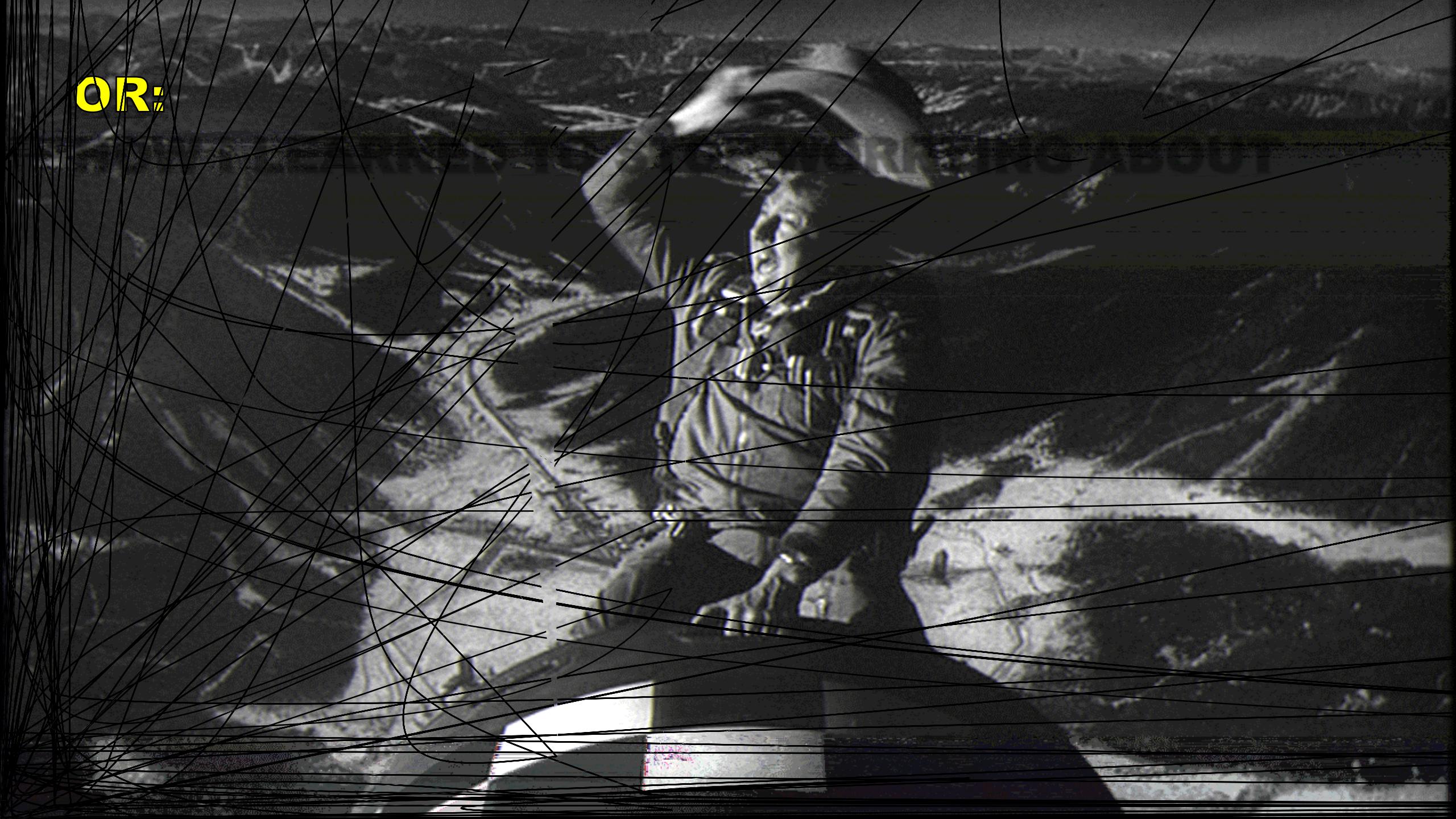




Async JavaScript at Netflix

Matthew Podwysocki @mattpodwysocki
github.com/mattpodwysocki/jsconfuy-2015

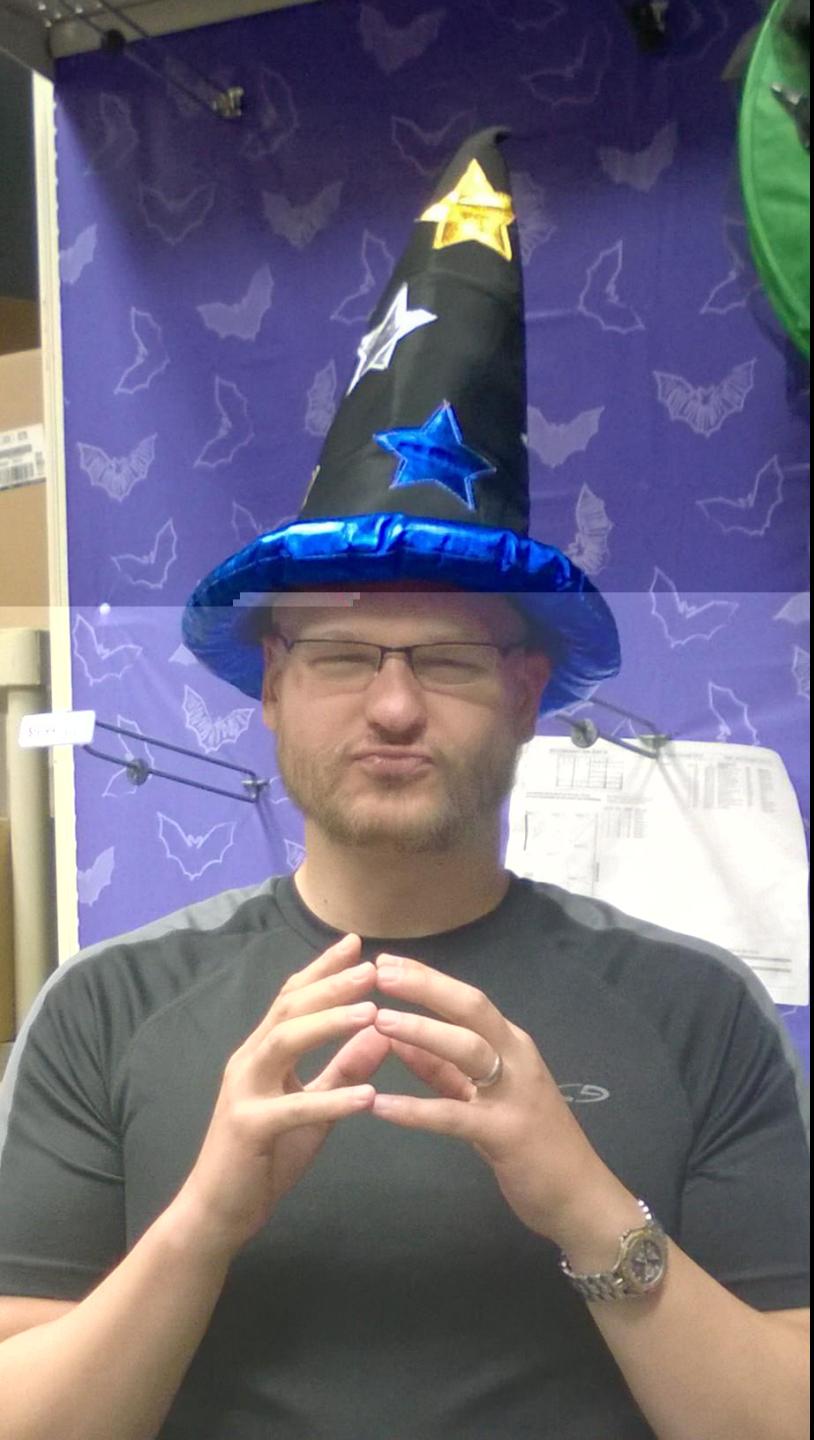
OR:



**Or "I thought I had a problem. I thought to myself,
"I know, I'll solve it with promises and events!".
have Now problems. two I**

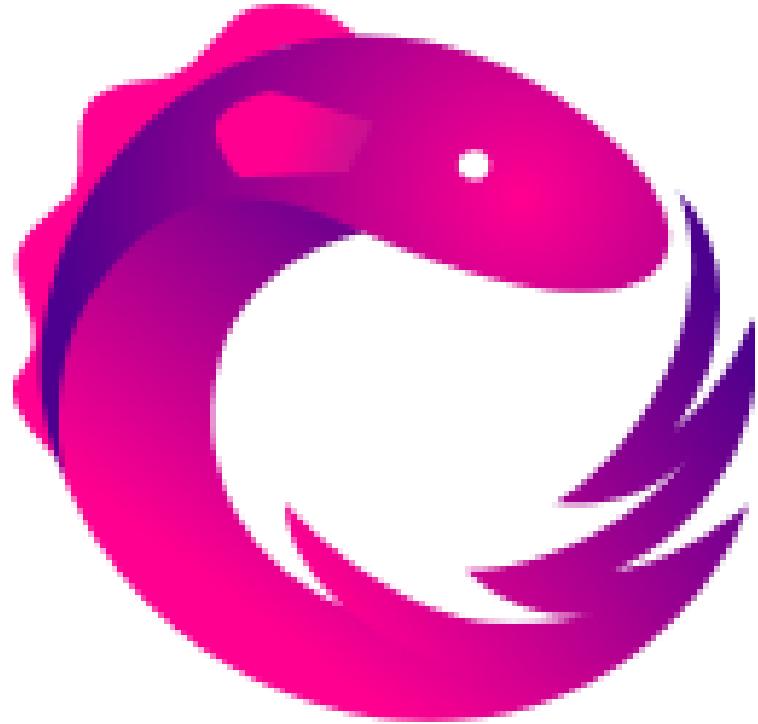


**trapd in Monad tutorl
plz help**



**Principal SDE
Open Sourcerer
@mattpodwysocki
github.com/mattpodwysocki**

MICRÖSOFT



!
(Rx)

00



NETFLIX

204

! ! ! ! !

