

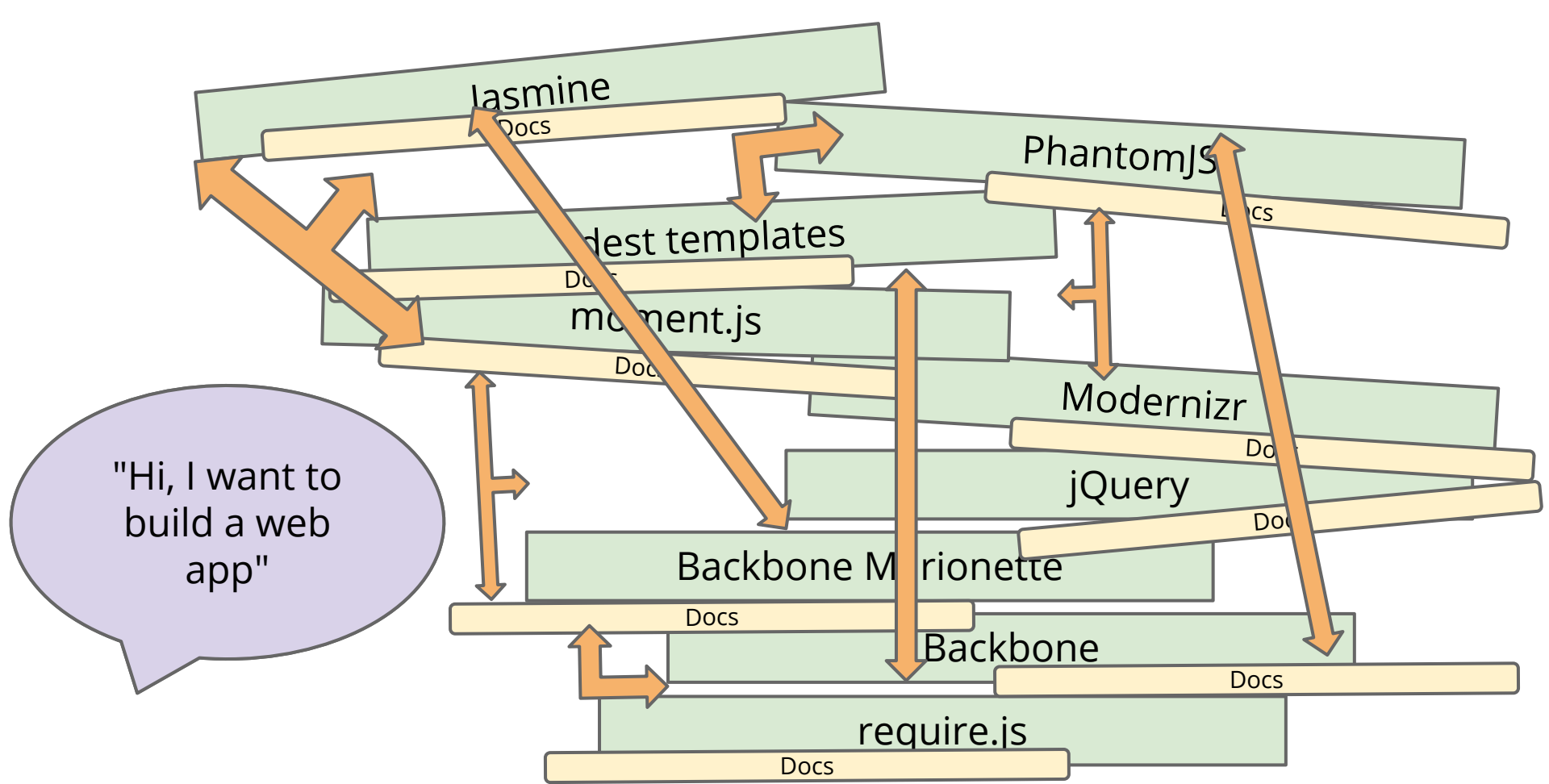


ANGULARDART

JFokus 2014

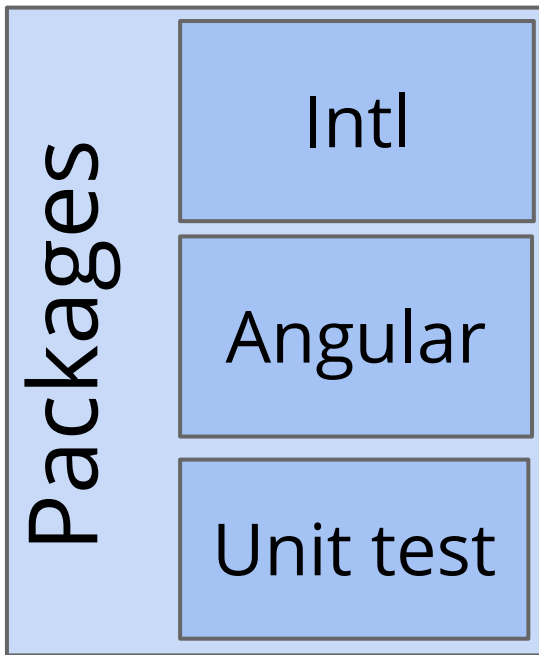


#dartlang #angular



#dartlang #angular

"Things are
consistent and
clear."



Dart SDK

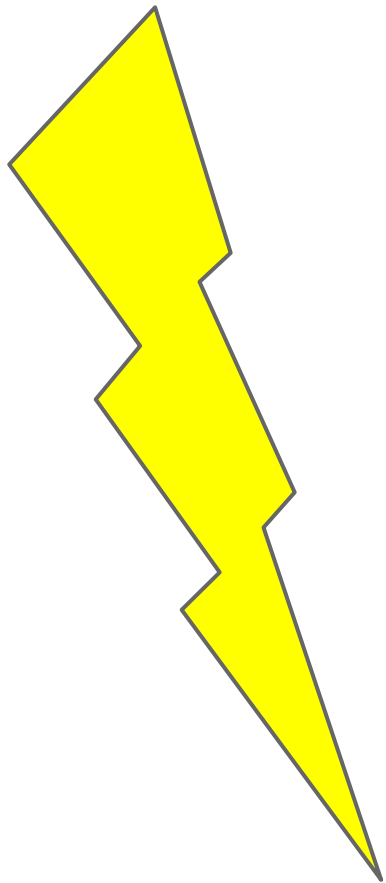




ANGULAR

"We rewrote Google's internal CRM app in 6 months, from scratch, with Dart and Angular."

