DART

versus

motörhead

Christian Grobmeier
http://www.grobmeier.de
@grobmeier

- Programmiersprache von Google
- Kann JS ersetzen
- Läuft in einer VM
- Für „ernsthafte“ Anwendungen
- Jung und rockt

- Heavy Metal Band aus England
- Ersetzt Hawkwind
- Läuft mit Whisky
- Zur täglichen Anwendungen
- Alt und rockt

Band

Lemmy

Bass + Gesang

Wizzo

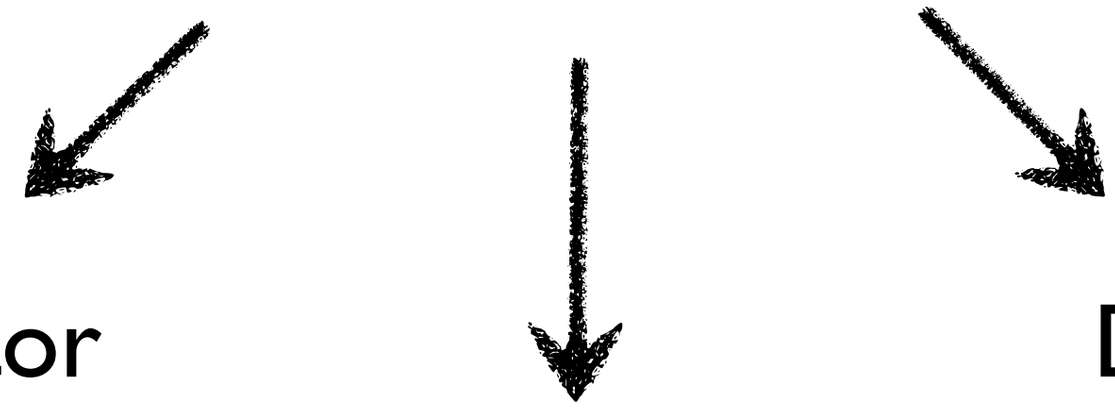
Schlagzeug

Mikkey Dee

Gitarre

Band

DART



Dart Editor

Dartium

Dart SDK

Releases

DART

Lang Spec v0.08 :-(

kaum Libraries :-(

21 Studioalben!

+ unzähliges mehr

Highlights

+ 17 mehr!

DART Highlights

Dart ist „Mainstream“, hat eine main-Methode und kennt nur ein null.

DART Highlights

Dart kennt Interfaces und Klassen

DART Highlights

```
class MyReceiver extends Isolate {
}
```

Einfachvererbung

```
interface Hobbit extends A, B{
}
```

Mehrvererbung

```
class Frodo
        implements Hobbit, Hungry {
}
```

DART Highlights

Dart kennt optionale Typen

```
dart --enable_type_checks App.dart
```

Dynamic

```
class Frodo {
  var hungry;
  bool sleepy;
  Ring ring;
}
```

DART Highlights

Dart does not need to drink whisky.

It has whisky in it‘s veins.

```
class Frodo {
   Frodo(num age) : super() {
   }
}
```

```
var frodo = new Frodo(111);
```

Konstruktor

```
class Frodo {
   Frodo.eat(){
   }
}
var frodo = new Frodo.eat();
```

Ein „named Constructor“ verhält sich wie ein überladener Konstruktor in Java.

```
class Frodo {
   var hungry;
   Frodo.cook(this.hungry);
}

new Frodo.cook(true);
```

```
class Frodo {
  bool _hungry;
  bool get hungry() => _hungry;
  void set hungry(bool x) {
    _hungry = x;
  }
}
```

Get/Set kann nicht überschrieben werden und können nicht überschreiben.

```
Frodo f = new Frodo();
f.hungry = true;
```

Setter werden wie Standardproperties aufgerufen.

GET/SET Methoden vermeiden!

DART Highlights

```
class Hobbit {
 factory Hobbit() {
   return new Hobbit._internal();
 }

 Hobbit._internal() {
   print("Construct");
  }
}

main() {
 Hobbit hobbit = new Hobbit();
}
```

Das Factory Pattern mit Dart

DART Highlights

Dart ist Library-Scoped

DART Highlights

Privacy

```
class Frodo {
  eat(){ }
  _sleep() { }
}
```

Public

Private

DART Highlights

Privacy

```
#library(„Frodo“);
class Frodo {...
```

Public

```
#library(„Mordor“);
#import(„Frodo.dart“);
new Frodo()...
```

Private Elemente sind nicht sichtbar

DART Highlights

Dart Isolates

Threads => Isolates

- Isolates erlauben „multithreading“
- Isolates kommunizieren via Ports
- Isolates werden durch „spawn“ing erzeugt

DART

Isolates

```
<body>
 <script type="application/dart">
   main() {
     // Do some DART
   }
 </script>
 <script type="application/dart">
   main() { }
 </script>
</body>
```

Isolate

Isolates

Light

- Leben im gleichen Thread
- Nur ein Isolate auf einmal

Heavy

- Erzeugen einen neuen Thread
- Mehrere Isolates auf einmal

Dart entscheidet über den Geschmack.
Isolates können keine States teilen!
Isolates sind immer asynchron!