The Netflix App is Asynchronous

- Cr r 'U c r
- Rrc g
- Fc c'Ceeg
- Cplc lqp
- lg 10 qf gnDlpf lpi



Two Years Ago...

- E qo r rg "c pej qpq "eqf g
- Enlgp "cpf 'Ug g "f g grqr g " k j n "eq r rgf
- F khg gp "r nc hq o ."f khg gp "cr r qcej g " q"c pej pq



Today at Netflix

- T " gf "qp" g g "cpf "erkgp
- 52- "f g gmqr g " lpi "T "lp"7"f khg gp "rcpi ci g
- Uco g"c pej qpq "o qf grkg g jg g



Real-Time is Everywhere...



SPOTLIGHTER

Let's Face It, Asynchronous Programming is Awful!



**“We choose to go to solve asynchronous
programming and do the other things,
not because they are easy, but because
they are hard”**



Hq o g "WJ"R g kf gp 'Lqj p"HOgppgf "/3 4
]ekc lqp"pggf gf _

Callback Hell

```
function play(movi Id, callback) {  
    var movi Tick t, playError,  
        tryFinish = function () {  
            if (playError) {  
                callback(playError);  
            } else if (movi Tick t && play r.initializ d) {  
                callback(null, tick t);  
            }  
        };  
    if (!play r.initializ d) {  
        play r.init(function (rror) {  
            playError = rror;  
            tryFinish();  
        })  
    }  
    authoriz Movi ( function ( rror, tick t) {  
        playError = rror;  
        movi Tick t = tick t;  
        tryFinish();  
    });  
});
```





culturepub.fr

next
ad ➔

Events and the Enemy of the State

```
var isDown = fals , stat ;  
  
function mous down ( ) {  
    isDown = tru ;  
    stat  = { startX:  .offs tx,  
              startY:  .offs tY; }  
}  
  
function mous mov  ( ) {  
    if (!isDown) { r turn; }  
    var d lta = {  ndX:  .cli ntX - stat .startX,  
                  ndY:  .cli nyY - stat .startY };  
    // Now do som thing with it  
}  
  
function mous up ( ) {  
    isDown = fals ;  
    stat  = null;  
}
```

```
function dispos () {  
    l m.r mov Ev ntList n r('mous down', mous down, fals );  
    l m.r mov Ev ntList n r('mous up', mous up, fals );  
    doc.r mov Ev ntList n r('mous mov ', mous mov , fals );  
}  
  
l m.addEv ntList n r('mous down', mous down, fals );  
l m.addEv ntList n r('mous up', mous up, fals );  
doc.addEv ntList n r('mous mov ', mous mov , fals );
```





First Class Async with Promises

```
play r.initializ ()  
  .th n(authoriz Movi , loginError)  
  .th n(playMovi , unauthoriz dMovi )
```

Breaking the Promise...

```
var promis ;  
  
input.addEventListner('k yup', () => {  
  
  if (promis ) {  
    // Um, how do I cancel?  
  }  
  ls {  
    promis = getdata( .target.value ).then(populateUI);  
  }  
}, fals );
```

Aborting a fetch #27

[New issue](#)[Open](#)

annevk opened this issue 26 days ago · 182 comments

od to abort something initiated with `fetch()` in a way that is not overly

sion

592

625

or `fetch()` returns an object that is more than a promise going forward or either an object or a callback that gets handed an object.

[Owner](#)**Labels**

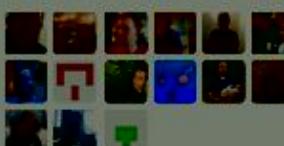
None yet

Milestone

No milestone

Assignee [jakearchibald](#)**Notifications**[Subscribe](#)

You're not receiving notifications from this thread.

15 participants

annevk commented 26 days ago

Goal

Provide developers with a method to easily abort a fetch request without complicating.

Previous discussions

- [#20](#)
- [slightlyoff/ServiceWorker#20](#)
- [slightlyoff/ServiceWorker#21](#)
- [whatwg/streams#297](#)

Viable solutions

We have two contenders. Either `fetch()` is passed something



3.0 cancellation overhaul #415

[New issue](#)

Open

petkaantonov opened this issue on Dec 27, 2014 · 79 comments



petkaantonov commented on Dec 27, 2014

Owner

The current cancellation seems to be the most problematic feature due to some annoying limits in the design:

- Awkward API for attaching the callback that cancels the work (through typed catch handler)
- Cannot compose because it is impossible to tell the difference between rejection and cancellation (e.g. `Promise.all(...).cancel()` cannot do the obvious thing and also call `cancel()` on the promises in the array)
- `.cancel()` is asynchronous

Since all consumers and producers must be trusted/"your code", there is no reason to enforce that cancellable promises are single-consumer or create new primitives.

Edit: The below design has been scratched, see [#415 \(comment\)](#)

Labels

3.0.0

Milestone

No milestone

Assignee

No one assigned

Notifications

Subscribe

You're not receiving notifications from this thread

Finally Cleaning Up...

```
var r sourc ;  
  
try {  
    var r sourc = g tR sourc ();  
} catch ( ) {  
    throw ;  
} finally {  
    r sourc && r sourc .dispos ();  
}
```

```
g tr sourc ()  
.succ ss(  
    function (r sourc ) {  
        })  
.catch(  
    function ( rr) {  
        });  
  
// How do I cl an up my r sourc ?
```



MAKE GIFS AT GIFOUP.COM



UNSAFE AT ANY SPEED

The Designed-In Dangers
of The American Automobile
By Ralph Nader



stroom prosesing



Ng "Hceg"k.'U gco 3" g g" g ldrg

- Vj g"r c g"o g j qf "f kf p
- Vj g" f c c "g gp " c gf "lo o gf lc gn." gcf "q "pq
- Ecp 'l "eqp o g"c" r ge h gf "p o dg "qh'd g
- Rc g"cpf " g o g" g g"lo r q ldrg" q'i g " k j





