

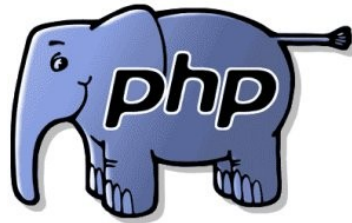
Functional Reactive Programming

In JS

Daniel Howlett

Gibraltar JS . Wednesday 13th November 2013

A few things about me



Behat



watchwithme.co.uk

dancras.co.uk

@dancras

All images used without permission but with sincere apologies

This is (not) a revolution

Composability

True abstractions

New tooling
opportunities

Concise code



The future is blindingly bright



More complex apps



Functional Reactive Programming

Intimidating syntax

```
main :: IO ()
main = withCredential $ do
  liftIO . putStrLn $ "# your home timeline (up to 100 tweets):"
  homeTimeline []
  C.$= CL.isolate 100
  C.$$ CL.mapM_ $ \status -> liftIO $ do
    let sn = userScreenName . statusUser $ status
        tweet = statusText status
    T.putStrLn $ T.concat [ sn, ": ", tweet]
```

Intimidating terminology

Continuation

Currying

Monad

Applicative

Combinator

Functor

Tail call

Flat Map

Immutable

“My JS is already functional”

Lo-Dash

UNDERSCORE.JS

The logo for Underscore.js, featuring the text "UNDERSCORE.JS" in a dark blue, sans-serif font. Below the text is a horizontal bar composed of two segments: a blue segment on the left and a dark blue segment on the right.

Imperative and declarative



```
var productList = [];  
var featuredProducts = [];  
  
_.each(allProducts, function (product) {  
  if (product.category !== currentCategory) {  
    return;  
  }  
  
  if (product.isFeatured) {  
    featuredProducts.push(product);  
    return;  
  }  
  
  productList.push(product);  
});
```

Imperative and declarative continued


```
var lists = _(allProducts)
  .filter(function (product) {
    return product.category === currentCategory;
  })
  .splitBy(function (product) {
    return product.isFeatured;
  })
  .value();
```



A real world example

Choose your customisations


How would you like to **Control** your blind?  

Left Hand Control ▼

Shape: **Wave** change 

Choose your **Braid**  

☐ None ☒ Braid +£5.00

Braid Style: **Thin Braid** 

Choose your Control Colour ←

Choose your **Type of Pull**

A functional dsl

```
$form.options( "shape", ["Wave", "Arched"] )  
  .depend( "width", $.matchers.lte(200) );  
  
$form.options( "trimming", ["Eyelet"] )  
  .depend( "shape", $.matchers.values("Straight") );  
  
$form.control( "braid" )  
  .depends("trimming", $.matchers.values("Braid") );
```