- Internal team at Google



Dart Lightning Tour



#dartlang #angular



#dartlang #angular



#dartlang #angular



Dart Principles



Productivity

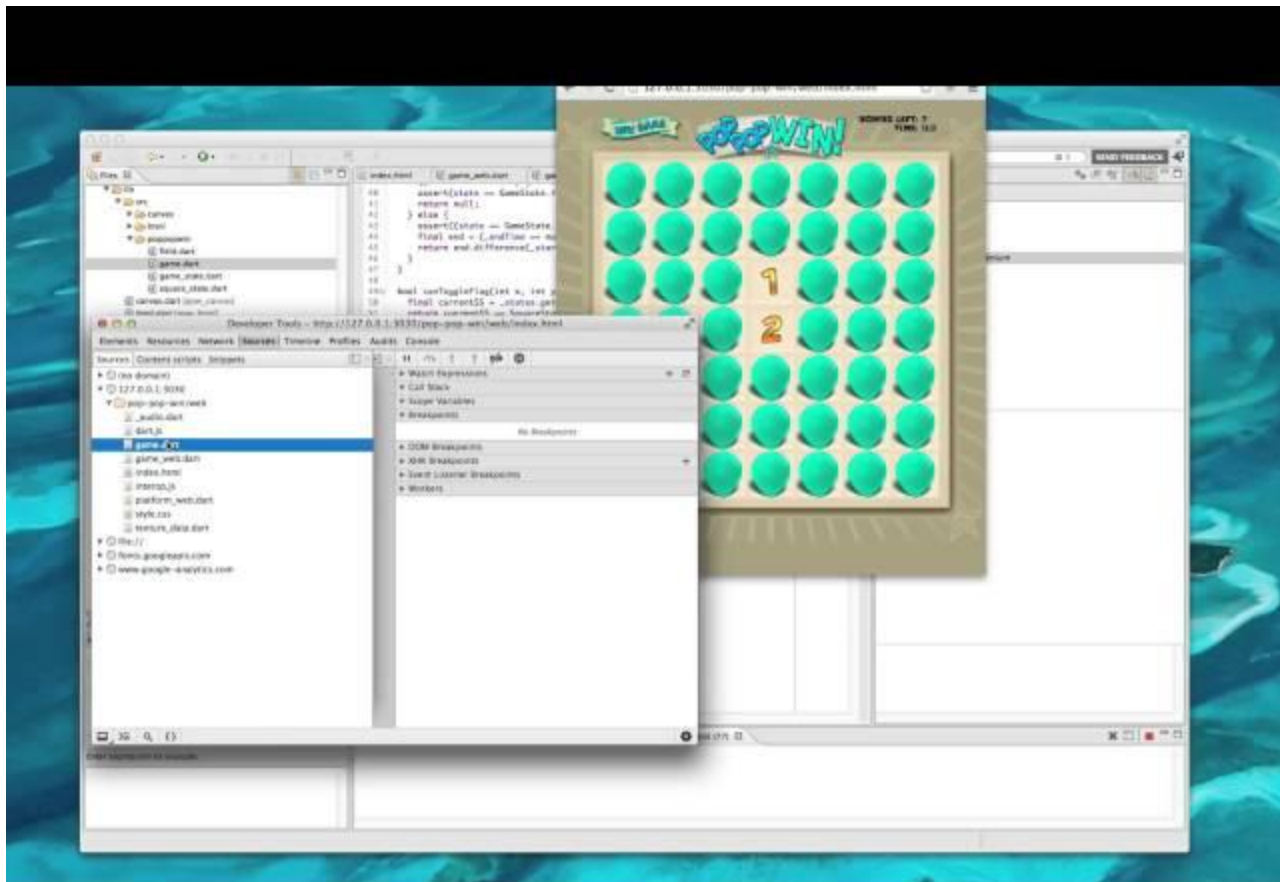


Performance



Scalability





#dartlang #angular

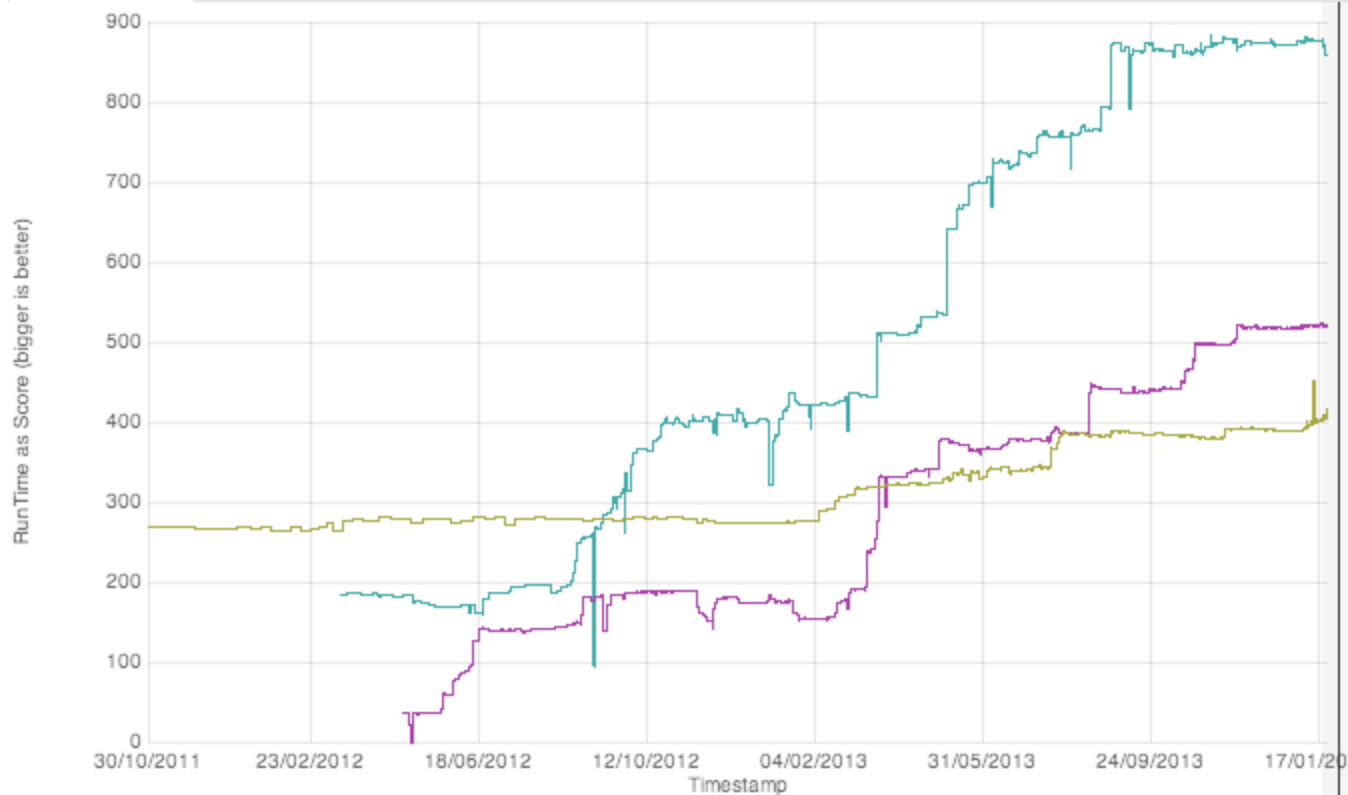


DeltaBlue

FluidMotion

Richards

Tracer

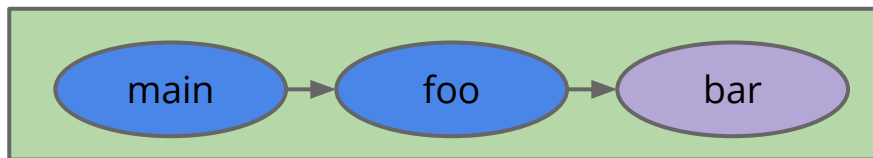
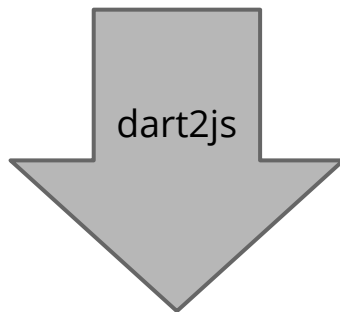
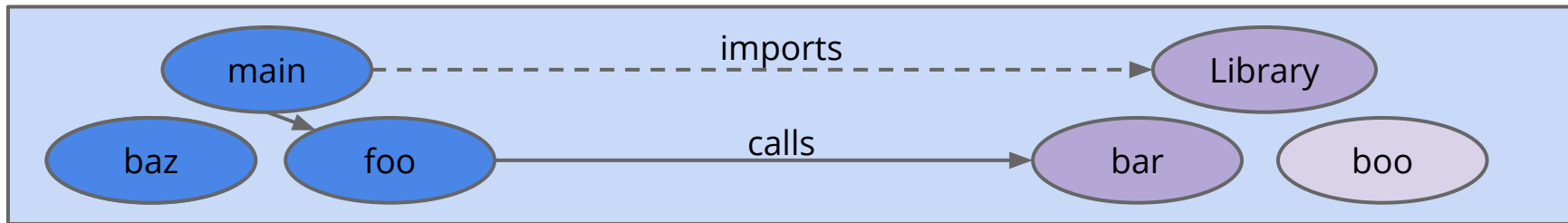


dart(31985)	861.50
dart2js: v8(31985)	523.24
js: v8(18829)	416.00



#dartlang #angular

Tree shaking



Simple syntax, ceremony free

```
class Hug {
```



#dartlang #angular

Simple syntax, ceremony free

```
class Hug {  
  num strength;  
  Hug(this.strength);  
}
```