U gco 4"Grge le"Dqqi cmaq

– Ncpf gf "lp"20 G

– Pq " rrq gf" Qd ge "O qf g

– Hnq lpi "o qf g" g "pqp, hnq lpi "
o qf g

– lp qf egf" j g'hqmq lpi 'U gco "
enc g

- Tgcf cdrg

- Y kcdrg

- F r ng

- Rc j q i j

U gco "55"315

– Ncpf gf "lp"203304

– Cf f "eq m1 peq m1a kg



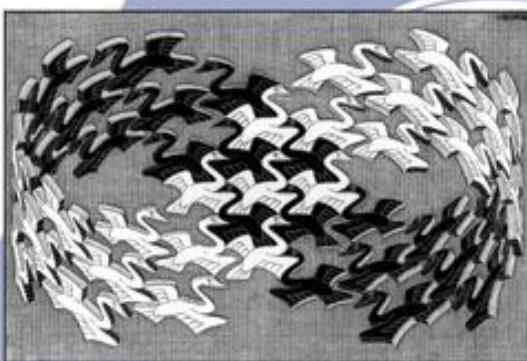
1994



Design Patterns

Elements of Reusable
Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

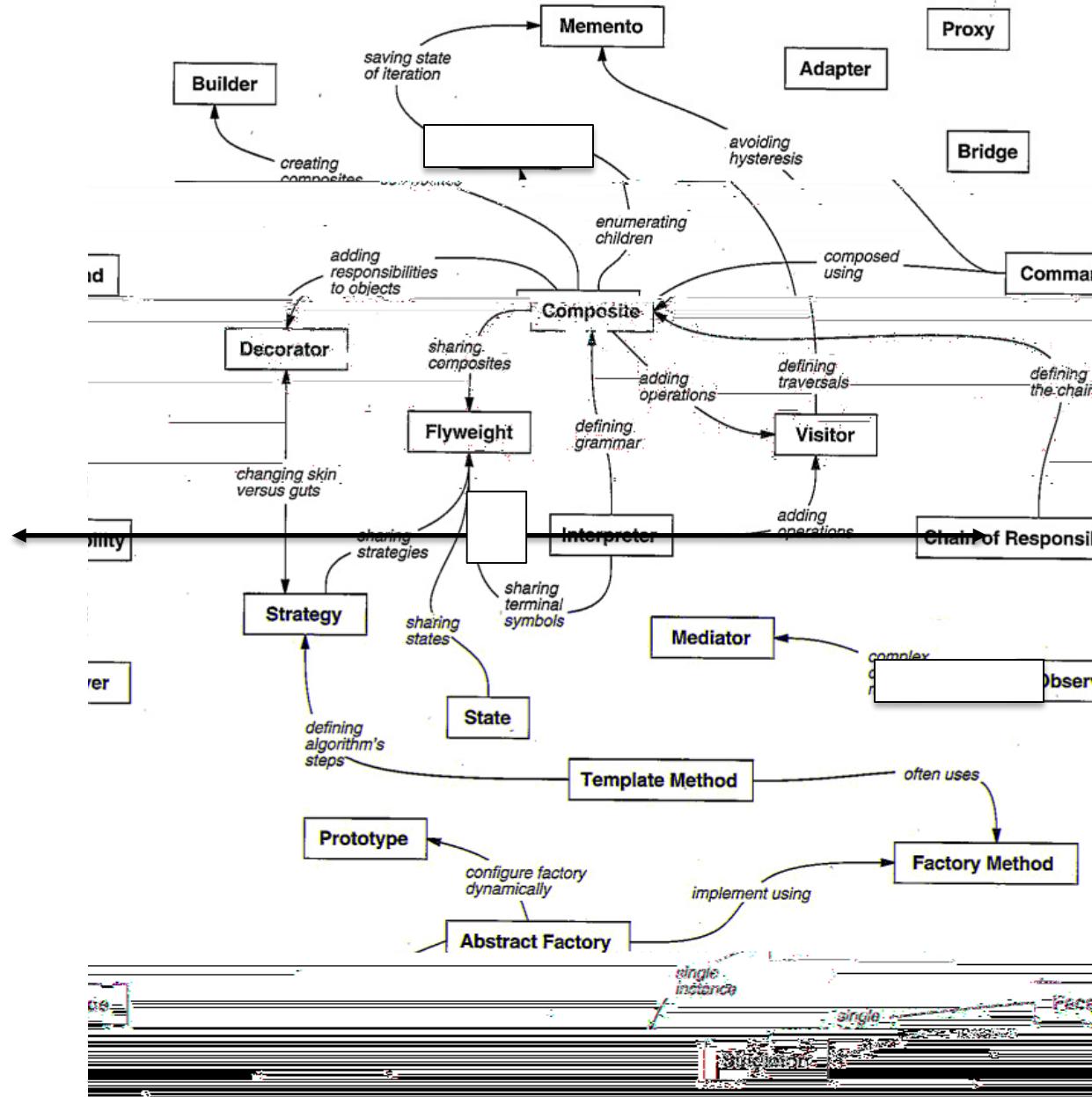


Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES





ern Relationships

Design Patte

Iterator Pattern with ES2015

```
> var iterator = getNumbers();  
> console.log(iterator.next());  
> { value: 1, done: false }  
> console.log(iterator.next());  
> { value: 2, done: false }  
> console.log(iterator.next());  
> { value: 3, done: false }  
> console.log(iterator.next());  
> { done: true }  
>
```

Observer Pattern with the DOM

```
> document.addEventListener(  
  'mousemove',  
  (e) =>  
    console.log(e);  
); ━━
```

```
> { clientX: 425, clientY: 543 }  
> { clientX: 450, clientY: 558 }  
> { clientX: 455, clientY: 562 }  
> { clientX: 460, clientY: 743 }  
> { clientX: 476, clientY: 760 }  
> { clientX: 476, clientY: 760 }  
> { clientX: 476, clientY: 760 }
```

! ! !
! !

[{x: 23, y: 44}, {x:27, y:55}, {x:27, y:55}]

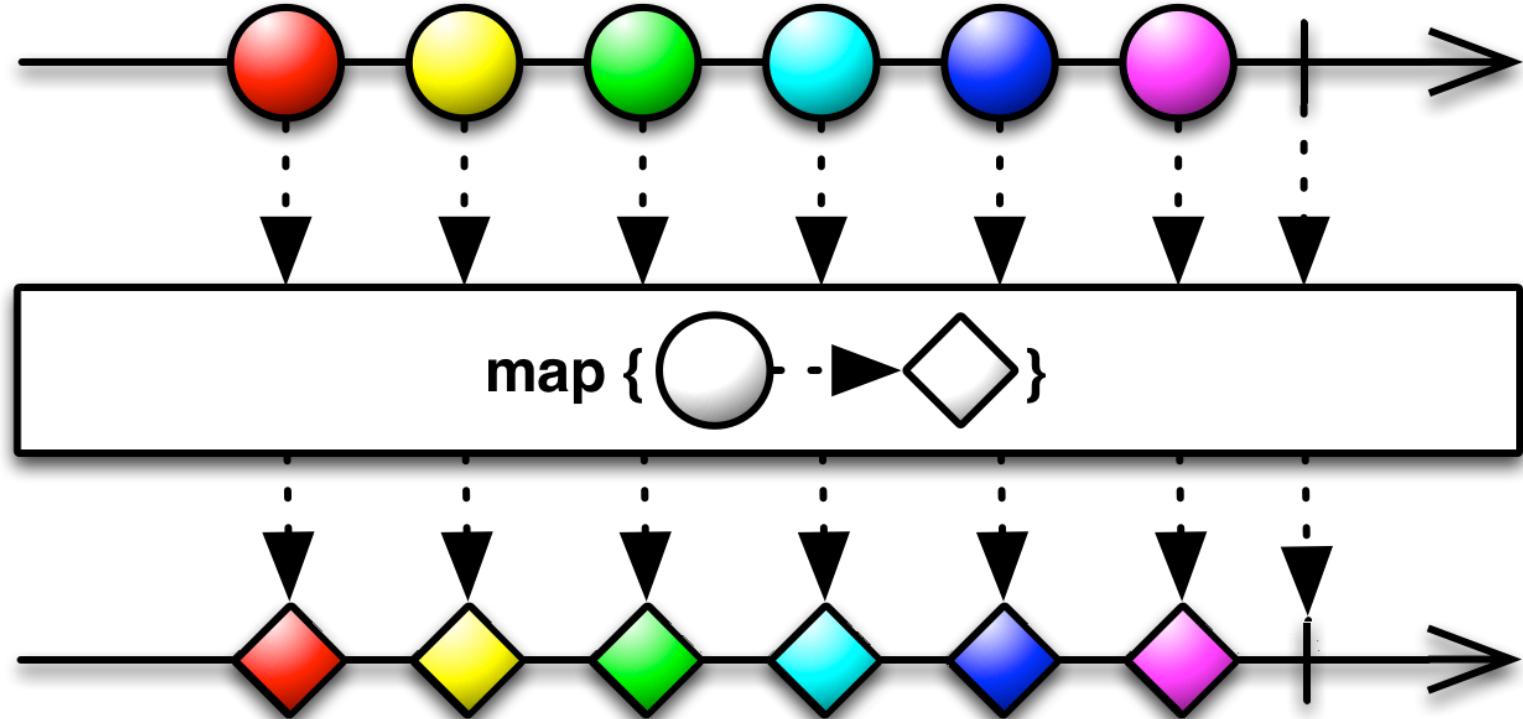


! ! !

