

Dart, it's Angular too!

Pierre Reliquet – Zenika

Twitter : @preliquet

Github : PierreReliquet





Breizh C@mp



Why Dart ?

Because it's not



JavaScript

What's Dart

- Web programming language
- Both client and server side
- Open source
- Active community
- Developed by Google

Dart's goals

- Create a whole ecosystem
 - ◆ Pub
 - ◆ Dart editor + plugins
 - ◆ Dartium, dart2js
- Easy to learn
- High productivity & [performances](#)



Simple syntax

```
class BzhCamp {  
    ...  
}
```

Simple syntax

```
class BzhCamp {  
    List<Attendee> attendees = []; // var attendees = [];  
    ...  
}
```


Simple syntax

```
class BzhCamp {  
    List<Attendee> attendees = []; // var attendees = [];  
    BzhCamp(this.attendees);  
    ...  
}
```

Simple syntax

```
class BzhCamp {  
    List<Attendee> attendees = []; // var attendees = [];  
    BzhCamp(this.attendees);  
    BzhCamp operator +(BzhCamp other) {  
        return new BzhCamp(this.attendees+other.attendees);  
    }  
    ...  
}
```

Simple syntax

```
class BzhCamp {  
    List<Attendee> attendees = []; // var attendees = [];  
    BzhCamp(this.attendees);  
    BzhCamp operator +(BzhCamp other) {  
        return new BzhCamp(this.attendees+other.attendees);  
    }  
    void askQuestions({int number: 1}) {  
        ...  
    }  
}
```

Simple syntax

```
class BzhCamp {  
    List<Attendee> attendees = []; // var attendees = [];  
    BzhCamp(this.attendees);  
    BzhCamp operator +(BzhCamp other) {  
        return new BzhCamp(this.attendees+other.attendees);  
    }  
    void askQuestions({int number: 1}) {  
        ...  
    }  
    String toString() => "${attendees.length} attendees participated to the event";  
}
```

Simple syntax

@Injectable

class BzhCamp {

List<Attendee> attendees = []; *// var attendees = [];*

BzhCamp(**this**.attendees);

BzhCamp **operator** +(BzhCamp other) {

return new BzhCamp(**this**.attendees+other.attendees);

}

void askQuestions({**int** number: 1}) {

...

}

String toString() => "\${attendees.length} attendees participated to the event";

}

Dart is not JS

- No undefined just null
- Only true is truthy
- Real scopes
- Future is clean

‘Callback Hell’ aka ‘Pyramid of Doom’!

```
myapp.asyncOperation1(function(firstData) {  
  // Success  
  myapp.asyncOperation2(firstData, function(secondData) {  
    // Success  
    myapp.asyncOperation3(secondData, function(thirdData) {  
      myapp.treatData(thirdData);  
    }, function(err) {  
      // Error handling  
    });  
  }, function(err) {  
    // Error handling  
  });  
}, function(err) {  
  // Error Handling  
});
```