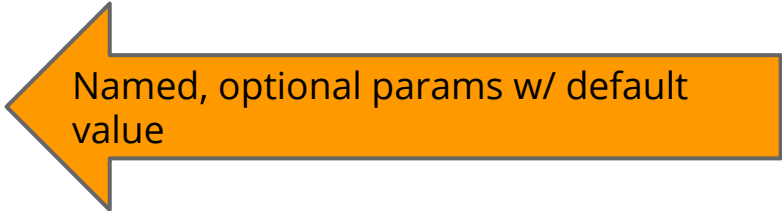
Simple syntax, ceremony free

```
class Hug {  
  num strength;  
  Hug(this.strength);  
  
  Hug operator +(Hug other) {  
    return new Hug(strength + other.strength);  
  }  
}
```



Simple syntax, ceremony free

```
class Hug {  
  num strength;  
  Hug(this.strength);  
  
  Hug operator +(Hug other) {  
    return new Hug(strength + other.strength);  
  }  
  
  void patBack({int hands: 1}) {  
    // ...  
  }  
}
```



Named, optional params w/ default value



Simple syntax, ceremony free

```
// ...
```

```
Hug operator +(Hug other) {  
    return new Hug(strength + other.strength);  
}
```

```
void patBack({int hands: 1}) {  
    // ...  
}
```

One-line function

```
String toString() => "Embraceometer reads $strength";  
}
```



Simple syntax, ceremony free

```
// ...
```

```
Hug operator +(Hug other) {  
    return new Hug(strength + other.strength);  
}
```

```
void patBack({int hands: 1}) {  
    // ...  
}
```

```
String toString() => "Embraceometer reads $strength";
```

```
}
```



String interpolation



#dartlang #angular

Simple syntax, ceremony free

```
main() {  
    var hug = new Hug(10);  
  
}
```



Simple syntax, ceremony free

```
main() {  
  var hug = new Hug(10);  
  var bigHug = hug + new Hug(100);  
  
}
```



Simple syntax, ceremony free

```
main() {  
    var hug = new Hug(10);  
    var bigHug = hug + new Hug(100);  
    bigHug.patBack(hands: 2);  
}
```



Simple syntax, ceremony free

```
main() {  
    var hug = new Hug(10);  
    var bigHug = hug + new Hug(100);  
    bigHug.patBack(hands: 2);  
    print(bigHug); // Embraceometer reads 110  
}
```



Method Cascades

```
var button = new ButtonElement();  
button.id = 'fancy';  
button.text = 'Click Point';  
button.classes.add('important');  
button.onClick.listen((e) => addTopHat());  
  
parentElement.children.add(button);
```



Method Cascades

```
var button = new ButtonElement()  
  ..id = 'fancy'  
  ..text = 'Click Point'  
  ..classes.add('important')  
  ..onClick.listen((e) => addTopHat());  
  
parentElement.children.add(button);
```



Callable objects

```
class Registrator {  
    doIt(User user) { ... }  
}  
  
var reg = new Registrator();  
reg.doIt(user); // implied!
```

```
class Registrator {  
    call(User user) { ... }  
}  
  
var reg = new Registrator();  
reg(user); // just call it
```

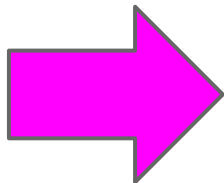


Annotations

```
@Extreme(ftw: true)
class Awesome {
  @upHigh int highFives;
}
```



Clean semantics and behavior



#dartlang #angular

Clean semantics and behavior

Examples:

- Only *true* is truthy
- There is no *undefined*, only *null*
- No type coercion with `==`, `+`
- Throw error on missing method



Scope of this?

```
class AwesomeButton {
```

```
  AwesomeButton(button) {
```

```
    button.onClick.listen((Event e) => this.atomicDinosaurRock());
```

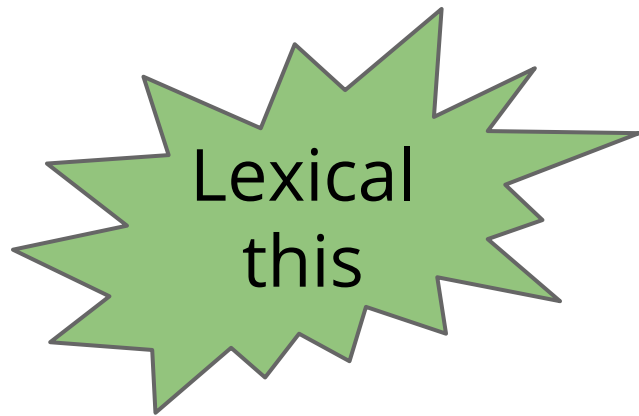
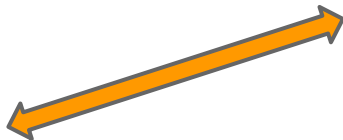
```
  }
```

```
  atomicDinosaurRock() {
```

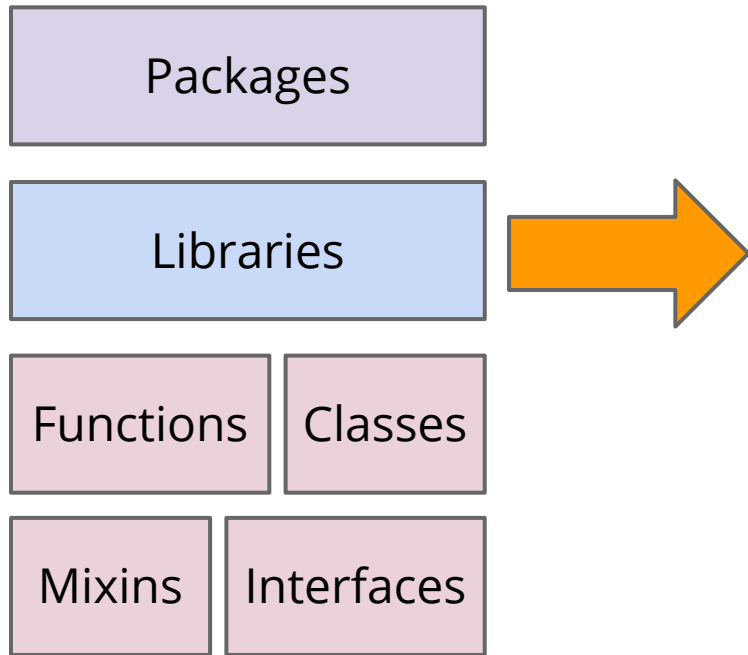
```
    /* ... */
```

```
  }
```

```
}
```



Scalable structure



```
library games;
```

```
import 'dart:math';
```

```
import 'players.dart';
```

```
class Darts {
```

```
  // ...
```

```
}
```

```
class Bowling {
```

```
  // ...
```

```
}
```

```
Player findOpponent(int skillLevel) {
```

```
  // ...
```

```
}
```

#dartlang #angular



Scary Async

```
catService.getCatData("cute", (cat) {  
  catService.getCatPic(cat.imageId, (pic) {  
    imageWorker.rotate(pic, 30, (rotated) {  
      draw(rotated);  
    });  
  });  
});
```



4 levels
deep!



The Future looks bright

```
catService.getCat("cute") // returns a Future  
  .then((cat) => catService.getCatPic(cat.imageId))  
  .then((pic) => imageWorker.rotate(pic, 30))  
  .then((rotated) => draw(rotated))  
  .catchError((e) => print("Oh noes!"));
```





NGULARDART



#dartlang #angular

Angular Principles



D.R.Y.