! ! ! both /

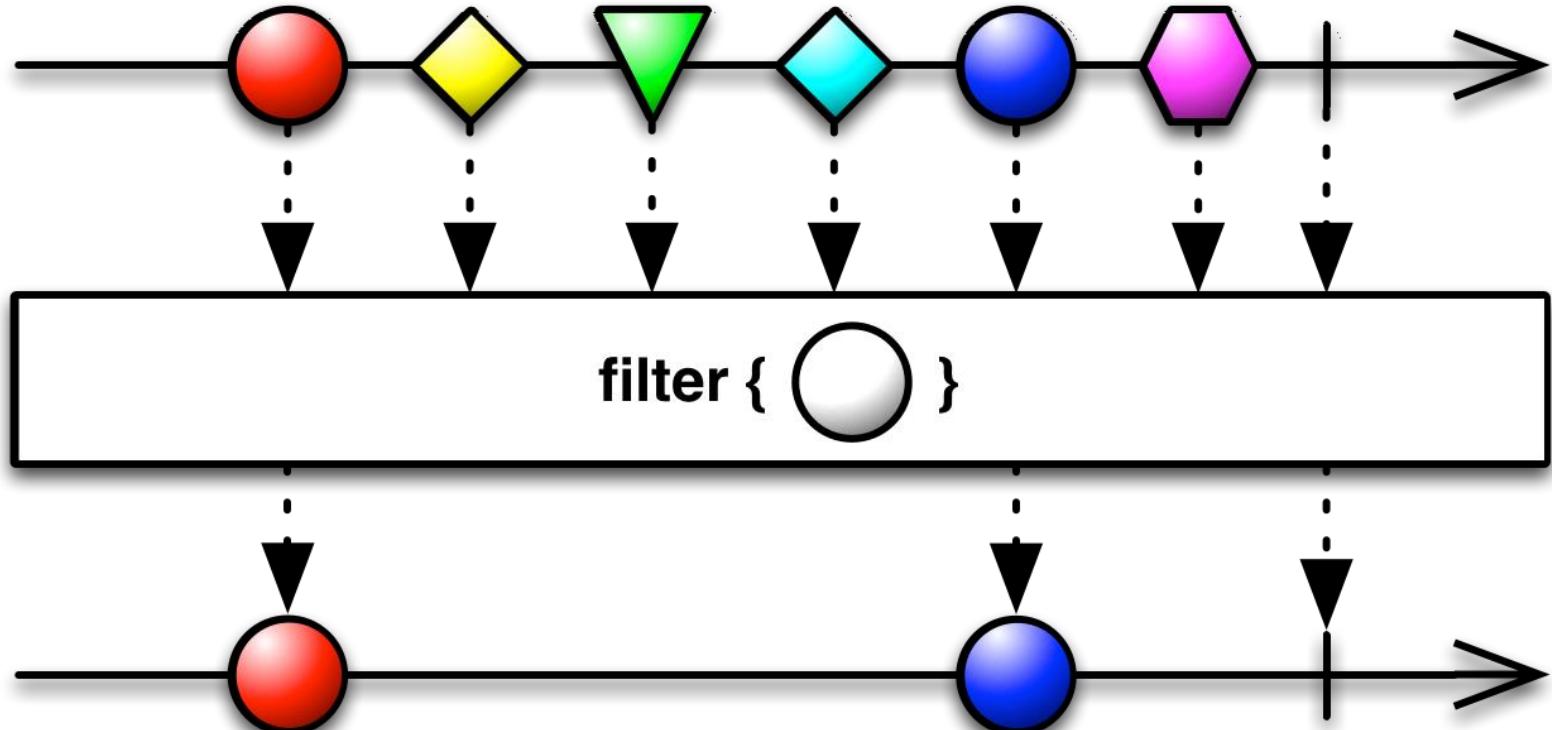
map()

V cp hq o " J g'kgo "go k gf "d "cp'E qnge lqp"d "cr r n lpi "c"
h pe lqp" q"gcej "qh' j go



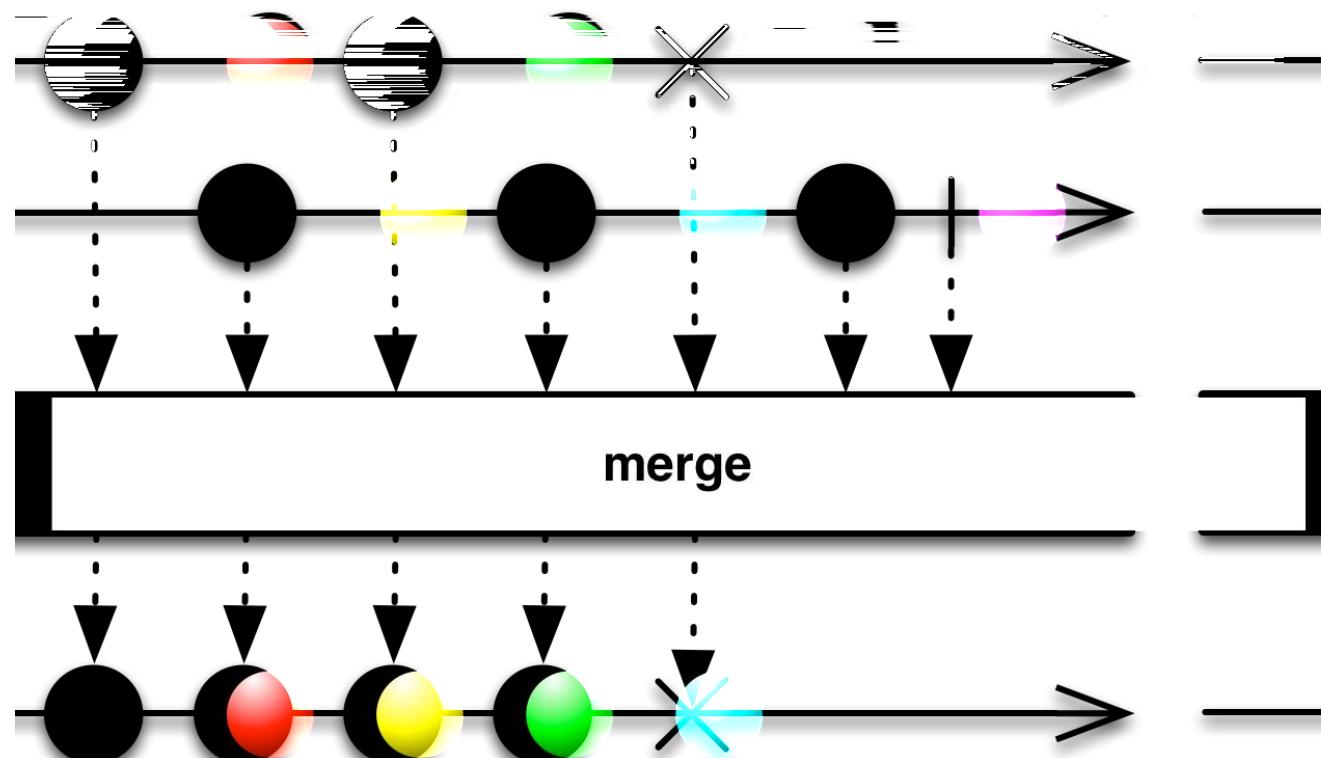
filter()

Hng 'kgo "go k gf "d "c'Eqnge kqp



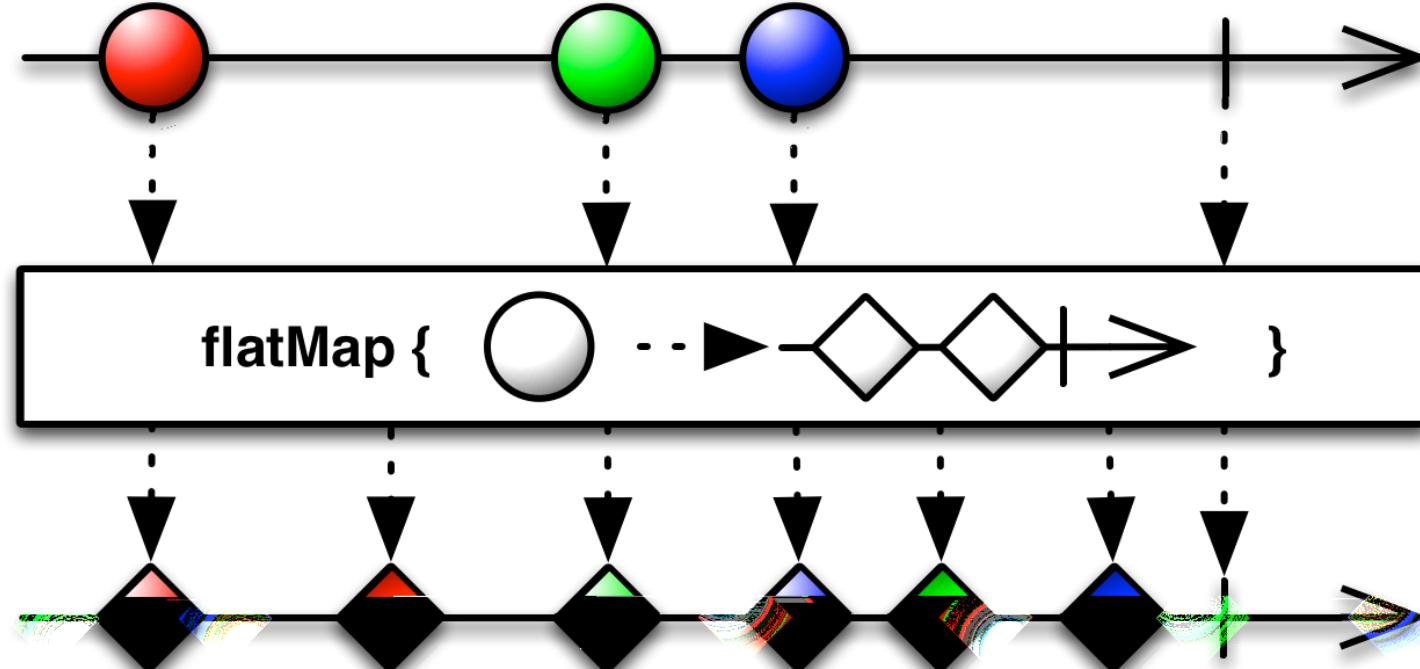
mergeAll()

eqo dlpqg'o nk rg'Eqmge lqp 'lp q'qpg'd "o g i lpi " j gk"
go k lqp



flatMap()