Future<Bright>

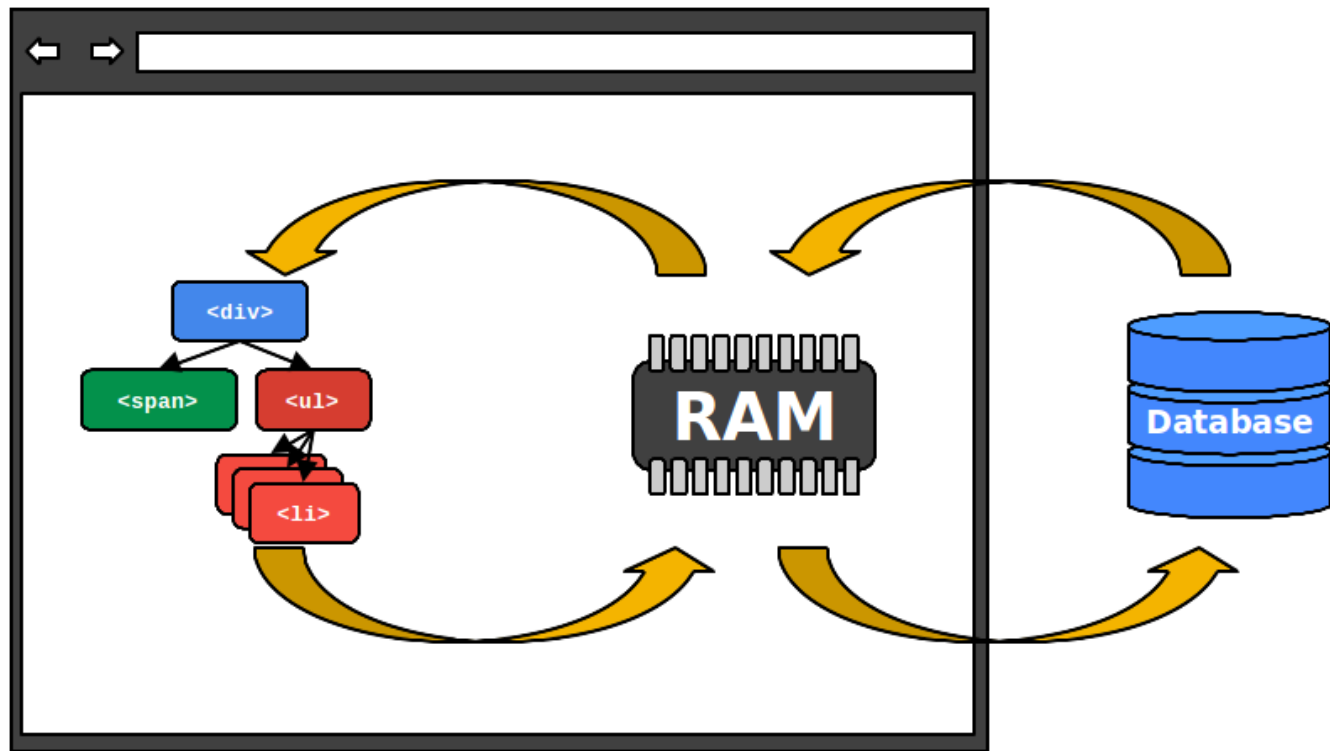
- Dart includes Future since first release to annihilate callback hell :

asyncOp1()

```
.then((var firstData) => asyncOp2(firstData))  
.then((var secondData) => asyncOp3(secondData))  
.then((var thirdData) => treatData(thirdData))  
.catchError((e) => print(e));
```




Philosophy



Imperative vs Declarative

```
<div id="mydiv">
  <h1>Foo</h4>
  <div>Bar</div>
</div>
```

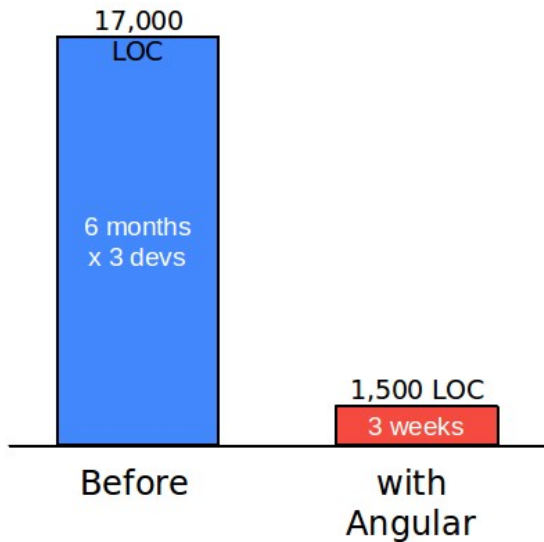
```
-
$('#mydiv').on('click', function() {
  console.log('foo');
});
```

```
<div ng-click="log()">
  <h1>Foo</h4>
  <div>Bar</div>
</div>
```

```
--
function log() {
  console.log('foo'
}
```

Simplicity & concision

Googlefeedback





ANGULARDART

Why AngularDart?



