Component-based Web-Apps with AngularJS

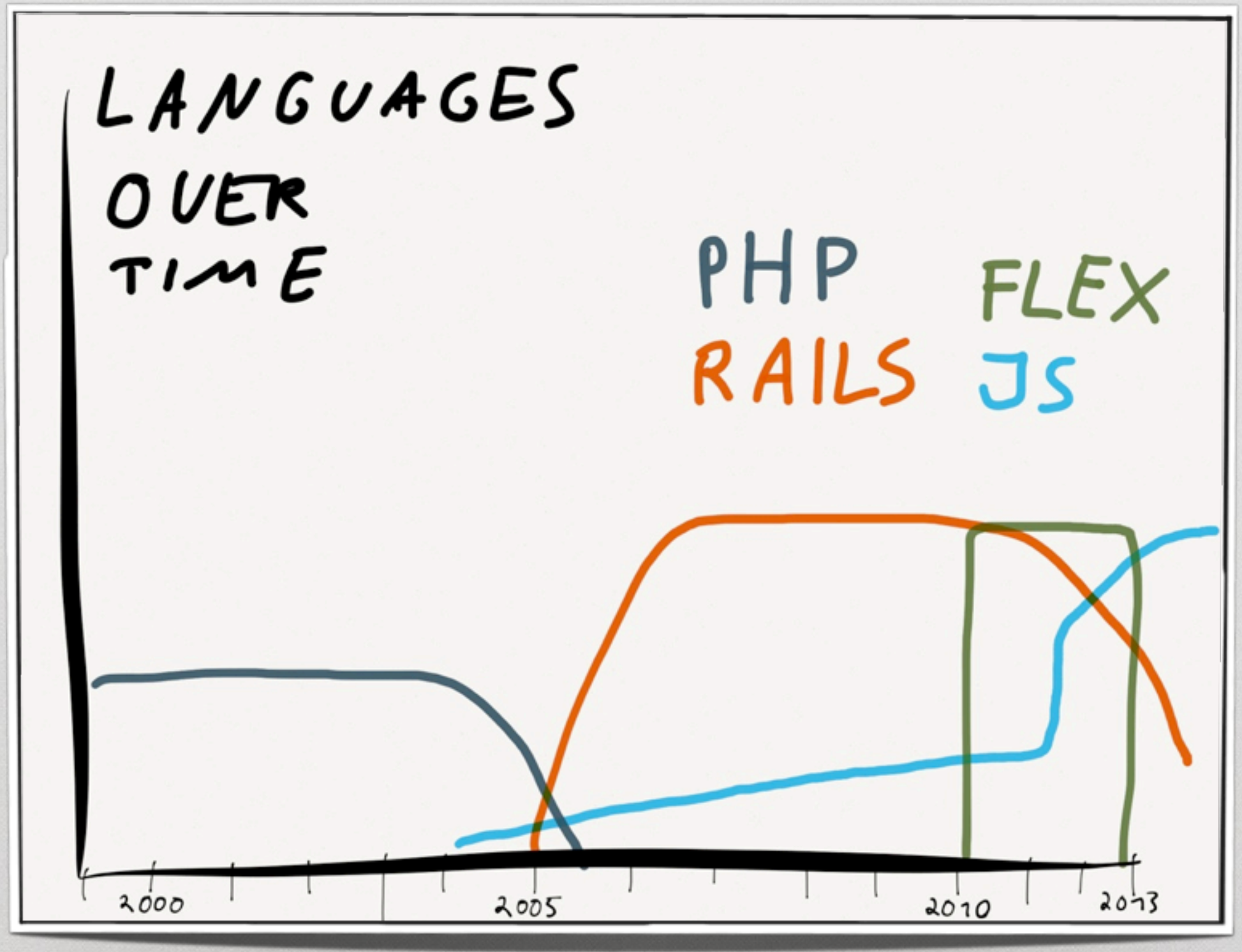
Jan Varwig

jan.varwig.org

@agento

My Background

- 1999 PHP & MySQL
- 2007 Rails
- 2010 Flex/Air
- Oops, I accidentally Javascript


LANGUAGES
OVER
TIME
PHP
RAILS
FLEX
JS
2000
2005
2010
2013

My Background

- Frontend guy
- Since November 2012
- Massive CMS Frontend
- Built entirely in AngularJS

Untitled

+ Add

-_-

Untitled

Languages

Info

Default: English

English

Klingon

dsaljkdsalkjdsaljk

Text (localized, required)

foobar

Number between 42 and 99

42

Tags (at least 3)

foo, bar, baz

Bacon Time (required)

2013-05-24

11:02:50

Place of Baconing (required)

Search Address

51.11041991029264

Latitude

10.8984375

Longitude

Entry Information

Content Type	Super Duper Validated Type
ID	hh355xls0
Current version	101
Created	2 days ago
Created by	Jan Varwig
Published	2 days ago
Published by	Jan Varwig
Published version	100