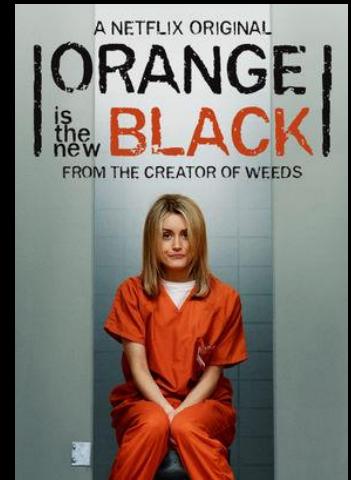
V cp hq o " J g'kgo "go k gf "d "c'Eqmge lqp'kp q"
Eqmge lqp ." J gp'hc gp" J k 'kp q"c" lpi rg'Eqmge lqp



Top-rated Movies Collection

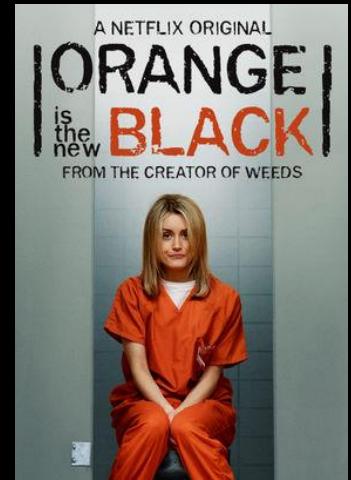
```
l t g tTopRat dFilms = (us r) => {
  us r.vid oLists
    .map((vid oList) =>
      vid oList.vid os
        .filt r((v) => v.rating === 5)
    ).m rg All();
};

g tTopRat dFilms(m )
  .forEach(displayMovi );
```

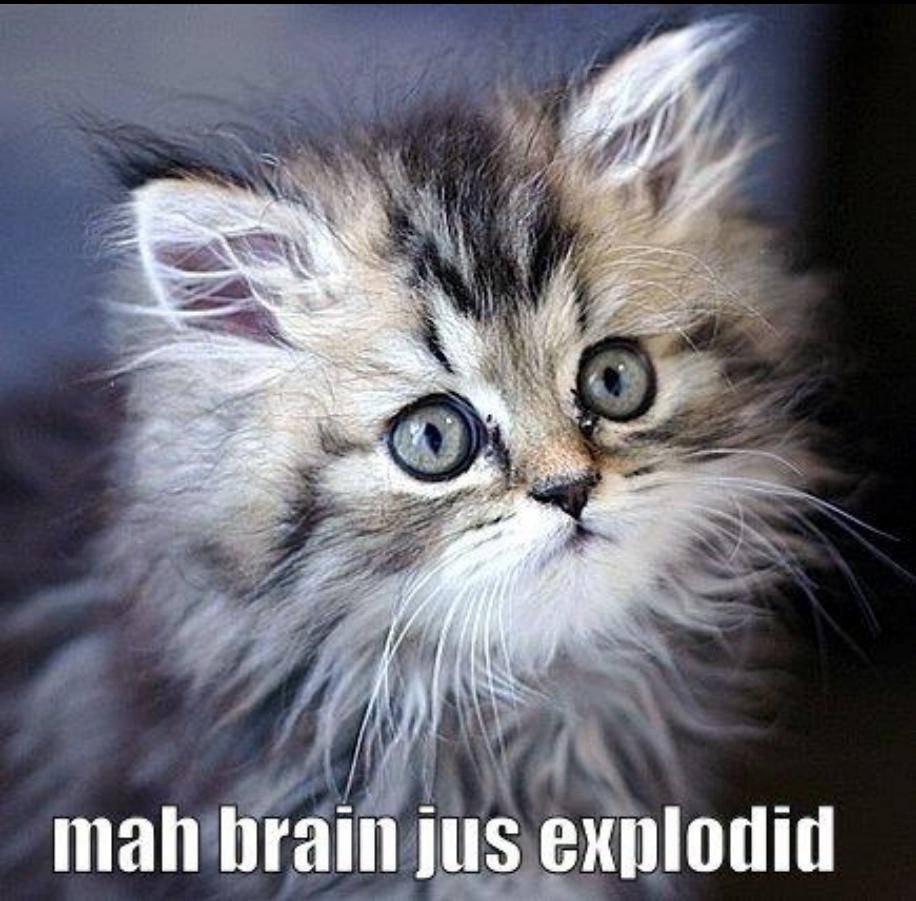


Top-rated Movies Collection

```
l t g tTopRat dFilms = (us r) =>  
  us r.vid oLists  
    .flatMap((vid oList) =>  
      vid oList.vid os  
        .filter r((v) => v.rating === 5)  
    )  
};  
  
g tTopRat dFilms(m )  
  .foreach(displayMovie );
```



What if I told you...



mah brain jus explodid

...that you could create a drag event...
...with the almost the *same code*

Mouse Drags Collection

```
l t g tEl m ntDrags = ( lmt) =>
  dom.mous down( lmt)
    .map((md) =>
      dom.mous mov (docum nt)
        .filter .takeUntil(dom.mous up( lmt));
    ).m rge All()
};

g tEl m ntDrags(imag )
  .forEach(mov Imag )
```



Mouse Drags Collection

```
l t g tEl m ntDrags = ( lmt) =>  
  dom.mous down( lmt)  
    .flatMap((md) =>  
      dom.mous mov (docum nt)  
        .filter .takeUntil(dom.mous up( lmt))  
    )  
};  
  
g tEl m ntDrags(imag )  
  .forEach(mov Imag )
```





=
Everything is a stream

First-Class Asynchronous Values

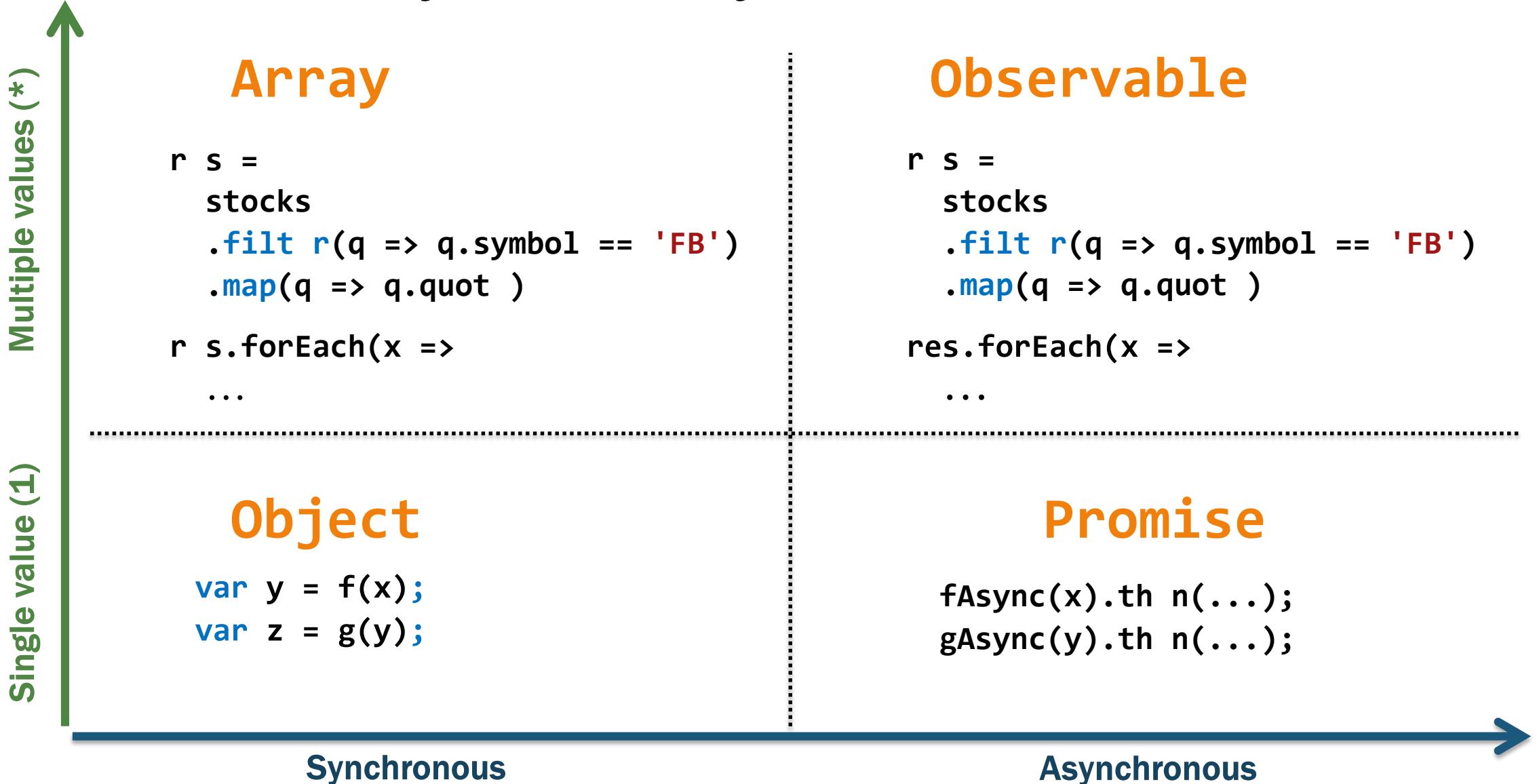
An object is first-class when it:^{[4][5]}