DART

Isolates

```
process() {
   port.receive((var m, SendPort r) {
        print ("Got message: ${m}");
        r.send("close");
        port.close();
   });
}
```

DART

Isolates

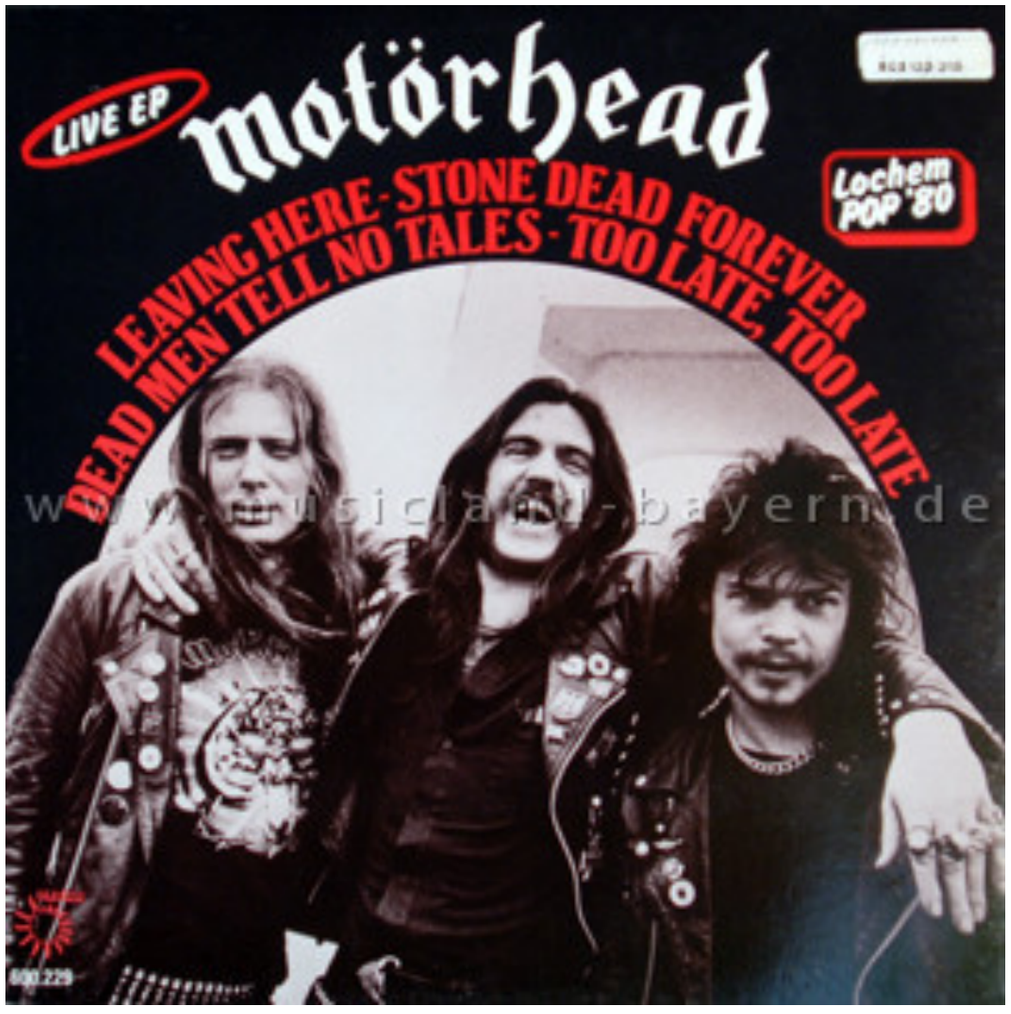
```
main() {
  port.receive((var m, SendPort r) {
    print("Got message: ${m}");
    port.close();
  });

  SendPort s = spawnFunction(process);
  s.send("start", port.toSendPort());
}
```

- Kein Multithreading - einmal Motörhead, immer Motörhead
- Mit Motörhead ist man niemals einsam
- Ist der eine Thread einmal beendet, gibt es keine neuen Alben mehr

Road Crew

DART

- HTML
- Templates
- JSON
- Dartdoc
- Frog
- etc.



- We are the Road Crew
- Unbekannte Personen

HTML Library

```
document.query('#myid');
document.query('.foo');
document.queryAll('.foo');
```

Inspiriert von jQuery

HTML Library

```
elem.attributes.contains('name');
elem.attributes['name'];
elem.attributes['name'] = 'value';
elem.attributes.remove('name');
```

Collections!