✔ All changes saved successfully

Unpublish

- Component-based App development framework
- Works directly in the DOM
- Strongly encourages good coding practices
  - Testing
  - Separation of concerns

Topics

- A little more in-depth than last time
- Develop a small App to
  - give an Impression how to actually work with AngularJS
  - demonstrate awesomeness of Component-based development

Todo-List

- … again
- Understood problem that fits into 15 minutes
- Spiced up:
  - Due dates
  - Nested Components
  - Persistence

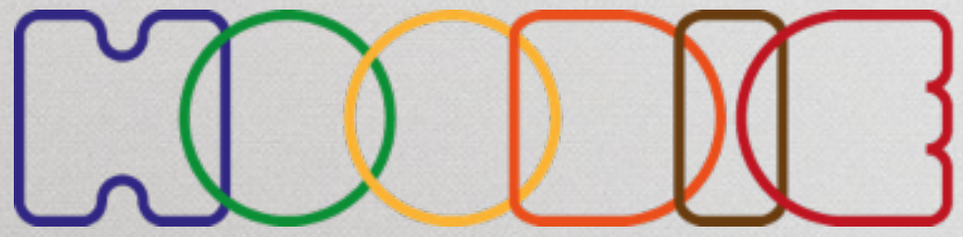
Worlds smallest Todo-List

```
<code>
  function Foo($scope){
    $scope.tasks = [];
    $scope.addTask = function() {
      $scope.tasks.push($scope.newTask);
      $scope.newTask = null;
    };
    $scope.removeTask = function (index) {
      $scope.tasks.splice(index, 1);
    };
  }
</code>
```

```
<form name="form" ng-controller="Foo">
  <h2>{{tasks.length}} Tasks</h2>
  <p>
    <input type="text" ng-model="newTask"/>
    <button ng-click="addTask()">Add</button>
  </p>
  <ul>
    <li ng-repeat="(index, task) in tasks">
      {{task}} <button ng-click="removeTask(index)">X</button>
    </li>
  </ul>
</form>
```

A little bigger

- Fancy form
- Fancy table
- Persistence using Hoodie



Optional: Setup

```
$ npm install -g yo grunt-cli
  bower generator-angular

$ yo angular:app

$ grunt server
```

Optional: Setup

- Just make sure to load angular.js
- Organize and write your Javascript/Coffeescript however you see fit

Create Classes

- No special code
- Stored in Angular module
  - Concerned with code organization
  - Not for loading code!

Create TodoList Component

- Wraps and isolates everything we do
- Use by inserting <x-todo-list/> in your code
- Directive defines component
  - HTML template
  - Code

Controller for TodoList

- Sets up the Scope
- Scope provides to the view:
  - Models
  - Methods
  - Any other state
- Also:
  - Change propagation in $digest cycle
  - Event delivery through $emit, $broadcast, and $on

```
$scope.$watch(‘expression’, function(value, old) {
  console.log(‘expression is now’, value);
});
```

```
$scope.$broadcast(‘whoop’);

$scope.$on(‘whoop’, function(event) {
  console.log(‘received whoop’, event);
});
```

Display table

- Access any methods/properties on the scope/models
- DOM-manipulating directives
  - ngRepeat
  - ngShow

Add interactivity

- Solve Tasks, Clear List
- Attach Handlers easily
  - ngClick
  - ngMouseenter
- Call any code on the scope

Add Items through form

- Strong form support through directives
  - ngModel, ngForm
  - Validations
- Working together through Scope (ngDisabled)

Persistence

- Angular provides
  - $http (AJAX)
  - $resource (Even easier for REST backends)
- Very easy to integrate your own stuff
  - Store Tasklist in Hoodie
  - Use events to trigger saving

Done

Skipped

- Watchers
- Isolated Scopes and Transclusion to create Widgets
- Compile/Link in directives (how to manipulate the DOM)
- Module System/Dependency Injection
- Testing
- Filters, Forms(custom parsers/filters/validations)

No Views or Pages
Only Components

- Web Apps vs. Web Pages
- Angular has support for Routes and Views
  - $routeProvider & ngView, UI-router
- Make intelligent, autonomous components
- Connect them together
- Let the Scope drive interactivity

Current implementations

- Desktop: XAML, MXML, SwiXML, XUL, XCode
- AngularJS
- rAppid.js
- X-Tag

Curious?

- Join the AngularJS Berlin Meetup
  - June 12th
  - @angular_berlin
  - http://lanyrd.com/series/angularjs-berlin/
- bbuzzinga ...storm hackathon
  - http://www.meetup.com/Retresco-events/events/120661962/

KTHXBYE

jan.varwig.org

@agento