- can be stored in variables and data structures
- can be passed as a parameter to a subroutine
- can be returned as the result of a subroutine
- can be constructed at runtime
- has intrinsic identity (independent of any given name)



WIKIPEDIA
The Free Encyclopedia

The General Theory of Reactivity



What is Reactive Programming Anyhow?

. ! ! ! “readily responsive to a stimulus” !
/ ! ! ! ! ! ! ! ! ! ! ! !

Wanna really know what Reactive Programming Is?

! ! ! ! ! ! ! ! !
oo / 0 .

Functional Reactive Programming (FRP) is...

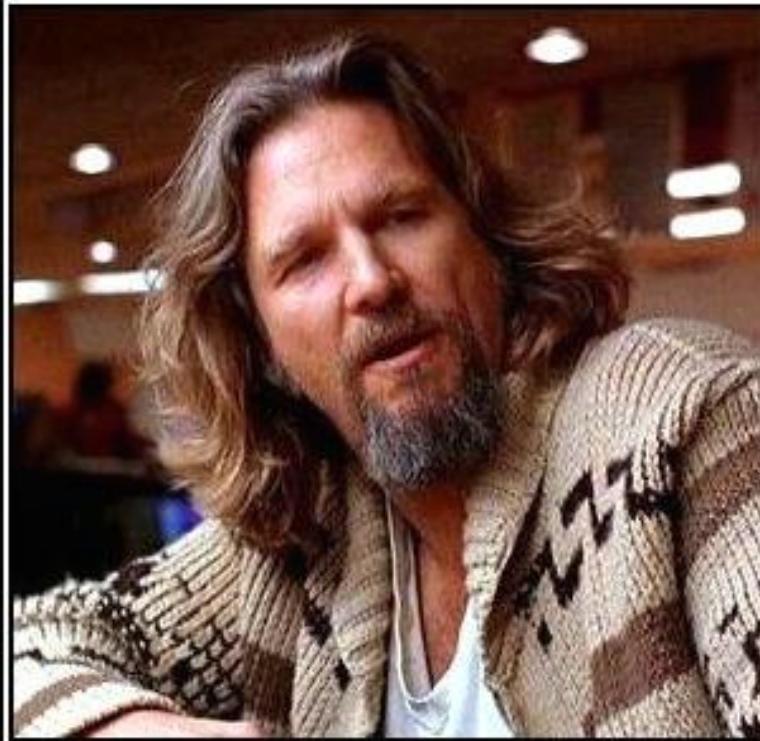
C'eqpegr "eqp k lpi "qh'

- Eqp lp q "Mlo g
- Dgj c lq " cn g "q g" lo g
- G gp "Fke g g'r j g" o gpc" kj"c" cn g"cpf "c" lo g
- Eqo r q klqpcnldgj c "lq "dgj c lq "cpf "g gp

Y j c "k" "pq

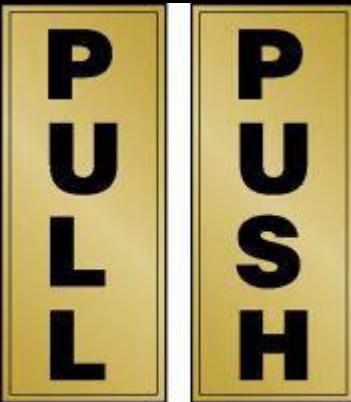
- J k j "q f g 'h pe lqp "o cr ."hng ." gf eg
- O q " q/écmgf "HT R" n

Call Us CEP, Compositional Event Processing



THE DUDE

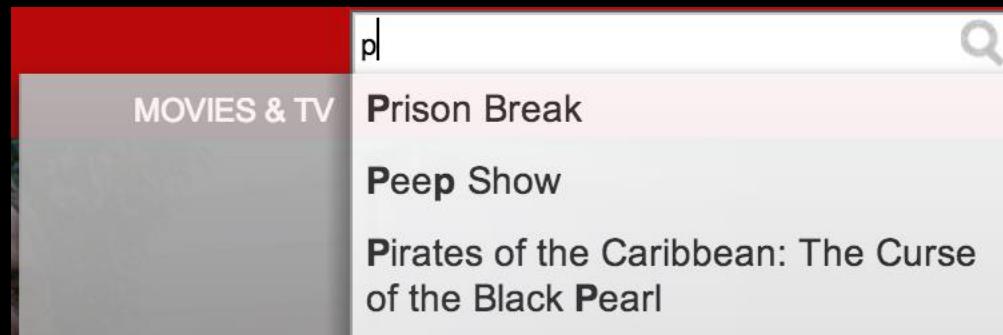
He's the Dude. So that's what you call him. You know, that or, uh, His Dudeness, or uh, Duder, or El Duderino.



```
l t sourc  = g tStockData();  
  
sourc  
.filt r((quot ) =>  
  quot .pric  > 30;  
)  
.map((quot ) =>  
  quot .pric ;  
)  
.forEach((pric ) =>  
  consol .log('> $30: +' + pric );  
);
```

```
l t sourc  = g tStockData();  
  
sourc  
.filt r((quot ) =>  
  quot .pric  > 30;  
)  
.map((quot ) =>  
  quot .pric ;  
)  
.forEach((pric ) =>  
  consol .log('> $30: +' + pric );  
);
```

Netflix Search



Netflix Search with Observables

```
let data = dom.k yup(input)
    .map(() => input.value)
    .d bounc (500)
    .distinctUntilChang d()
    .flatMapLat st(
        (term) => search(term)
    );
```

```
data.forEach((data) => {
    // Bind data to the UI
});
```

! !
!O!

```
.map(() => input.value)
    .d bounc (500)
    .distinctUntilChang d()
    .flatMapLat st(
        (term) => search(term)
    );
```

! ! ! !
! !

! ! !

! ! ! ! !

! ! ! !

What exactly is Rx?