A lonely bug



Some compile-to-js FP languages

ClojureScript

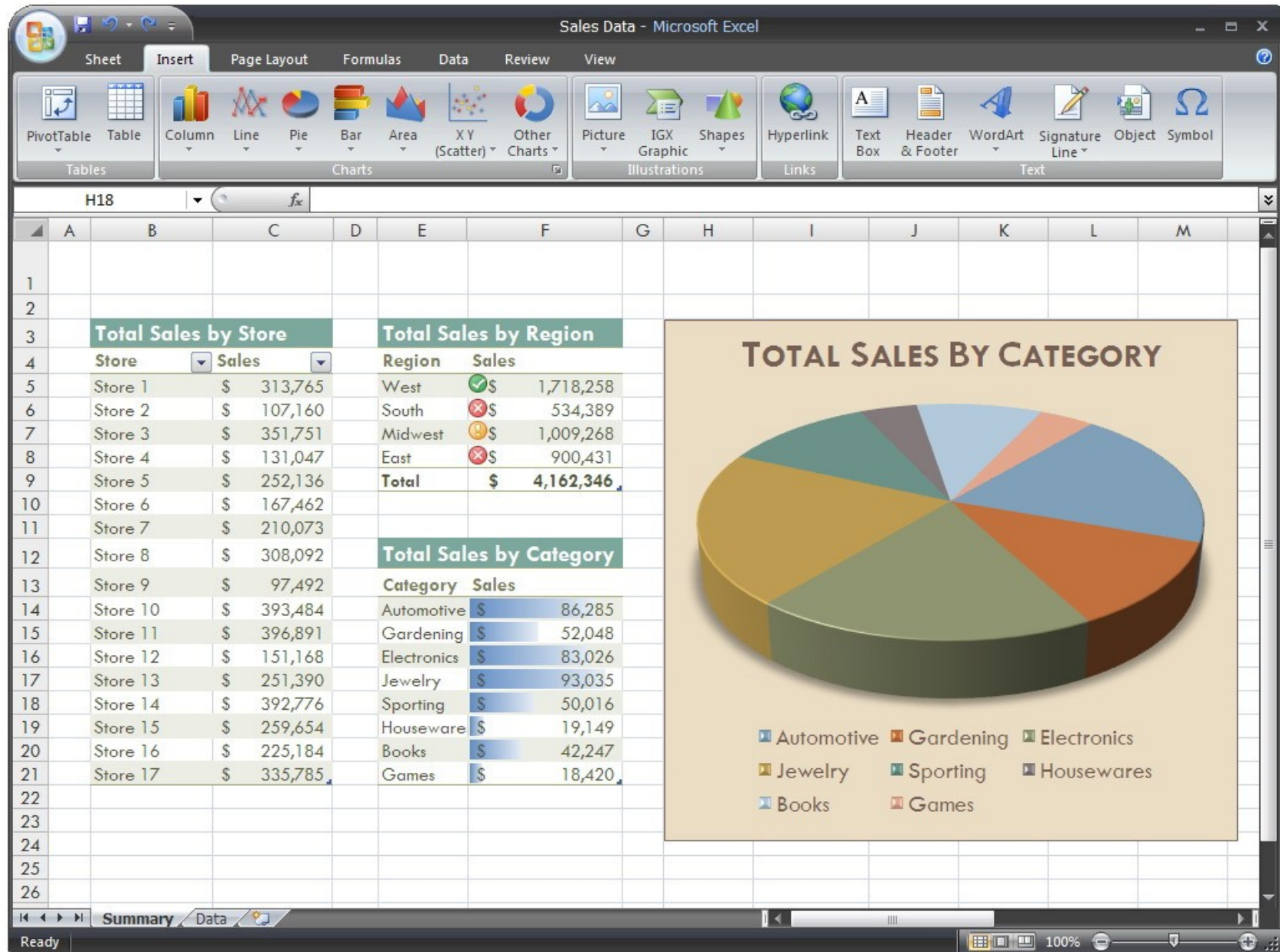
Roy

LiveScript

purescript

Functional Reactive Programming

Your mum can probably do it

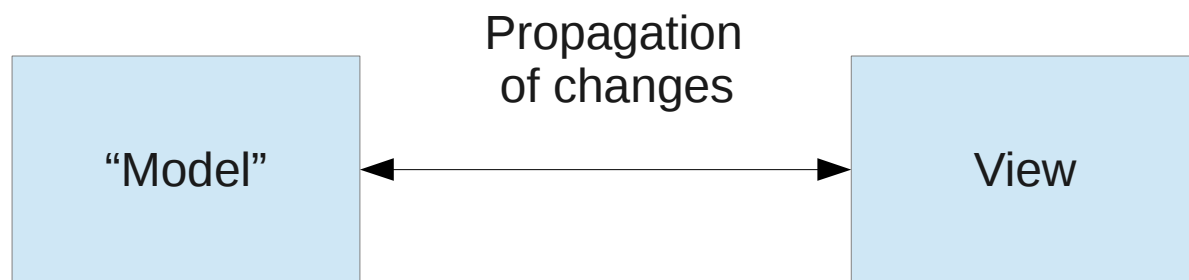


Angular is reactive

ng-repeat

ng-bind

ng-model



But it can get pretty hairy

```
scope.$watch(attrs.ngModel, function (newValue) {  
    scope.time = newValue;  
});  
  
scope.$watch('time', function (newValue) {  
    scope.parts.minute = newValue % 60;  
    scope.parts.hour = (newValue - scope.parts.minute) / 60;  
});  
  
scope.$watch('[parts.hour, parts.minute]', function (newParts) {  
    scope.time = newParts[0] * 60 + newParts[1];  
}, true);
```

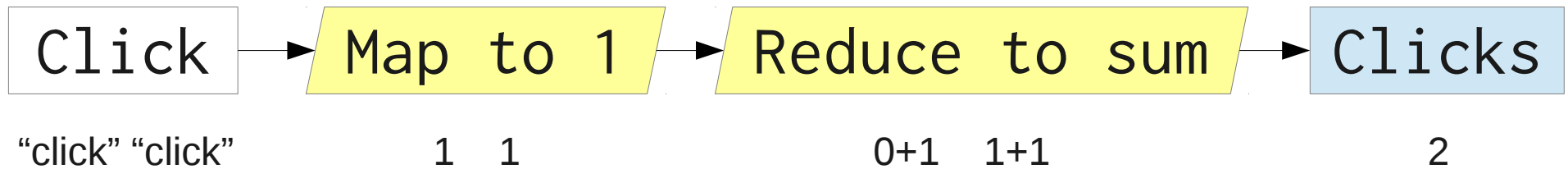
Functional Reactive Programming

A simple example

Positioning a box at the cursor

```
$('#box').css({  
  left: mouseX,  
  top: mouseY  
});
```

Event streams and properties



```
$('#box').text(clicks);
```

Promises can already do this

```
// A promise for an array of cinemas  
$scope.cinemas = cinemaService.fetchCinemas();
```

```
<ul>  
  <li ng-repeat="cinema in cinemas">  
    {{ cinema.name }}  
  </li>  
</ul>
```

Angular views are “Promises/A+ aware”

They can be composed the same too

```
fetchFilms: function (date, cinemas) {  
    return cineworldService.fetchResource('films', {  
        date: date,  
        cinema: cinemas,  
        full: true  
    })  
    .then(this.stripFilmTitlePrefix)  
    .then(this.addAlphebetisationTitle)  
    .then( __.bind(this.addPerformances, this, date, cinemas) )  
    .then(this.groupFilmVersions)  
    .then(this.addImdbData);  
},
```


