10 Reasons You're Going to Love Dart

Chris Strom
@eee_c
- Browser Language (primarily)
- Familiar
- Incorporates Lessons Learned
- Browser Language (primarily)
- Familiar
- Incorporates Lessons Learned
A language that doesn't affect the way you think about programming, is not worth knowing.

-Alan Perlis
Ew... Static Typing
non-standard language

hostile to the open-web
Nobody is using it...
say_hi(name) {
  return "Hi, ${name}";
}

say_hi("Bob"); // "Hi, Bob"
Quick Dart Primer: Functions

```dart
say_hi(name) {
  return "Hi, \${name}";
}
say_hi("Bob"); // "Hi, Bob"
```
Quick Dart Primer: Functions

say_hi(name) {
  return "Hi, ${name}";
}

say_hi("Bob"); // "Hi, Bob"
Quick Dart Primer: Functions

```dart
say_hi(name) {
  return "Hi, \${name}";
}

say_hi("Bob"); // "Hi, Bob"
```
Quick Dart Primer: Functions

say_hi(name) {
  return "Hi, ${name}";
}

say_hi("Bob"); // "Hi, Bob"
say_hi(name) {
    return "Hi, ${name}";
}
say_hi("Bob"); // "Hi, Bob"
Quick Dart Primer: Functions

```dart
say_hi(name) {
    return "Hi, ${name}";
}

say_hi("Bob"); // "Hi, Bob"
```
Quick Dart Primer: Functions

say_hi(name) {
    return "Hi, ${name}";
}

say_hi("Bob"); // "Hi, Bob"
Quick Dart Primer: Functions

say_hi(name) {
  return "Hi, ${name}";
}

say_hi("Bob"); // "Hi, Bob"
Quick Dart Primer: Functions

say_hi(name) {
  return "Hi, ${name}";
}
say_hi("Bob"); // "Hi, Bob"
Quick Dart Primer: Functions

say_hi(name) => "Hi, ${name}"

say_hi("Bob"); // "Hi, Bob"
Quick Dart Primer: Anon. Functions

```dart
dart
var say_hi = (name) => "Hi, ${name}";
say_hi("Bob"); // "Hi, Bob"
```
Oh, Ruby… I ❤️ You So Much

class Cookie
  attr_reader :number_of_chips
  def initialize(num)
    @number_of_chips = num
  end
end

c = Cookie.new(1)
c.number_of_chips => 1
class Cookie {
  var number_of_chips;

  Cookie(num) {
    number_of_chips = num;
  }
}
class Cookie {
    var number_of_chips;

    Cookie(num) {
        number_of_chips = num;
    }
}
class Cookie {
    var number_of_chips;
    Cookie(this.number_of_chips);
}
Magic Instance Variables

class Cookie {
  var number_of_chips;
  Cookie(this.number_of_chips);
}

> var cookie = new Cookie(12);
> cookie.number_of_chips; // 12
> cookie.number_of_chips = 25;
> cookie.number_of_chips; // 25
Magic Instance Variables

class Cookie {
    var _calories;
    // ...
}

> var cookie = new Cookie();
> cookie._calories;  // Error!
> cookie._calories = 25;  // Error!
Getters and Setters

class Cookie {
  var _calories;
  get calories => _calories / 3;
  set calories(v) { _calories = v; } 
}

> var cookie = new Cookie();
> cookie.calories = 300;
> cookie.calories; // 100
Optional Parameters

class Cookie {
    var calories;
    Cookie({calories}) {
        this.calories = calories;
    }
}

> c1 = new Cookie(calories: 300);
> c2 = new Cookie();
Optional Parameters

class Cookie {
  var calories, number_of_chips;
  Cookie(
    this.calories,
    this.number_of_chips
  );
}

> c1 = new Cookie(calories: 300);
> c1.calories; // 300
Optional Parameters

class Cookie {
    var calories, number_of_chips;
    Cookie({
        this.calories,
        this.number_of_chips
    });
}