Ncpipci g'pg cto qf gn' kj "5"eqpegr

30Qd g g1Qd g cdng

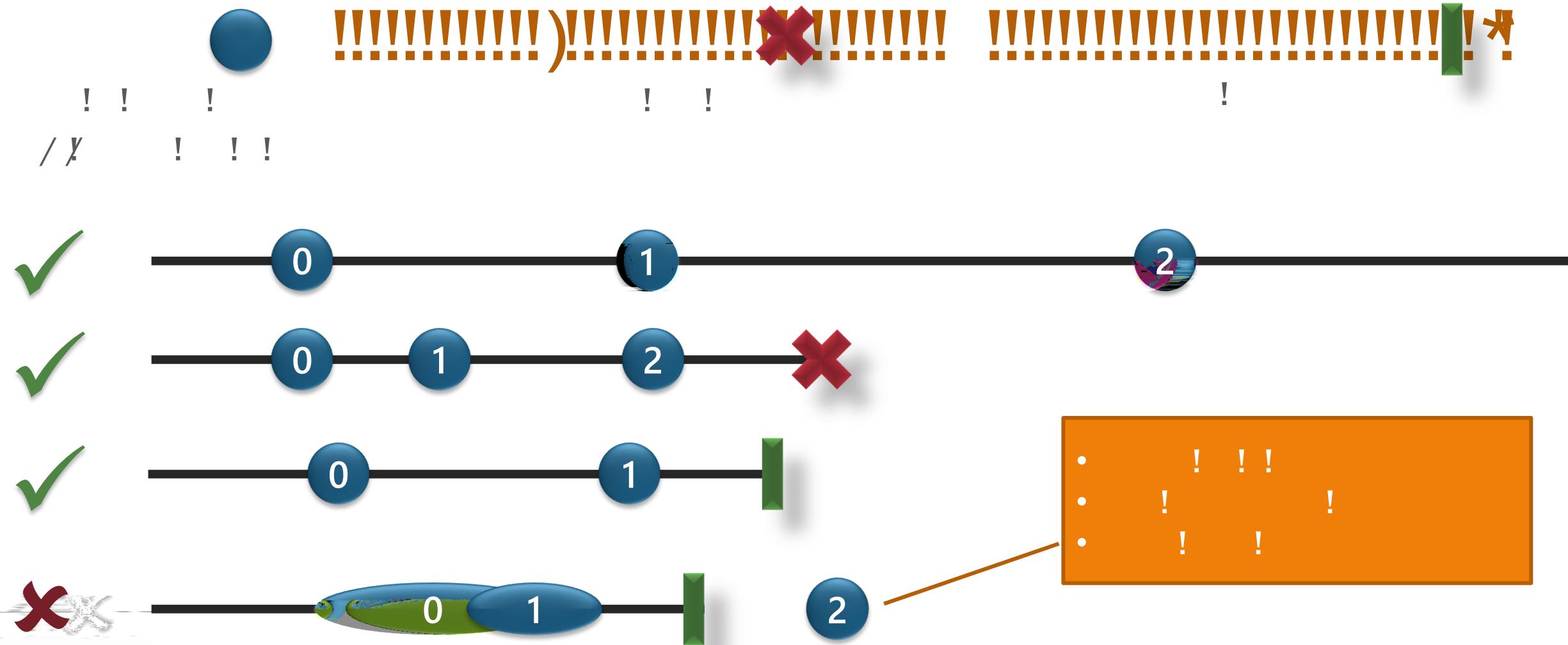
40S g "qr g c lqp " o cr1Hng1 gf eg+

50Jq 1Y j g g1Y j gp

– Uej gf rg "c" g "qh' r g " q"r c co g g k g'eqpe gpe



Rx Grammar Police



What exactly is Rx?

Ncpil ci g'pg cto qf gn' kj "5"eqpegr

30Qd g g1Qd g cdng

40S g "qr g c lqp " o cr1Hng1 gf eg+

50Jq 1Y j g g1Y j gp

– Uej gf rg "c" g "qh' r g " q"r c co g g k g'eqpe gpe





Observables - Querying UI Events

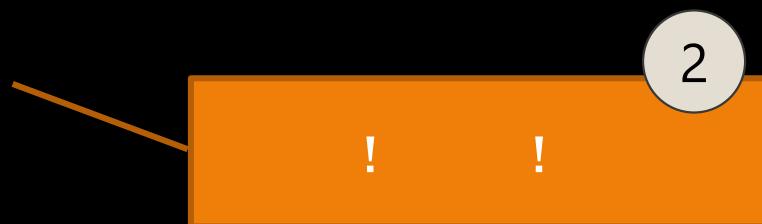
```
let mousedrag = mousedown.flatMap((md) => {  
    // calculate offsets when mouse down  
    let startX = md.offsetX,  
        startY = md.offsetY;  
  
});
```





Observables - Querying UI Events

```
let mousedown = mousedown.flatMap((md) => {  
    // calculate offsets when mouse down  
    let startX = md.offsetX,  
        startY = md.offsetY;  
  
    // calculate diffs until mouse up  
    returnmousemove.map((mm) => {  
        return {  
            left: mm.clientX - startX,  
            top: mm.clientY - startY  
        };  
    })  
});
```

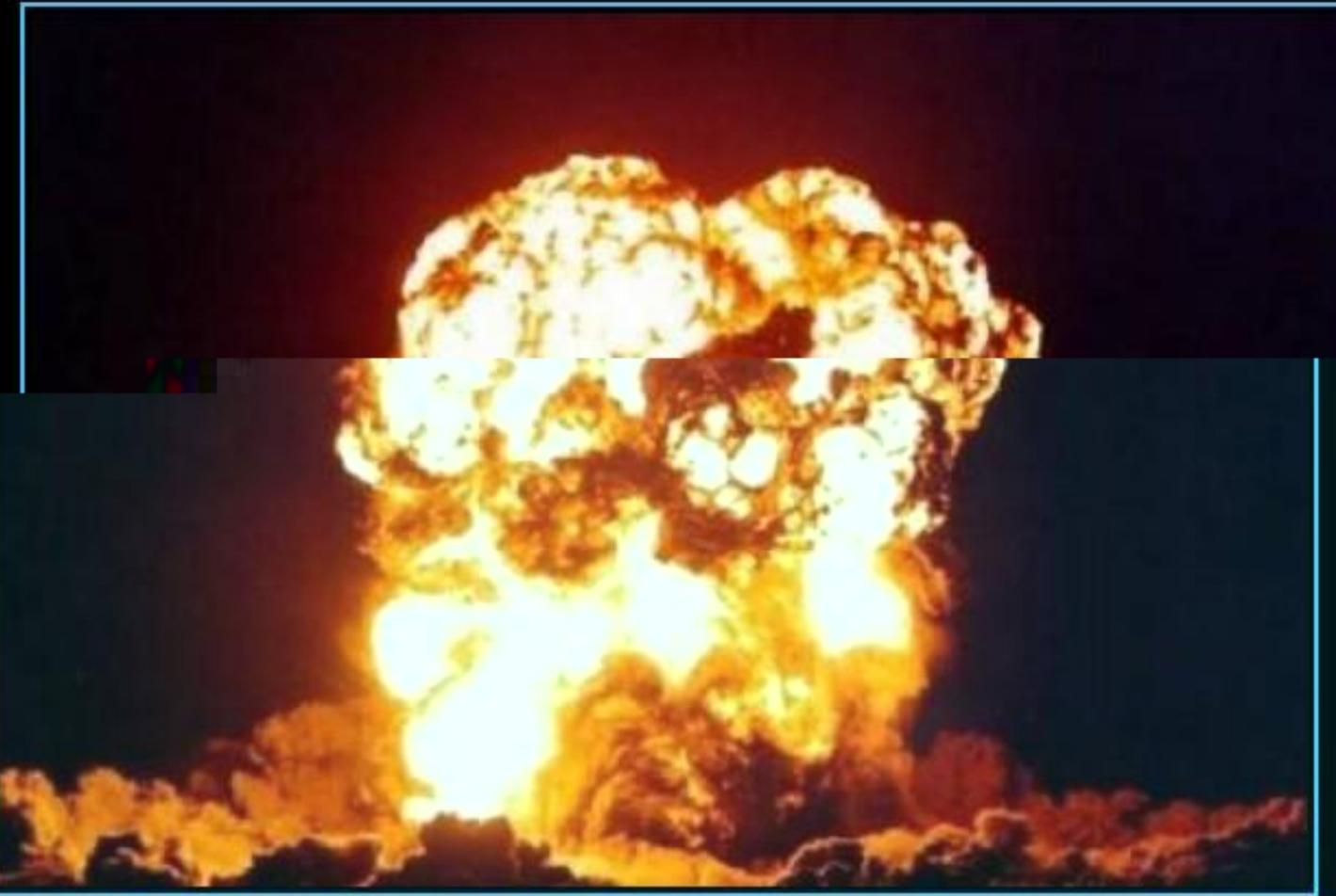




Observables - Querying UI Events

```
let mousedrag = mousedown.flatMap((md) => {  
    // calculate offsets when mouse down  
    let startX = md.offsetX,  
        startY = md.offsetY;  
  
    // calculate diffs until mouse up  
    returnmousemove.map((mm) => {  
        return {  
            left: mm.clientX - startX,  
            top: mm.clientY - startY  
        };  
    }).takeUntil(mouseup);  
});
```





PROTONIC REVERSAL

You crossed the streams, didn't you?

Your Netflix Video Lists

Netflix Row Update Polling

The screenshot shows a Netflix mobile application interface. At the top, there's a red header bar with the word "NETFLIX" on the left and "2 / 10" on the right. Below the header, there's a row of six movie thumbnails: "Band Baaja Baaraat", "The Mystery of the Sphinx", "Herod's Lost Tomb", "Arabia", "Ironclad", and another partially visible thumbnail. The main content area has a dark background. It displays the title "Top 10 for tester_jhusain_control" in white. Below this, there are two rows of movie thumbnails. The first row includes "There Will Be Blood", "Band Baaja Baaraat", "Broken English", "The Hunted", and "Raani". The second row includes "Ranveer Singh, Anushka Sharma", "Comedies, Foreign Movies", and "Director: Maneesh Sharma". At the bottom, there's a section titled "Popular on Netflix" with several more movie thumbnails.

2 / 10

Band Baaja Baaraat

2010 NR 2h 19m

★★★★★

Shruti and Bittoo decide to start a wedding planning company together after they graduate from university, but romance gets in the way of business.