Who ?

- Angular core team
- Dart team
- Green Tea team
- community



More than a port
Philosophy + Syntax + Dartisms

[from James deBoer's talk](#)

All my beloved directives

```
<div class="contact-card span4" ng-repeat="contact in
contactList.contacts | searchFilter:search">
  <div class="contact-card-inner">
    <h4>{{contact.firstName}} {{contact.lastName}} </h4>
    <div class="contact-address">{{contact.address}} </div>
    <div class="contact-phone">{{contact.phone}} </div>
  </div>
</div>
```

Snippet : a JS template

```
<html ng-app>
  <head>...</head>
  <body ng-controller='ContactController as ctrl'>
    <h1>Hello World</h1>
    <p>Hello {{ctrl.name}} from AngularJS!</p>
    <input type="text" ng-model="ctrl.name"></input>
    <script type="text/javascript" src="angular.js"></script>
    <script type="text/javascript" src="controllers.js"></script>
  </body>
</html>
```

Snippet : a Dart template

```
<html ng-app>
  <head>...</head>
  <body ctrl>
    <h1>Hello World</h1>
    <p>Hello {{ctrl.name}} from AngularDart!</p>
    <input type="text" ng-model="ctrl.name"></input>
    <script type="application/dart" src="helloworld.dart"></script>
    <script type="text/javascript" src="packages/browser/dart.js"></script>
  </body>
</html>
```

Snippet : JS bootstrapping

```
angular.module('MyApp', []) // creating a new module  
  .controller('HelloWorldCtrl', function() {// registering a controller  
    ...  
  }
```

Snippet : Dart bootstrapping

```
void main() { // everything start in a main method  
  applicationFactory()  
    ..addModule(new MyApp())  
    ..run();  
}  
class MyApp extends Module {  
  MyApp() {  
    bind>HelloWorldCtrl);  
  }  
}
```

Snippet : controller

```
@Controller(selector: "[ctrl]", publishAs: "ctrl")
```

```
class HelloWorldCtrl {
```

```
    String name = "Breizhcamp";
```

```
    Http _http;
```

```
    HelloWorldCtrl(this._http);
```

```
}
```

Angular JS naming evolved

- Directives became
 - **Components** : create custom tags to enhance HTML
 - **Decorators** : add behaviour to existing HTML tag
- Filters became
 - **Formatters**

