THE DART SID
OF DISRUPTIO

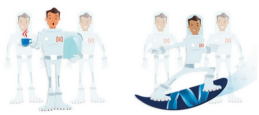
+NicolasFrancois / @nicofrancois

+GuillaumeGirou / @GirouGuillaume

GDG DevFest
2015 Season

[sf≡ir]

[sfeir]



Team Leader
#BACK
(JAVA / .NET / SCALA...)

Team Leader
#FRONT

Team Leader
#MÉTIER

Team Leader
#OPS

SFEIR ET GOOGLE

Utilisation de GWT depuis **2006**

Mise en oeuvre d'AppEngine sur projets d'envergures depuis **2010**

Utilisation d'AngularJS et veille sur Dart depuis **2012**

Utilisation de Compute Engine et BigQuery depuis **2013**

2015 : Utilisation de Polymer et Dataflow

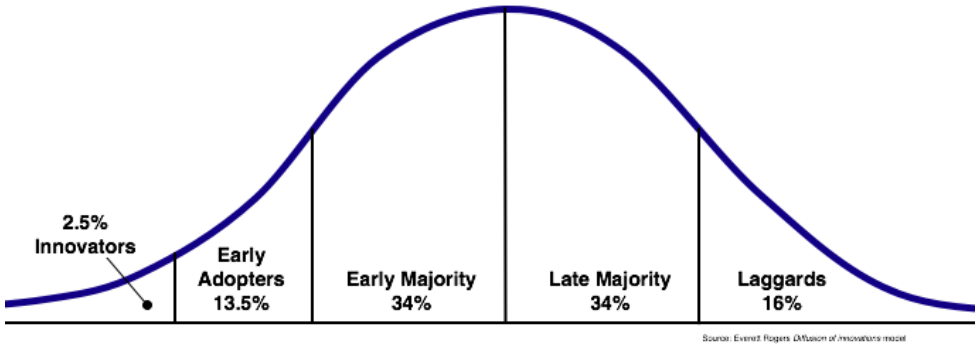
AVANT DE COMMENCER

POURQUOI ?

- Nouveau langage pour le web
- Répond aux problématiques d'aujourd'hui
- Performance
- Moderne

DISRUPTIF ?

Une technologie de rupture, également connue comme « rupture technologique », est une innovation technologique qui porte sur un produit ou un service et qui finit par remplacer une technologie dominante sur un marché.
(Wikipedia)



OBJECTIF DE L'ATELIER

Etre capable de développer une application web moderne

INTRODUCTIO
AU LANGAGE

HELLO WORL

```
void main(){
    var hello = "Hello DevFest Paris";
    print(hello);
}
```

```
Hello DevFest Paris
```

FONCTION TO LEVEL

```
void main(){
    var who = "Hello DevFest Paris";
    print(sayHello(who));
}

String sayHello(String name) =>  "Hello $name"
```

```
Hello DevFest Paris
```

MA 1ÈRE CLAS

```
class MaClasse {

}
```

```
var classe = new MaClasse();
MaClasse unAutreClasse = new MaClasse();
```

CLASSE COMPLÈTE

```dart
import 'dart:math';

class Point {
  final int x, y;
  Point(this.x, this.y);
  num distanceTo(Point other) {
    var dx = x - other.x;
    var dy = y - other.y;
    return sqrt(dx * dx + dy * dy);
  }
  String toString() => "($x, $y)";
}
```

```dart
var p = new Point(2, 3);
print(p);// Affiche '(2, 3)'
var x = p.x;
var distance = p.distanceTo(new Point(5, 5));
```

HÉRITAGE

```
class Person {
  Person.fromJson(Map data) {
    print('in Person');
  }
}
```

```
class Employee extends Person {
  // Person n'a pas de constructeur par défaut
  // Il faut appeler super.fromJson(data).
  Employee.fromJson(Map data) : x.fromJson(data) {
    print('in Employee');
  }
}
```

LIST

```
// Creation de listes
var vegetables = [] // == new List();
var fruits = ['apples', 'oranges'];
```

```
// Ajout d'éléments
fruits.add('kiwis');
fruits.addAll(['grapes', 'bananas']);
assert(fruits.length == 5);
```

```
// Tri
fruits.sort();
assert(fruits[0] == 'apples');
```