Values in containers

Property



Change over
time

Promise



Success / fail
after time

Code starts life on a white board



Tests are cumbersome documentation

day should increase when i increment it

day should wrap when i increase it above max

day should decrease when i decrement it

day should wrap when i decrease it below min

day should be directly settable

day should be equal to max if set above max

day should be equal to min if set below min

day should notify when a choice causes day to be wrapped

day should notify when a choice causes day to be limited

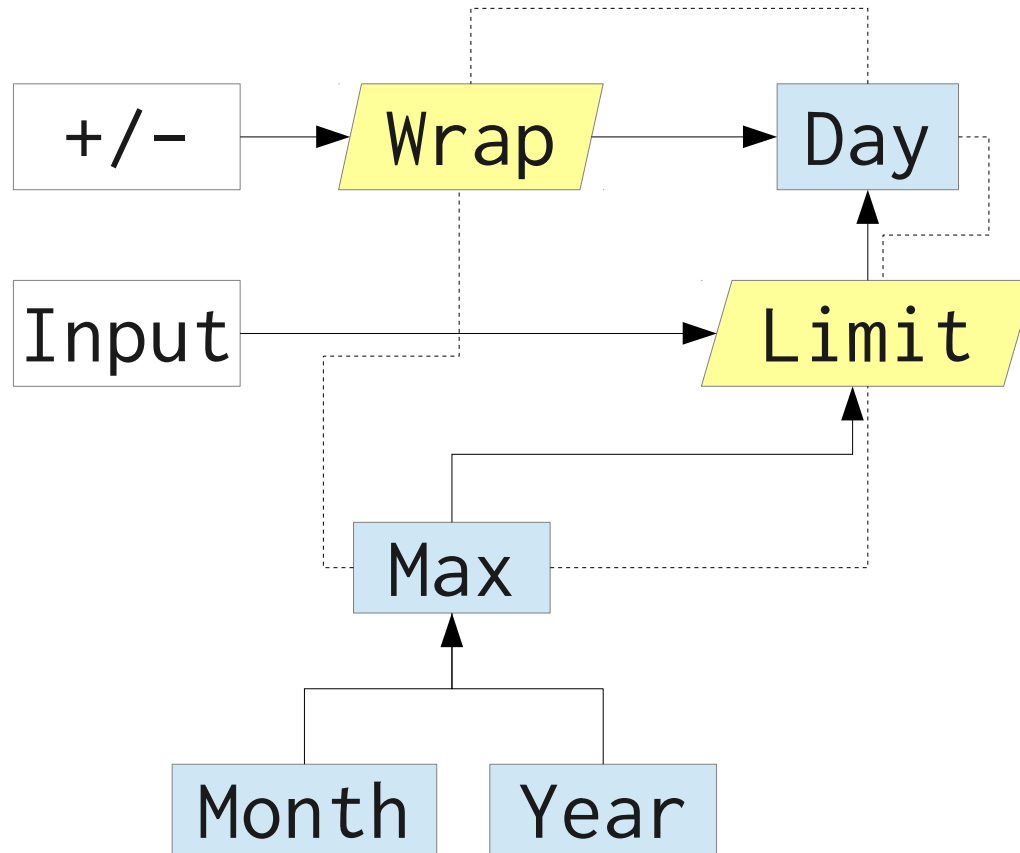
min day should be one by default

min day should be as configured in min date given month and year are at min

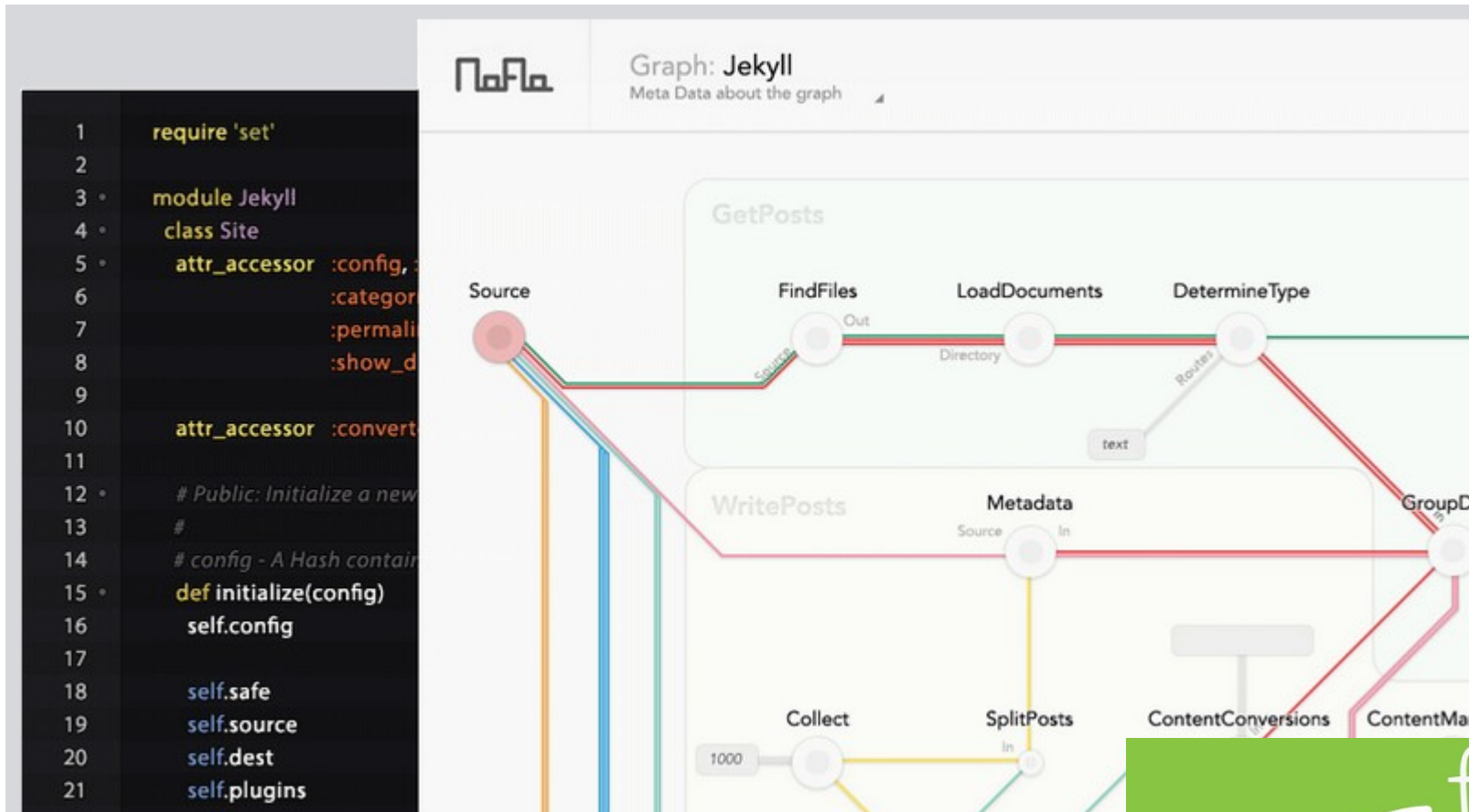
max day should be based on current month and year

...and many more

We need a high level overview



NoFlo



NoFlo Demo

<http://noflojs.org/noflo-ui/#example/6699161>



elm

```
import Mouse
import Window

main = lift2 scene Mouse.position Window.dimensions

scene (x,y) (w,h) =
  let (dx,dy) = (toFloat x - toFloat w / 2, toFloat h / 2 - toFloat y)
  in collage w h
      [ ngon 3 100 |> filled blue
