.name}</span>
      </div>
    </div>
    <content></content>
  </div>
</template>
```

[] - attribute binding

\${} - content as a string

Router

- Shiny new toy
- Child Routers
- Screen Activation

Angular developers during 2.0 release



UNANSWERED QUESTIONS

**UNANSWERED QUESTIONS
EVERYWHERE**