MAP

```
var gifts = {
  // Keys        Values
  'first'  : 'partridge',
  'second' : 'turtledoves',
  'fifth'  : 'golden rings'
};
```

```
gifts['fourth'] = 'calling birds';    // Add a key-value pair
```

```
assert(gifts['first'] == 'partridge')
assert(gifts.length == 4);
```

```
var keys = gifts.keys;
var values = gifts.values;
```

LAMDBA

```
var teas = ['green', 'black', 'chamomile', 'earl grey'];
```

```
// Affichage en majuscule
var loudTeas = teas.map((tea) => tea.toUpperCase());
loudTeas.forEach(print);
```

```
// La camomille n'est pas caéfinée.
bool isDecaffeinated(String teaName) => teaName == 'chamomile';
```

```
// Filtrage
var decaffeinatedTeas = teas.where(isDecaffeinated);
```

```
// Construction map
var decaffeinatedTeasMap =
   new Map.fromIterable(teas, value: isDecaffeinated);
```

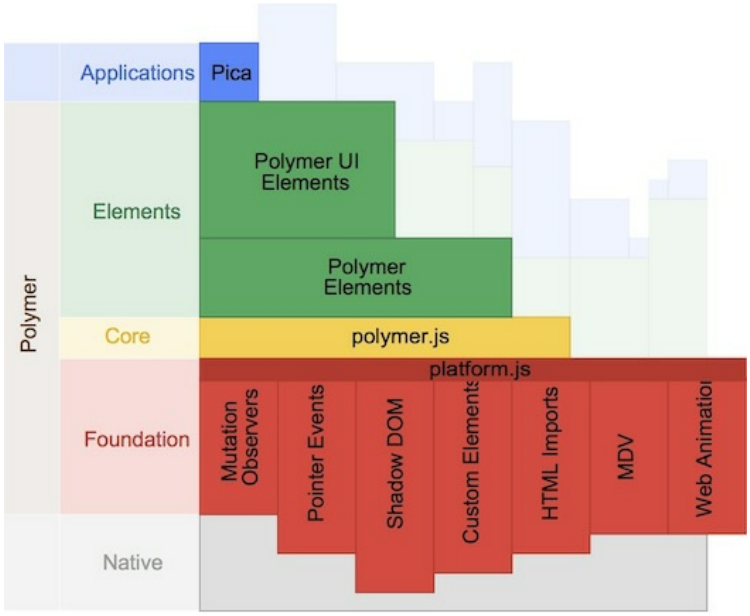
FUTURE

```
Future result = costlyQuery();

// Enchainer méthodes
return result.then((value) => expensiveWork())
             .then((value) => lengthyComputation())
             .then((value) => print('done!'))
             // Gestion erreur
             .catchError((exception) => print('DOH!'))
```

POLYMER

VUE D'ENSEMB


Applications
Pica
Polymer
Elements
Polymer UI Elements
Polymer Elements
Core
polymer.js
platform.js
Foundation
Mutation Observers
Pointer Events
Shadow DOM
Custom Elements
HTML Imports
MDV
Native

CUSTOM ELEMENTS

```
<click-counter></click-counter>
```

```
<link rel="import" href="packages/polymer/polymer.html">

<element name="click-counter">
  <template>
    <style>
        /* Css */
    </style>
    <-- Html -->
  </template>
  <-- Code Dart -->
  <script src="click_counter.dart"></script>
</element>
```

HTML IMPORT

```
<link rel="import" href="/components/click_counter.html">
```

HTML TEMPLA

```
<template id="monTemplate">
  <div class="mon-style" >
  </div>
  <!-- Code html -->
</template>
```

```
<template if="{{maCondition}}">
```

```
<template repeat="{{maList}}">
```

OBSERVER

```
<div>
  {{name}}
</div>
```

```
@observable String name;
```

SHADOW DOM

```
<video controls src="/ma/video"></video>
```

```
<video controls="" height="300" src="/ma/video">
  #shadow-root
  <div>
    <div>
      <div>
        <input type="button">
        <input type="range" precision="float" max="596.48">
        <div style="display: none;">0:00</div><div>9:56</div>
          <input type="button">
          <input type="range" precision="float" max="1" style="">
          <input type="button" style="display: none;">
          <input type="button" style="">
        </div>
    </div>
  </div>
</video>
```

EXPLICATION L'ATELIER

VOTRE MISSIO

goo.gl/l7Oxy1


[sfeir]
[Nous recrutons]