Structure

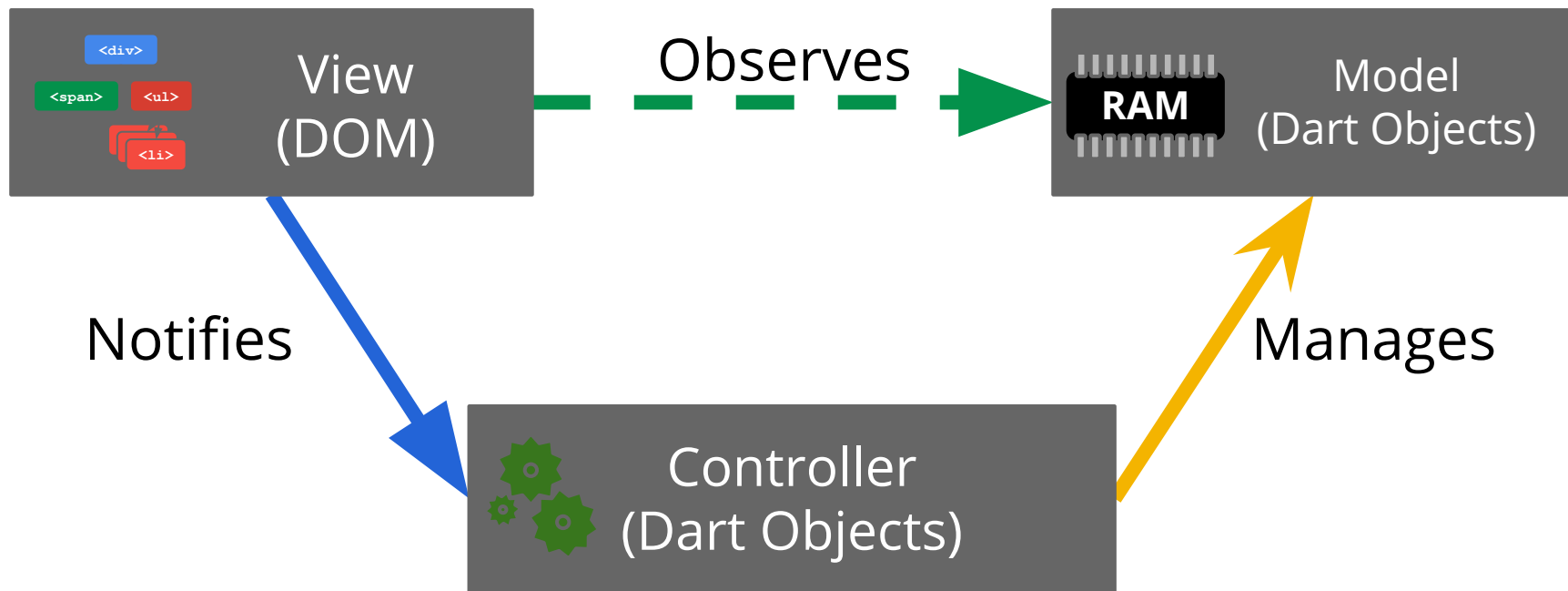


Testability



#dartlang #angular

Structure: Model-View-Controller



Model:

```
class Person { String name; }
```

View:

```
<div person>
  Hi, {{ctrl.person.name}}
</div>
```

Controller:

```
@NgController(
  selector: '[person]',
  publishAs: 'ctrl'
)
class Controller {
  Person person = new Person();

  Controller() {
    person.name = 'Bob';
  }
}
```

(demo controller)

#dartlang #angular



Controllers

- Expose model to view
- Handle user interaction
- Update model to change the view



Expressions

- {{ The stuff inside your markup }}
- Like Dart, but eval'd against the current scope
- No complex logic like loops or throws



Expression Examples

- `person.name`
- `shoppingCart.total()`
- `cart.total() - user.discountRate`



Feature: Data Binding

hello.dart

```
person.name = 'Bob';
```

hello.html

```
{{ person.name }}           // 1-way  
<input ng-model='person.name'> // 2-way
```



(demo simple_hello_world)

#dartlang #angular

Note: Binding for initial page load

Markup fine except for first page

```
<p>{{someObj.something}}</p>
```

Use ng-cloak to avoid unstyled content on load

```
<p ng-cloak>{{someObj.something}}</p>
```

Attribute form also avoid unstyled content on load

```
<p ng-bind='someObj.something'></p>
```



Feature: Directives

```
<div ng-repeat='item in cart.items'>  
  <span ng-bind='item.name'></span>  
  <button ng-click='cart.delete($index)'>  
    Delete  
  </button>  
</div>
```



(demo repeat_bind_click)

#dartlang #angular

Note: Directive validation

```
<div data-ng-repeat='item in cart.items'>  
  <span data-ng-bind='item.name'></span>  
  <button data-ng-click='cart.delete($index)'>  
    Delete  
  </button>  
</div>
```



View Templates

Directives

Markup {{ }}

Automatic sanitization

Filters



Feature: automatic sanitization

`<script>can't hack me` Hello, `<script>can't hack me</script>`

```
String userInput = "<script>I CAN HAZ XSS</script>" +  
    "<p>I am safe</p>";  
querySelector('#contents').innerHTML = userInput;
```

// Result:

```
<div id="contents">  
    <p>I am safe</p>                <!-- No <script> tag -->  
</div>
```



Filters

```
<p>{{ name | reverse }}</p>
```

Filters can take parameters

```
<p>{{ amount | currency:us }}</p>
```

Filters can chain

```
<p>{{ msg | embiggen | shoutify }}</p>
```



Filter example: Reverse

```
@NgFilter(name: 'reverse')
class Reverser {
  String call(String input) {
    if (input == null) return '';
    return input.split('').reversed.join('');
  }
}
```



Feature: Dependency Injection

Definition

```
@NgInjectableService() class Users { ... }
```

Usage

```
class Login {  
  Users users;  
  Login(this.users);  
}
```

```
module..type(Users)..type(Login);
```



(demo service_injection)

[#dartlang](#) [#angular](#)

Principle: Eliminate boilerplate

Dependency Injection

PODO Models, Controllers, etc

No DOM in most code paths



Feature: Injectable HTTP client

```
@InjectableService()  
class Users {  
  Http http;  
  
  Users(this.http);  
  
  Future<List<User>> getUsers() {  
    return http.get('users.json').then((resp) {  
      return (resp.data as List).map((u) => new User.fromJson(u)).toList();  
    });  
  }  
}
```

```
[  
  {"username": "Bob"},  
  {"username": "Alice"}  
]
```

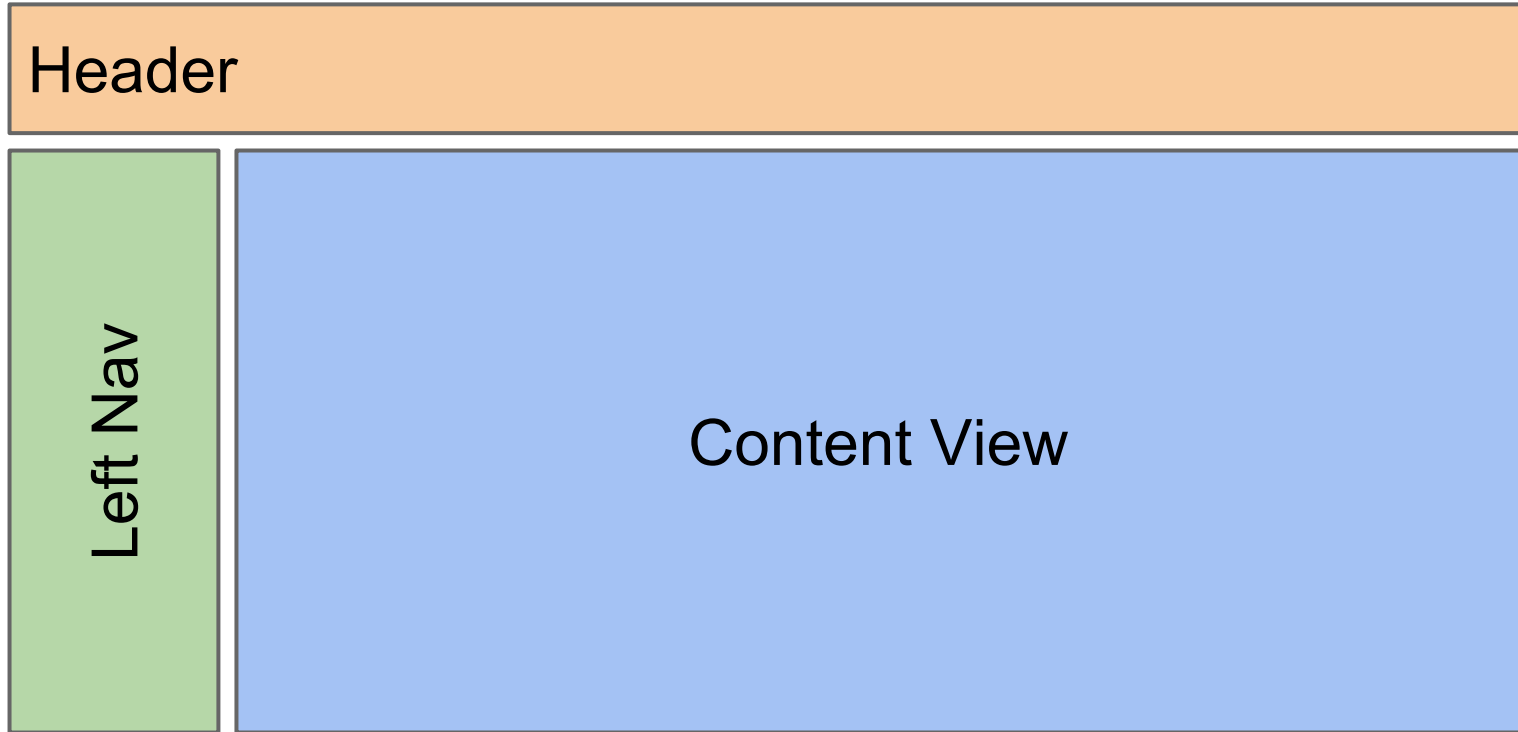
HTTP client aware of response types.



