

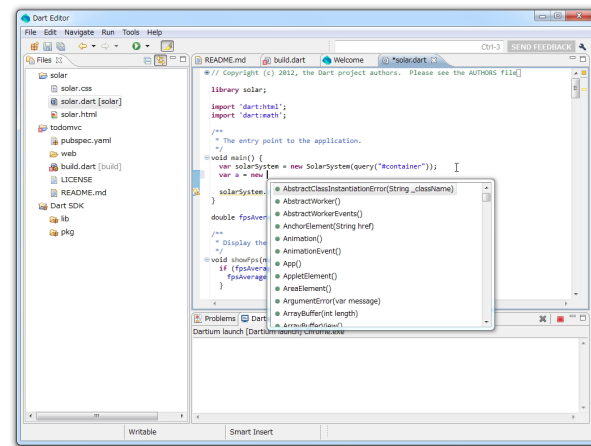
An introduction to DART

+Moritz Post, Software Engineer, EclipseSource



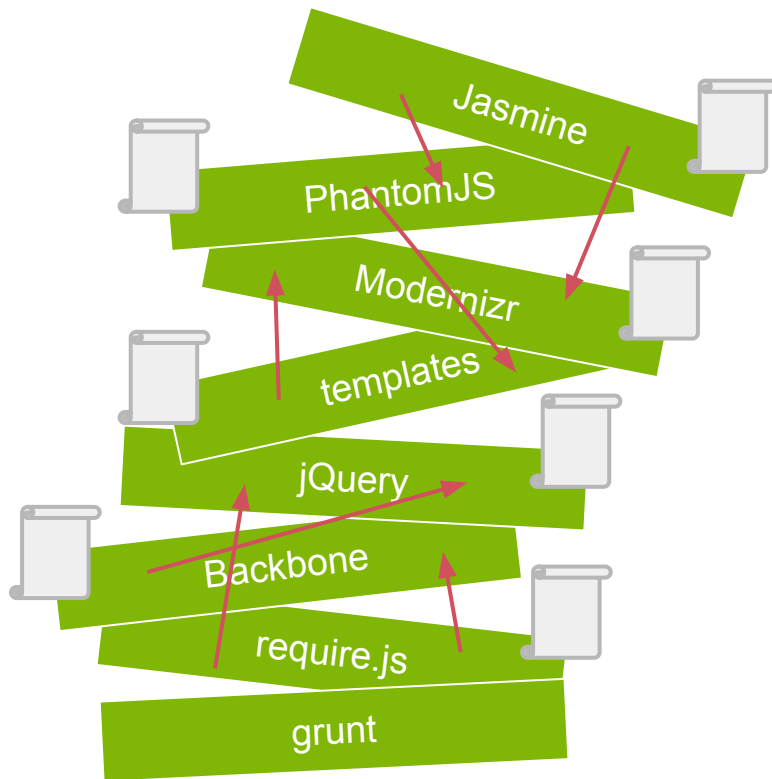
Dart - Batteries included

- Language and libraries
- Tools
- Virtual Machine
- Compiler to Javascript



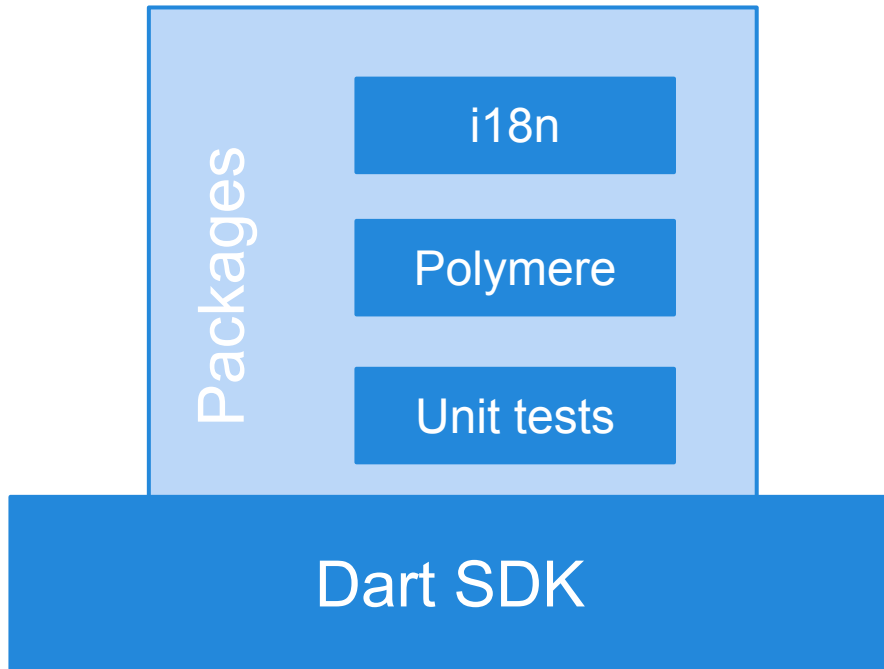
State of JavaScript

“I want to write
web apps!”