        , ngon 6 30 |> rotate (atan2 dy dx)
                   |> filled orange
                   |> move (dx, dy)
        ]
```

Elm Demo

<http://elm-lang.org/edit/examples/Reactive/Transforms.elm>

Have a play

Composability

True abstractions

New tooling
opportunities

Concise code

Questions?

Resources

- **Tools**

- Famo.us <http://famo.us/>
- NoFlo <http://noflojs.org/>
- Lo-Dash <http://lodash.com/>

- **Talks**

- Bodil Stokke: What Every Hipster ... Functional Programming
- <http://vimeo.com/68331937>
- Douglas Crockford: Monads And Gonads
- <http://www.youtube.com/watch?vdkZFtingAcM>
- Evan Czaplicki: Functional Reactive Programming in Elm
- <http://www.infoq.com/presentations/elm-reactive-programming>

- **Reading**

- Functors, Applicatives, Monads
- <http://adit.io/posts/2013-04-17-functors,applicatives,andmonadsinpictures.html>

- **Coursera courses**

- <https://www.coursera.org/course/progfun>
- <https://www.coursera.org/course/reactive>

More resources

- **Languages**
- <http://livescript.net/>
- <http://roy.brianmckenna.org/>
- <http://clojure.org/clojurescript>
- <http://functorial.com/purescript/>
- <http://elm-lang.org/>