(demo http)

#dartlang #angular

Feature: Routes



Routes

`<ng-view>` directive as content placeholder

RouteInitializer configures:

- Controller
- Template
- URL



Routes

```
class AppRouteInitializer implements RouteInitializer {  
  init(Router router, ViewFactory view) {  
    router.root  
      ..addRoute(  
        name: 'overview',  
        path: '/overview',  
        defaultRoute: true,  
        enter: view('views/overview.html'))  
      ..addRoute(name: 'analytics',  
        path: '/analytics',  
        enter: view('views/analytics.html'));  
  }  
}
```



Listening to Route changes

```
@NgDirective(  
  selector: '[active-when-route]'  
)  
class CurrentRoute {  
  Router router;  
  Element element;  
  
  CurrentRoute(this.element, this.router) {  
    router.onRouteStart.listen((e) {  
      toggleActive();  
    });  
  }  
}
```

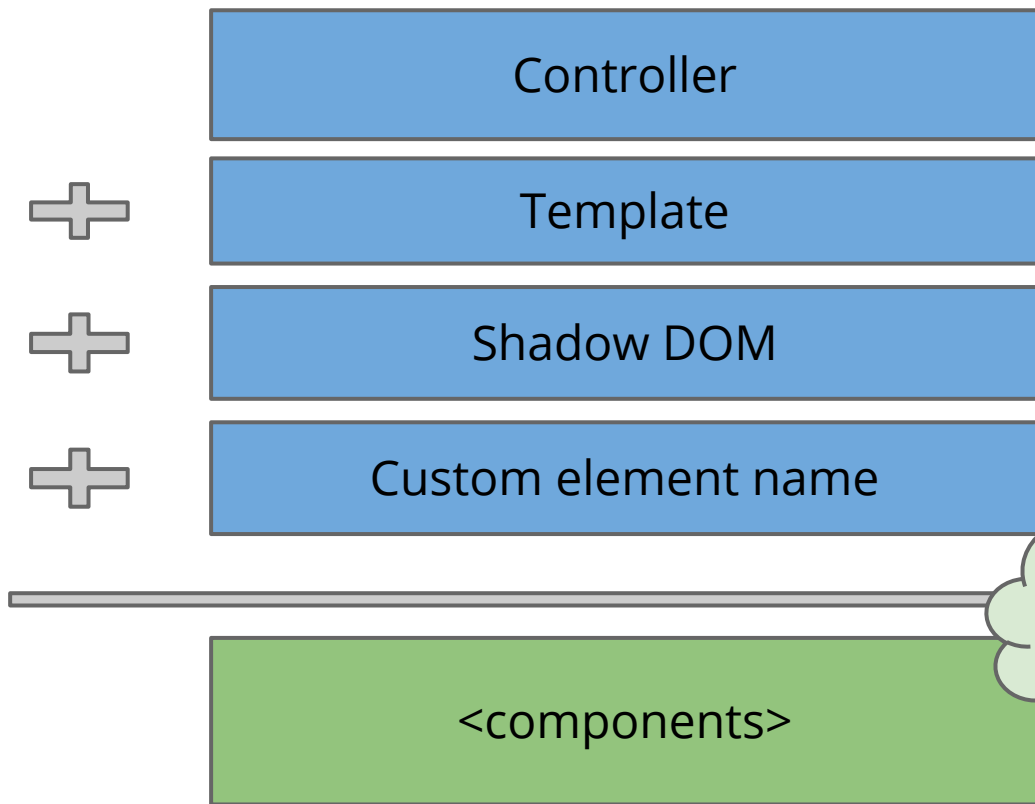


(demo routing)

#dartlang #angular

Feature: Components

First appearance
in AngularDart



aka "web
components
light"

#dartlang #angular



Shadow DOM in action

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <p>
      <video src="BigBuckBunny.ogv" poster="BigBuckBunny.png"
        height="360" controls="true" type="video/ogg">
        #document-fragment
        <div>
          <div style="display: none;"></div>
          <div>
            <div style="-webkit-transition: opacity 0.1s; t
              opacity 0.1s; opacity: 1;">
              <input type="button">
              <input type="range" precision="float" max="59
                <div style>0:27</div>
                <div style="display: none;">9:56</div>
              <input type="button">
              <input type="range" precision="float" max="1"
              <input type="button" style="display: none;">
              <input type="button" style>
            </div>
          </div>
        </div>
        <i>You need a HTML5 <video> capable browser to view
      </video>
    </p>
```

Components

```
@NgComponent(  
  selector: 'rating',  
  templateUrl: 'packages/ng_rating/rating.html',  
  cssUrl: 'packages/ng_rating/rating.css',  
  publishAs: 'cmp'  
)  
class RatingComponent {  
  @NgTwoWay('rating')  
  int rating = 0;  
}
```

```
<rating max-rating="5" rating="item.rating"></rating>
```



Putting it all together

What's yer pirate name?

Lars

Arrr! Write
yer name!

Arrr! Me name is

LARS THE CAPTAIN





Angular and Polymer, sitting in a DOM tree,
B-i-n-d-i-n-g.
First comes elements,
Then comes components,
Then comes the interop with the node dot bind.





Old 'n busted

```
▼<div tabindex="0" style="position: relative; min-height: 100%;">
  ▶<div class="vI8oZc cS">...</div>
  ▼<div class="nH" style="width: 1440px;">
    ▼<div class="nH" style="position: relative;">
      ▶<div class="nH w-asV aiw">...</div>
      ▼<div class="nH">
        ▼<div class="no">
          ▶<div style="width: 220px; height: 662px;" class="nH oy8MbF nn aeN">...</div>
          ▼<div class="nH nn" style="width: 1220px;">
            ▼<div class="nH">
              ▼<div class="nH">
                ▼<div class="ar4 z">
                  ▶<div id=":ro" class="aeH">...</div>
                  ▼<div class="A0">
                    ▼<div id=":rp" class="Tm aeJ" style="height: 642px;">
                      ▼<div id=":rr" class="aeF" style="min-height: 216px;">
                        ▼<div class="nH">
                          ▼<div class="Blthke nH oy8MbF" style role="main">
                            <div style></div>
                            <div class="afn"></div>
                            <div class="afn"></div>
                            ▼<div class="UI" gh="tl">
                              <div class="aDP"></div>
                              ▼<div class="ae4" style>
                                ▶<div class>...</div>
                                ▼<div class="Cp">
                                  ▼<div>
                                    ▼<table cellpadding="0" id=":nl" class="F cf zt">
                                      ▶<colgroup>...</colgroup>
                                      ▼<tbody>
                                        ▶<tr class="zA zE" id=":lv5">...</tr>
                                        ▶<tr class="zA zE" id=":33s">...</tr>
                                        ▼<tr class="zA zE" id=":373">
                                          <td class="PF xY"></td>
                                          ▶<td id=":374" class="oZ-x3 xY aid" style>...</td>
                                          ▶<td class="apU xY">...</td>
                                          ▶<td class="WA xY">...</td>
                                          ▶<td class="yX xY ">...</td>
                                          ▼<td id=":379" tabindex="0" role="link" class="xY">
                                            ">
                                              ▼<div class="xS">
                                                ▼<div class="xT">
                                                  ▼<div class="y6">
```

New hotness

<messages>

<message>

<subject>

Please fill out the TPS report

</subject>

<sent>2012-10-03</sent>

<summary>

I'm going to have to ask you to come in...

</summary>

</message>

<message>

<subject>

Reminder: fill out that TPS report!

</subject>

<sent>2012-10-04</sent>

<summary>

It's been 24 hours...

</summary>

</message>

...

</messages>

#dartlang #angular



Encapsulated

`<custom-element>`

Structure