> c2 = new Cookie(number_of_chips: 9);
> c2.number_of_chips; // 9
Optional Parameters

class Cookie {
    var calories, number_of_chips;
    Cookie({
        this.calories,
        this.number_of_chips: 2
    });
}

> c3 = new Cookie();
> c3.number_of_chips; // 2
Optional Parameters

class Cookie {
    var calories, number_of_chips;
    Cookie([this.calories, this.number_of_chips]);
}

> c1 = new Cookie(120, 9);
> c2 = new Cookie(120);
> c3 = new Cookie();
Method Cascades

cookie
  ..mix_it()
  ..stir_it()
  ..bake_it()
  ..eat_it();
Method Cascades

cookie
  ..calories = 120
  ..number_of_chips = 12
  ..bake_time = 30
  ..deliciousness = 42;
Method Cascades

bg.style
  ..position = 'absolute'
  ..top = '0px'
  ..left = '0px'
  ..width = "${doc.offsetWidth}px"
  ..height = "${doc.clientHeight}px"
  ..backgroundColor = 'black'
  ..opacity = '0.8'
  ..zIndex = '1000';
Unit Tests

test_create() {
    group("new DBs", () {
        setUp(removeFixtures);
        tearDown(removeFixtures);
        test("creates a new DB", () { /*...*/ });
    });
}
Unit Tests

test_create() {
    group("new DBs", () {
        setUp(removeFixtures);
        tearDown(removeFixtures);
        test("creates a new DB", () { /*...*/ });
    });
}
Unit Tests

// ...

test("creates a new DB", () {
    var db = new Dirty('test/test.db');
    expect(new File('test/test.db').exists(), equals(true));
});
Unit Tests

test("can write to the DB", () {
    var db = new Dirty('test/test.db');
    db['everything'] = {'answer': 42};
    db.close(expectAsync0(() {
        expect(
            new File('test/test.db').size(),
            greaterThan(0)
        );
    }));
});
test("can write to the DB", () {
    var db = new Dirty('test/test.db');
    db['everything'] = {'answer': 42};
    db.close(expectAsync0(() {
        expect(
            new File('test/test.db').size(),
            greaterThan(0)
        );
    }));
});
import 'dart:html';
import 'dart:json';
import 'cookie.dart';

main() {
  var cookie = new Cookie(12);
}

Client-side Libraries
Client-side Libraries

class Cookie {
    var number_of_chips;

    Cookie(this.number_of_chips);
}

library cookies;

class Cookie {
    var number_of_chips;

    Cookie(this.number_of_chips);
}
Client-side Libraries

```dart
import 'dart:html';
import 'dart:json';
import 'cookie.dart';

main() {
  var cookie = new Cookie(12);
}
```
Pub Packages

$ cd public/scripts
$ cat pubspec.yaml
name: killer_app
dependencies:
  hipster_mvc: any
$ pub install
Resolving dependencies...
Dependencies installed!
Pub Packages

$ cd public/scripts
$ cat pubspec.yaml
name: killer_app
dependencies:
  hipster_mvc: any
$ pub install
Resolving dependencies...
Dependencies installed!
Pub Packages

$ cd public/scripts
$ cat pubspec.yaml
name: killer_app
dependencies:
  hipster_mvc: any
$ pub install
Resolving dependencies...
Dependencies installed!
library cookie_model;
import 'package:hipster_mvc/hipster_model.dart';

class Cookie extends HipsterModel {
  Cookie(attributes) : super(attributes);
}

Pub Packages
Pub Packages

- pub.dartlang.org
- public git repositories
- great for local dev
Statically Typed Language
Statically Typed Language