Dart is consistent

“Everybody
builds on the
same stack”



Be much better

	Structure	Syntax	Semantics	Tools	Core Libs	Requires Compilation for Development	Performance
Vanilla JS	-	-	-	-	-	No	-
Dart						No	
Closure			-			Yes	-
CoffeeScript			-	-	-	Yes	-
TypeScript			-		-	Yes	-



Syntax



Semantics



Structure

```
class Hug {  
    num strength;  
  
    Hug(this.strength) ;  
  
    Hug.bear() : strength = 100;  
  
    Hug operator +(Hug other) {  
        return new Hug(strength + other.strength) ;  
    }  
  
    void patBack({int hands: 1}) {  
        // ...  
    }  
  
    String toString() => "Hugging with $strength";  
}
```

Clean semantics and behavior

Examples:

- Only ***true*** is truthy
- No ***undefined***, only ***null***
- Clear operator behavior with `==`, `+`

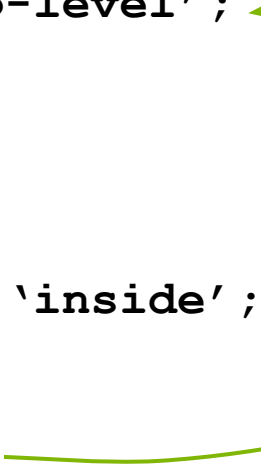
Missing getter?

`"hello".missing // ??`

➡ `NoSuchMethodError: method not found: "missing"`
`Receiver: "hello"`
`Arguments: []`

Variable scope?

```
var foo = 'top-level';  
  
void bar() {  
    if (true) {  
        var foo = 'inside';  
    }  
    print(foo);  
}
```



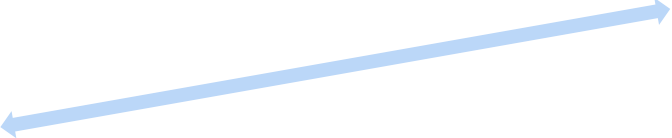
*Scope is
truly lexical*

→ top-level

```
main() { bar(); } // what does this print?
```

Scope of 'this'?

```
class AwesomeButton {  
  
    AwesomeButton(button) {  
        button.onClick.listen((Event e) => this.performMagic());  
    }  
  
    performMagic() {  
        // ..  
    }  
}
```



Truly lexical

Application structure

Libraries

Packages

Classes

Interfaces

Mixins

Functions

```
library games;
```

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

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

```
class StrategyGame extends Game {  
  //...  
}
```

```
Player findOpponent(int skill) {  
  // ..  
}
```

JS-Interop

Integrate with existing JavaScript code

```
var api = js.context.chartsApi;  
var data = js.array([1, 3, 3, 7]);  
var chart = new js.Proxy(api.BubbleChart, query('#chart'));  
chart.draw(data);
```

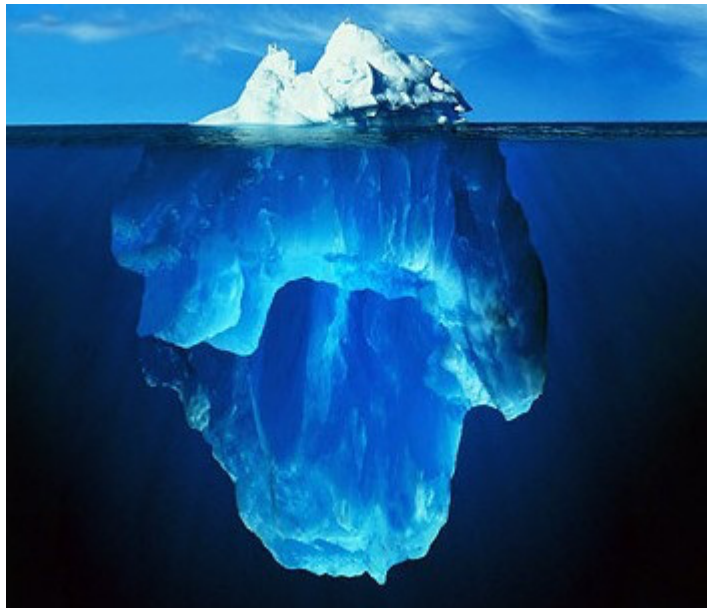


```
var api = chartsApi;  
var data [1, 3, 3, 7];  
var chart = new api.BubbleChart(querySelector('#chart'));  
chart.draw(data);
```



Much much more...

- Futures
- Streams
- Reflection
- Server-side
- Pub package manager
- Isolates
- Tooling
- Lots more...



Thank you

dartlang.org

+Moritz Post