```
<div>
  <input>
  <p>
    <span></span>
  </p>
</div>
```

Behavior

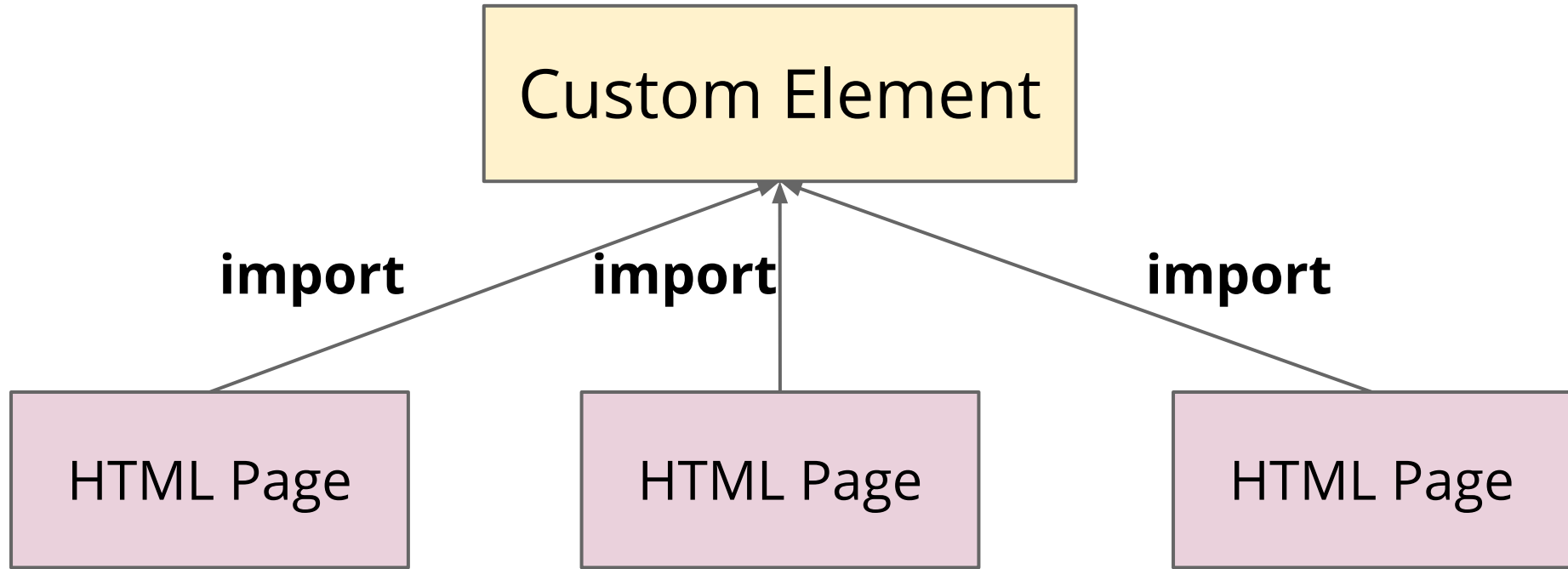
```
tag.verifyAccount();
```

Styles