(but not really)
Statically Typed Language

Sneakily
(Not Really) Static Typing

class Cookie {
    var number_of_chips;
    Cookie(this.number_of_chips);
}

(Not Really) Static Typing

class Cookie {
    int number_of_chips;
    Cookie(this.number_of_chips);
}

class Cookie {
    int number_of_chips;
    Cookie(this.number_of_chips);
}
var c1 = new Cookie(12);
var c2 = new Cookie('salt');
print("""
c1: ${c1.number_of_chips} chips
c2: ${c2.number_of_chips} chips"""");
(Not Really) Static Typing

class Cookie {
    int number_of_chips;
    Cookie(this.number_of_chips);
}

c1: 12 chips
c2: salt chips
(Not Really) Static Typing

```dart
$ dartanalyzer static_typing.dart
'String' is not assignable to 'int'
  2: var c1 = new Cookie(12);
  3: var c2 = new Cookie('salt');
```

~~~~~~~~~
Dartdocs

$ dartdoc cookie.dart
Cookie class

Constructors

new Cookie(int number_of_chips)

Properties

int number_of_chips
/// A yummy, yummy food.

class Cookie {
  /// The number of chips in the
  /// cookie. The more the better.
  int number_of_chips;

  /// We do not support cookies
  /// without chips at this time.
  Cookie(this.number_of_chips);
}

**Cookie class**

A yummy, yummy food.

**Constructors**

```dart
g中新 Cookie(int number_of_chips)

We do not support cookies without chips at this time.
```

**Properties**

```dart
int number_of_chips

The number of chips in the cookie. The more the better.
```
**Dirty class**

The `Dirty` class is a quick and dirty way to create a persistent `HashMap`. In addition to doing all of the usual `HashMap` things, it will store records in an append-only file database.

**Implements**

- `HashSet<K, V>`

**Constructors**

`new Dirty(path, {onLoad})`

The database at `path` will be created if it does not already exist. If the optional `onLoad` parameter is supplied, it will be invoked with a copy of the `Dirty` instance.

**Properties**

- `final bool isEmpty`
  
  Returns true if there is no `{key, value}` pair in the map.
**Dirty** class

The **Dirty** class is a quick and dirty way to create a persistent `HashMap`. In addition to doing all of the usual `HashMap` things, it will store records in an append-only file database.

**Implements**

`HashMap<K, V>`

**Constructors**

```dart
new Dirty(path, {onLoad})
```

The database at `path` will be created if it does not already exist. If the optional `onLoad` parameter is supplied, it will be invoked with a copy of the `Dirty` instance.

**Properties**

```dart
final bool isEmpty
```

Returns true if there is no `{key, value}` pair in the map.
Dirty class

The Dirty class is a quick and dirty way to create a persistent HashMap. In addition to doing all of the usual HashMap things, it will store records in an append-only file database.

**Implements**

HashMap<K, V>

**Constructors**

`new Dirty(path, {onLoad})`

The database at path will be created if it does not already exist. If the optional onLoad parameter is supplied, it will be invoked with a copy of the Dirty instance.

**Properties**

`final bool isEmpty`

Returns true if there is no {key, value} pair in the map.

docs inherited from Map<K, V>
Dartdocs

Class navigation

HipsterModel class

HipsterModel encapsulates individual records in your backend datastore. At its most concise, a model need only to implement the urlRoot method:

```dart
class ComicBook extends HipsterModel {
  get urlRoot => '/comics';
}
```

Implements
- Hashable

Static Methods
- String hash()

Constructors
- new HipsterModel([Map attributes])
  If attributes is not supplied, it will be initialized to an empty HashMap.
HipsterModel class
HipsterModel encapsulates individual records in your backend datastore. At its most concise, a model need only to implement the urlRoot method:

```dart
class ComicBook extends HipsterModel {
  get urlRoot => '/comics';
}
```

Implements
- Hashable

Static Methods
- String hash()

Constructors
- new HipsterModel([Map attributes])
  - If attributes is not supplied, it will be initialized to an empty HashMap.
Dartdocs

I ♥

Documentation
I ♥ Static Typing
Thanks!

Chris Strom
@eee_c
#pairwithme!!!

Buy the book :) 

Also: SPDY, Backbone.js, 3D Game Programming for Kids
Generative Constructors
Operators Overloading
Isolates
Futures