HTML Library

```
new Element.tag('div');

TableElement t = new Element.html(
'<table><tr><td>Hi</td></tr></table>'
);
```

Verbesserte Elementerstellung - mit Konstruktoren

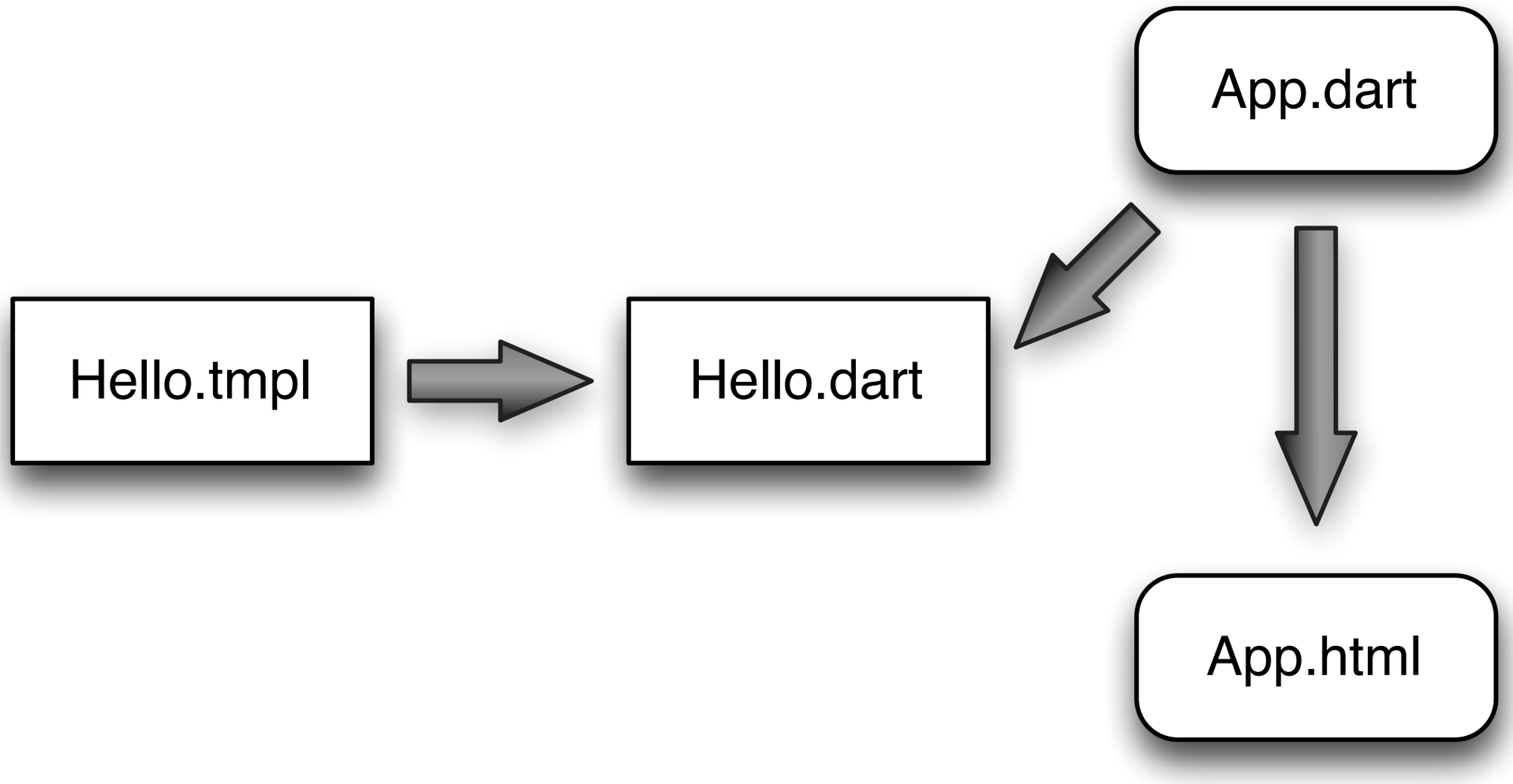
HTML Library

```
elem.on.click.add(
    (event) => print('click!'));
```

```
elem.on.click.remove(listener);
```

Alle Events sind in einem Property erreichbar.

DART Templates



DART Templates

```
template Hello(String to) {
  <div>${to}</div>
}
```

```
$ template Hello
```

DART Templates

```
main() {
   Hello h = new Hello("Motörhead");
   var panel = document.query("#p");
   panel.elements.add(h.root);
}
```

And the winner is...

DART

motörhead

Unentschieden

- ☆ Offenheit
- ☆ Stabile API
- ☆ Template System
- ☆ Releases
- ☆ Zukunft
- ☆ Browserübergreifend
- ☆ Android Support
- ☆ Geschwindigkeit
- ☆ Lautstärke

Röck ön!

Danke!

Christian Grobmeier
http://www.grobmeier.de
@grobmeier