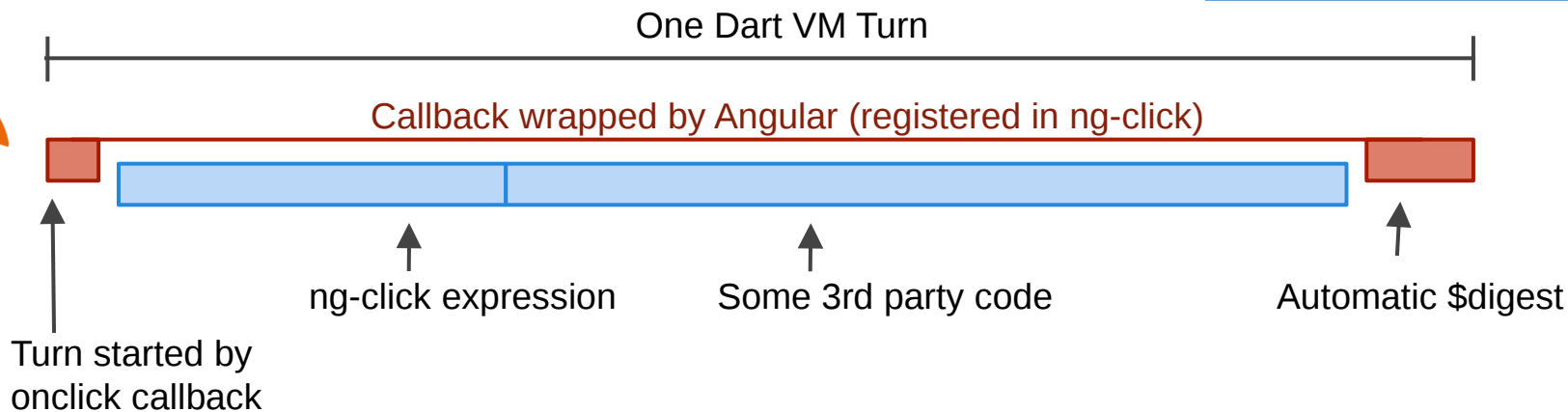
Dartisms

- Type based DI : [Vojta Jina's talk](#)
- Annotations
- Web standards
 - ◆ Shadow dom
- Zones : [Brian Ford's talk](#)

Zone ? What's that ?

- Simple def : A zone is an execution context
- Real def : dynamic extent **including asynchronous callbacks** declared in the zone
- No more `$scope.$apply()`;

[from James deBoer's talk](#)



AngularDart

Version 1.0 is coming

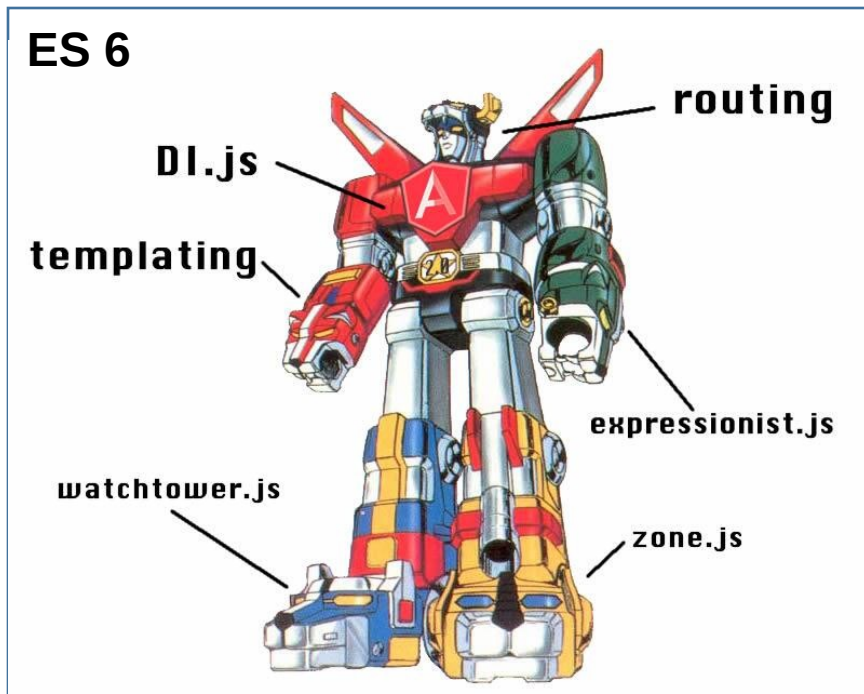
→ Five issues to solve on the GitHub milestone

AngularDart is awesome, Google experienced it!

What if that talk was not really about AngularDart?

AngularJS 2.0

→ Many learnings from AngularDart



ECMAScript 6 ~



- Classes
- Modules
- Inheritance
- Annotations ?

- Become familiar with ES6 through :
 - Dart
 - Traceur

ECMAScript 6 – snippet

```
import {  
  ChangeDetector  
} from './change_detection.js';  
  
export class DirtyCheckingChangeDetectorGroup extends ChangeDetector {  
  constructor(parent, cache) {  
    this._parent = parent;  
    this._getterCache = cache;  
  }  
  watch(context, field, handler) { ... }  
}
```

Snippet from [watchtower.js](https://github.com/GoogleChrome/watchtower.js)

Just a thought

AngularDart is not about **Dart**.
Such as **AngularJS** is not about **JS**.

It is all about making the web a better development platform.

Thank you!

Questions ?

N'oubliez pas le feedback en sortant !

✓ ou ✗ ?