```
<style>
  p { color: red; }
</style>
```

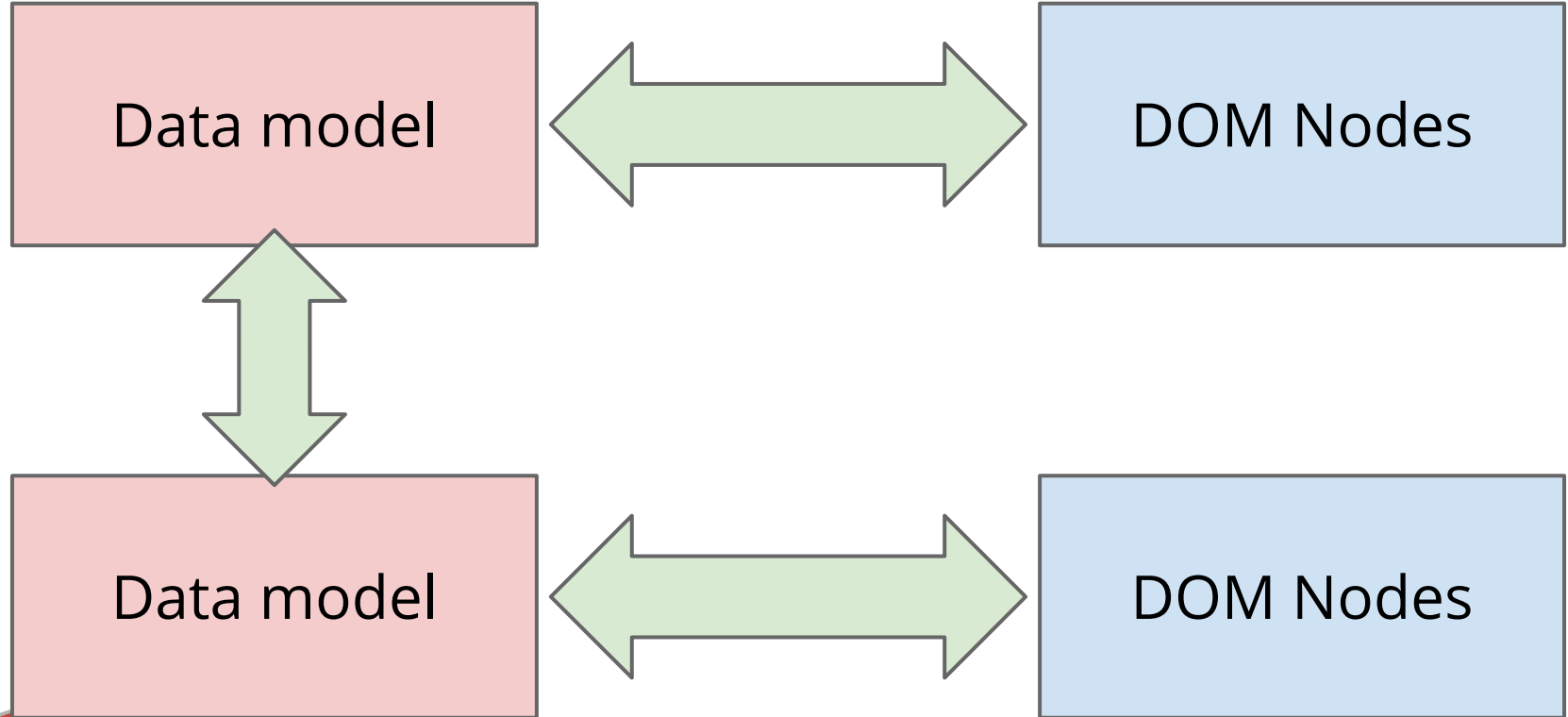


Reusable



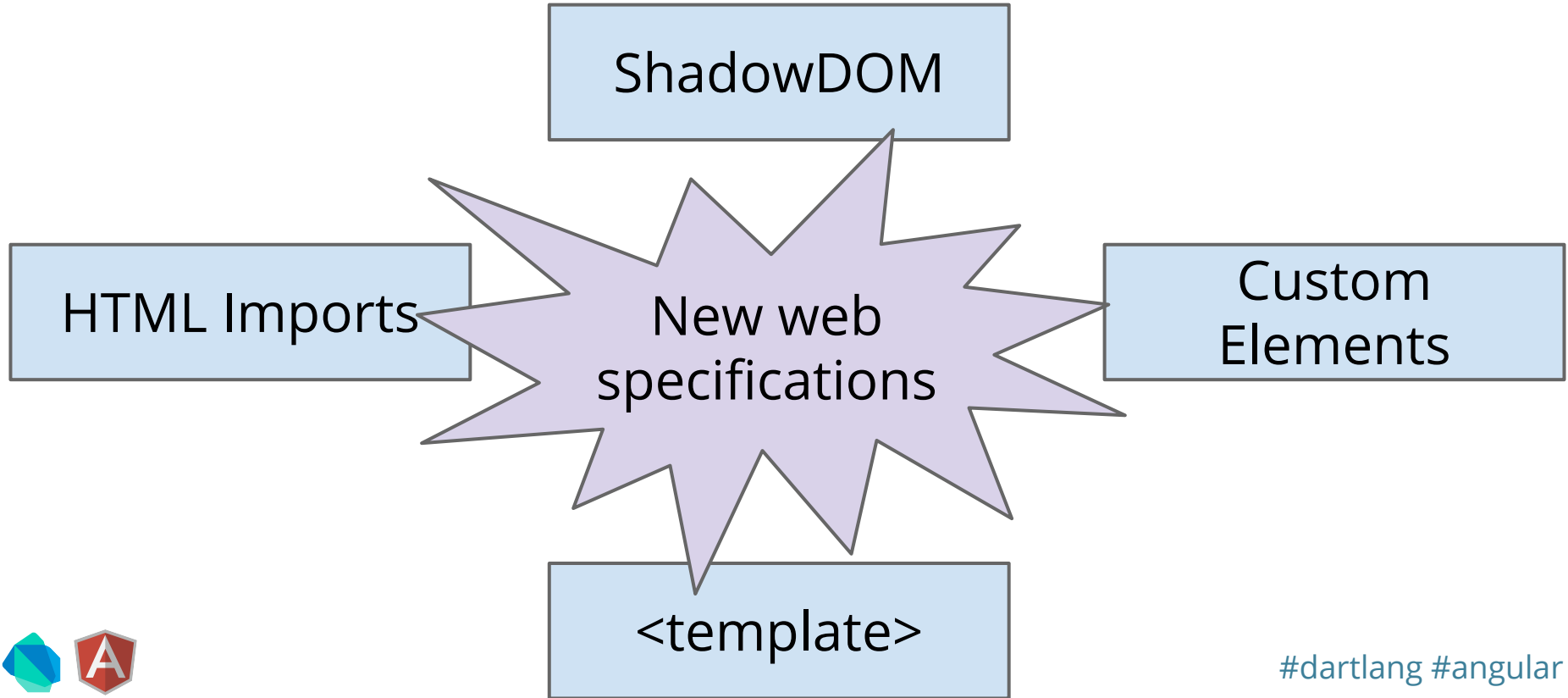


Data binding





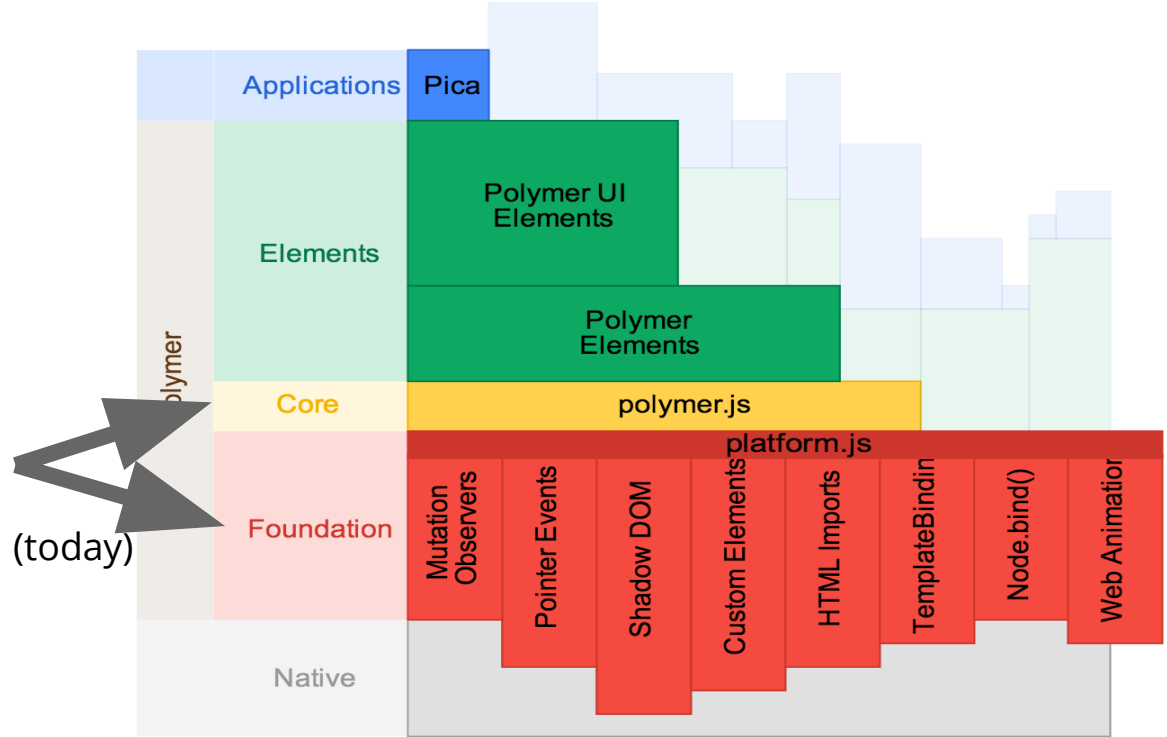
Future proof





Using web components today

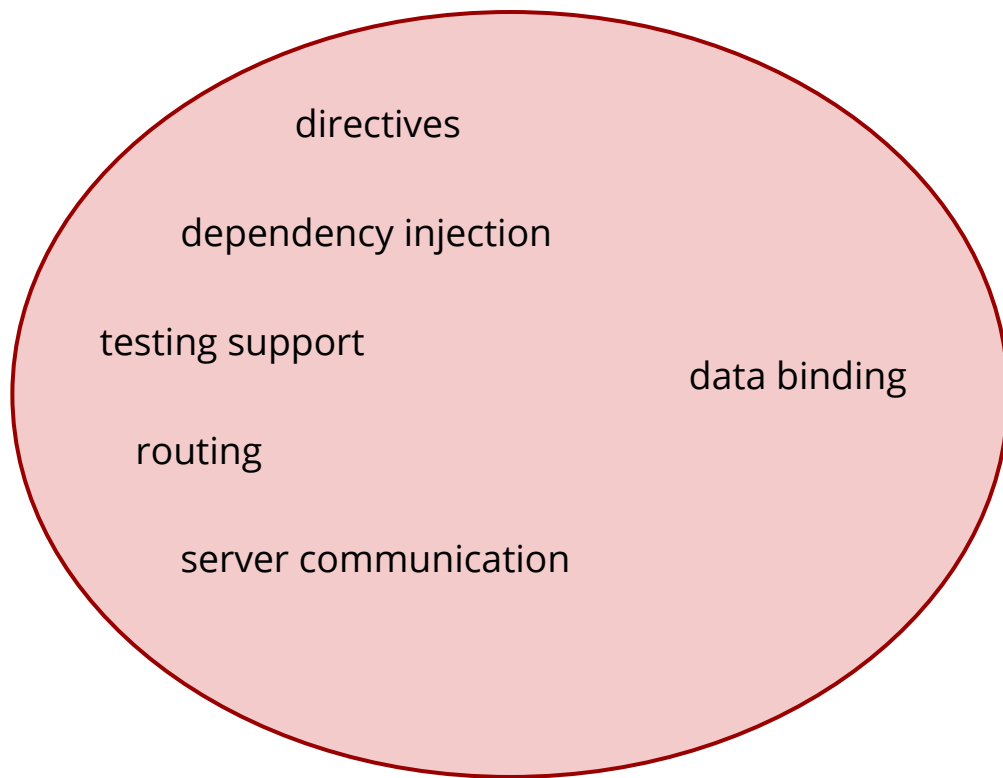
Polymer.dart



#dartlang #angular



Angular

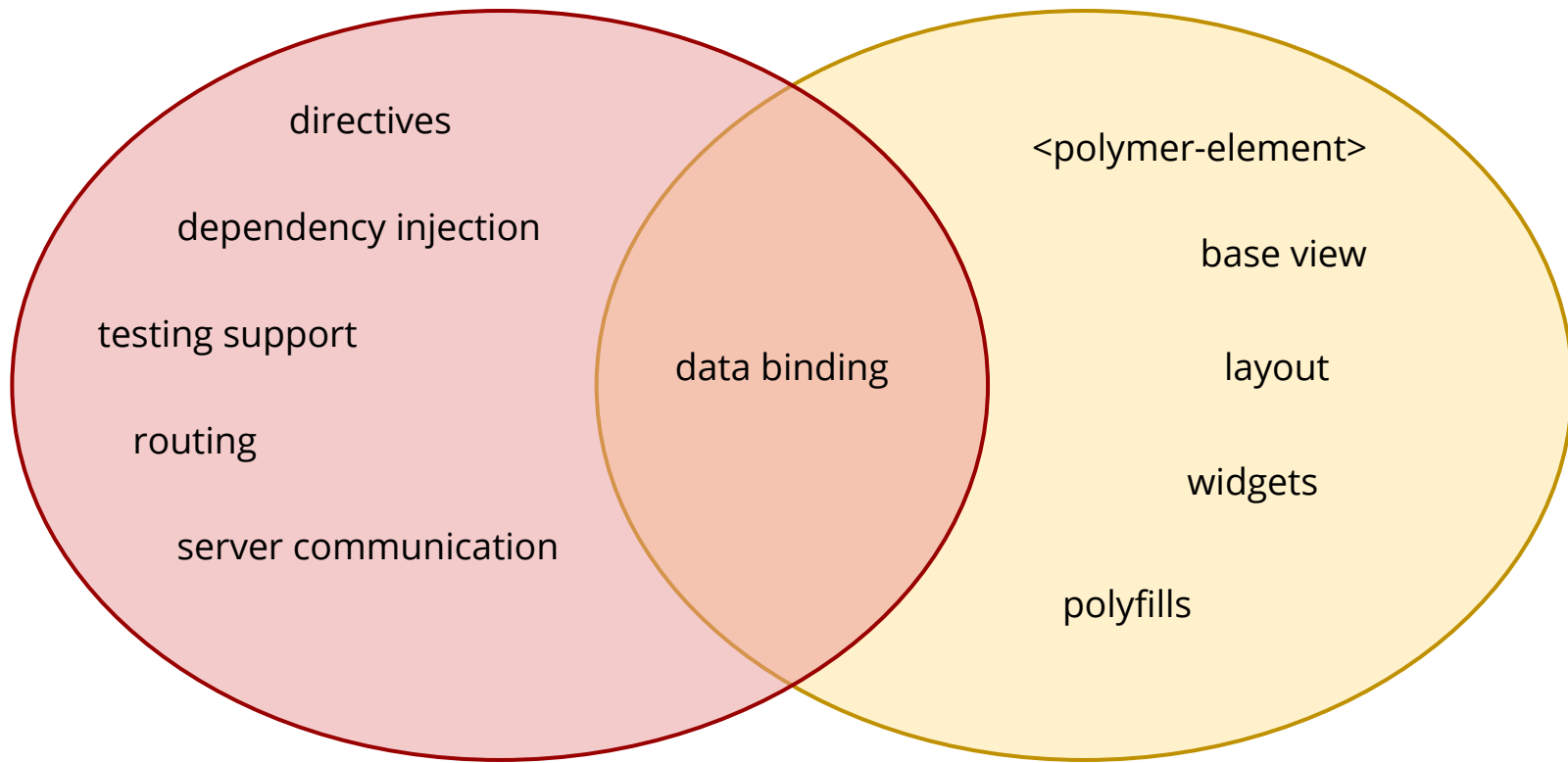


Foundation: Shadow DOM, Templates, Custom Elements, ...



Angular

Polymer



Foundation: Shadow DOM, Templates, Custom Elements, ...





#dartlang #angular

IDC expects **that tablet shipments will surpass desktop** PCs in 2013 and portable PCs in 2014

<http://www.marketwatch.com/story/worldwide-smart-connected-device-market-crossed-1-billion-shipments-in-2012-apple-pulls-near-samsung-in-fourth-quarter-according-to-idc-2013-03-26>



#dartlang #angular

48% of Facebook users on a given day are **only accessing it from mobile.**

<http://techcrunch.com/2013/10/30/nearly-half-48-of-daily-users-of-facebook-are-now-mobile-only-says-ceo-zuckerberg/>

#dartlang #angular



In the US, **93% more time** was spent online from May 2010 to May 2013. **80% of that growth was mobile** devices.

[http://www.comscore.](http://www.comscore.com/Insights/Blog/Theres_a_20_Billion_Pot_of_Gold_at_the_End_of_the_Mobile_Advertising_Rainbow)

[com/Insights/Blog/Theres_a_20_Billion_Pot_of_Gold_at_the_End_of_the_Mobile_Advertising_Rainbow](http://www.comscore.com/Insights/Blog/Theres_a_20_Billion_Pot_of_Gold_at_the_End_of_the_Mobile_Advertising_Rainbow)

[#dartlang](#) [#angular](#)



Demo time



#dartlang #angular

Resources

Tools

drone.io



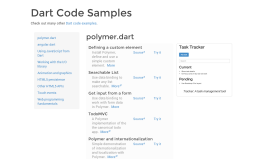
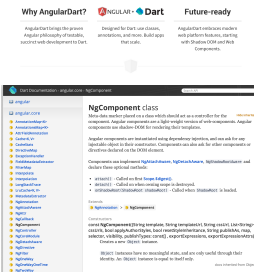
Karma

Spectacular Test Runner



docker

Docs



Libraries

MySQL

Postgres



Redis

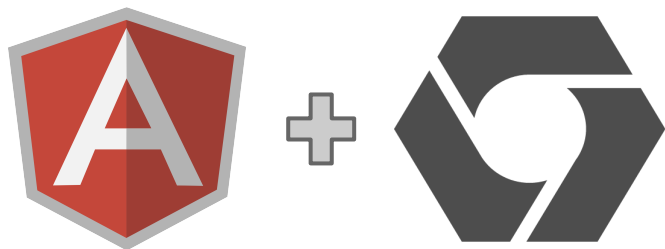
MongoDB

~700 in pub

Books



#dartlang #angular



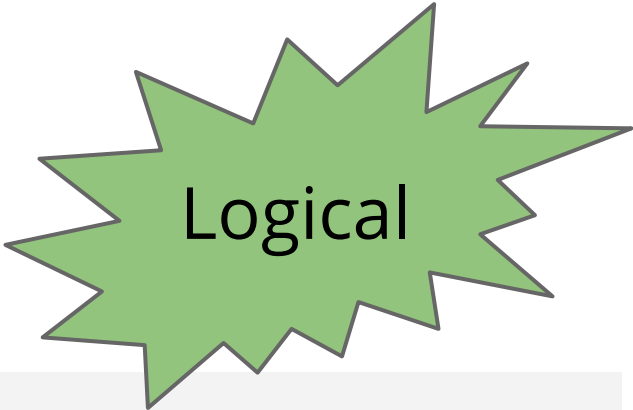
<thank-you!>

#dartlang #angular



String compared to number?

`'hello' > 1 // ??`



Logical

Class 'String' has no instance method '>'.



Variable scope?

```
var foo = 'top-level';  
  
main() {  
  if (true) { var foo = 'inside'; }  
  
  print(foo); // ?? What will this print?  
}
```



No
hoisting

Logical



top-level