Ranveer Singh, Anushka Sharma

Comedies, Foreign Movies

Director: Maneesh Sharma

Top 10 for tester_jhusain_control

Popular on Netflix

Client: Polling for Row Updates

```
function g tRowUpdat s(row) {
  let scrolls = Rx.Obsrvabl .fromEvent(document, 'scroll');
  let rowVisibility$ =
    scrolls.debounce(50)
      .map((scrollEvent) => row.isVisible(scrollEvent.offsetY))
      .distinctUntilChanged()
      .share();
  let rowShows$ = rowVisibility$.filter(v => v);
  let rowHid$ = rowVisibility$.filter(v => !v);

  return rowShows$
    .flatMap(Rx.Obsrvabl .int rval(10))
    .flatMap(() => row.getTrowData().takeUntil(rowHid$))
    .toArray();
}
```

Netflix Player



Player Callback Hell

```
function play(movieId, cancelButton, callback) {  
    var movieTicket,  
        playError,  
        tryFinish = function() {  
            if (playError) {  
                callback(null, playError);  
            }  
            else if (movieTicket && player.initialized) {  
                callback(null, ticket);  
            }  
        };  
    cancelButton.addEventListener("click", function() { playError = "cancel"; });  
    if (!player.initialized) {  
        player.init(function(error) {  
            playError = error;  
            tryFinish();  
        })  
    }  
    authorizeMovie(movieId, function(error, ticket) {  
        playError = error;  
        movieTicket = ticket;  
        tryFinish();  
    });  
});
```



Player With Observables

```
let authorizations =  
  player  
    .init()  
    .flatMap(() =>  
      playAttempts  
        .flatMap((movieId) =>  
          player.authorize(movieId)  
            .retry(3)  
            .takeUntil(cancels));  
    )  
);  
authorizations.forEach(  
  (license) => player.play(license),  
  (error) { showDialog("Sorry, can't play right now."); });
```



RxSocketSubject

[build](#) [passing](#)

A more advanced web socket wrapper for RxJS

Install

With bower:

```
bower install -S rx-socket-subject
```

With npm:

```
npm install -S rx-socket-subject
```

Goals

The goals of this project is to produce an observable WebSocket implementation that meets a set of common needs for projects I'm currently working on. RxJS-DOM has a fine WebSocket implementation, which I modelled the initial work on this implementation off of. However, I need something that also does the following:

```
l t sock t = RxSock tSubj ct.cr at (
  ['ws://n tflix.com/sock t1',
   'ws://n tflix.com/sock t2',
   'ws://n tflix.com/sock t3']);

sock t.forEach(( ) => {
  if ( .data === ' nd') {
    sock t.onCompl t d();
  }
  if ( .data === 'bad data') {
    sock t.onError(n w Error('bad data'));
  }
});

sock t.onN xt('som  data');
```

```
var d = g tData();  
  
d.r try(3)  
  .catch(d faultData)  
  .finally(() => d.cl anup())  
  .forEach(  
    (data) => {  
  
  },  
  ( rr) => {  
  
});
```

C:\>DIR A:

Not ready reading drive A
Abort, Retry, Fail?_

What is Rx?

Ncpil ci g'pg c"o qf gn" kj "5"eqpegr

30Qd g g1Qd g cdng

40S g "qr g c lqp " o cr1Hng1 gf eg+

50Jq 1Y jg g1Y jgp

– Uej gf rg "c" g "qh' r g " q"r c co g g kg"eqpe gpe



The Role of Schedulers

Mg " g lqp

- J q " q" p" l o g A
- Y j g g" q" r qf eg'g gp A
- Pg gf" q" pej qpk g" k j" j g" WA

Uej gf rg "c g" j g'cp g

- Uej gf rg "k p qf eg'e qpe gpe
- Qr g c q "c g" r c co g g k gf" d " ej gf rg
- R q k g " g " d g p g h k "c " gm

```
let d = scheduler.schedule(  
  () => {  
    // Asynchronously  
    // running work  
  },  
  1000);
```

```
d.dispose(); !
```

Testing concurrent code: made easy!

```
let scheduler = new TestScheduler();  
  
let input = scheduler.createHotObservable(  
    onNext(300, 'JSConf.UY'),  
    onNext(400, '2015'),  
    onCompleted(500));  
  
let results = scheduler.startWithCreate(() =>  
    input.pluck('length')  
);  
  
results.messages.assertEqual(  
    onNext(300, 9),  
    onNext(400, 4),  
    onCompleted(500));
```

Observables and Backpressure

```
[ g ."Qd g cdrg "ecp"j c g'dcemr g g  
- Ecp"dg'mq r c cdrg." cor ng." j q rg+  
- Ecp"dg'mq rg " d hqg ."r c cdrgD hqg gf ."eqp qm gf +
```

```
l t pausabl = chattyObs rvabl .pausabl Buff r d();  
pausabl .paus ();  
pausabl .r sum ();
```

```
l t controll d = chattyObs rvabl .controll d();  
controll d.r qu st(10);
```

Ur Kitteh of Death

Awaits

REFLETION

Async/Await

Ego lpi " q"c'Lc cUe k "Gpi lpg"Pgc '[q

– Cf f "c pe cpf "c ck"ng q f 'hq "R qo k g

– Ceegr gf "lp q'U ci g'3"qh'GE O C Ue k "9"lp'Lcp c "4236

```
async function chainAnimationsAsync( l m, animations) {  
  l t r t = null;  
  try {  
    for (l t anim of animations) {  
      r t = await anim( l m);  
    }  
  } catch ( ) { /* ignor and k p going */ }  
  r turn r t;  
}
```

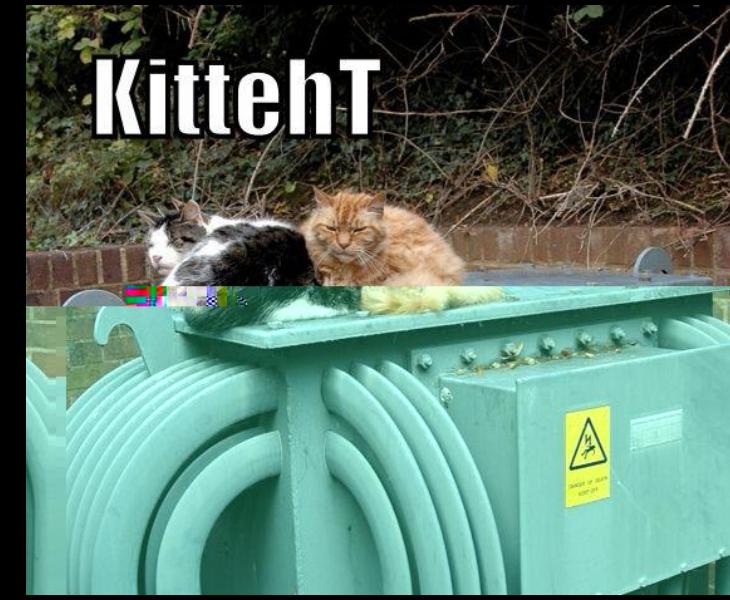
Async/Await with Observables and Generators...

T W cpf'1 gpg c q

- Cf f "c pe1c ck"ecr cdkkkg " q" lpi rg" cn g"Qd g cdrg
- C ckcdrg"lp"cp " p lo g" jc "jc '1 gpg c q

```
Rx.spawn(function* () {
  var r sult = yi ld g t('http://jsconf.uy')
    .r try(3)
    .catch(cach dv rsion)
    .finally(doCl anup);

  consol .log(r sult);
})();
```



Async Generators

GU9"cpf "Dg qpf

- Hk "enc "g gp "lp" j g'Lc cUe k " p log
- R qr q gf "lp'L pg"4236"c "ME5

```
async function* g tDrags( l m nt) {  
  for (l t mous down on l m nt.mous down) {  
    for (l t mous mov on  
      docum nt.mous mov .tak Until(docum nt.mous up)) {  
        yi ld mous mov ;  
      }  
    }  
  }  
}
```

This is an interactive learning course with exercises you fill out right in the browser. If you just want to browse the content click the button below:

Show all the answers so I can just browse.

Functional Programming in Javascript

Functional programming provides developers with the tools to abstract common collection operations into reusable, composable building blocks. You'll be surprised to learn that most of the operations you perform on collections can be accomplished with **five simple functions**:

1. map
2. filter
3. concatAll
4. reduce
5. zip

Here's my promise to you: if you learn these 5 functions your code will become shorter, more self-descriptive, and more durable. Also, for reasons that might not be obvious right now, you'll learn that these five functions hold the key to easily simplifying asynchronous programming. Once you've finished this tutorial you'll also have all the tools you need to avoid race conditions, propagate and handle asynchronous errors, and sequence events and AJAX requests. In short, these 5 functions will probably be the most powerful, flexible, and useful functions you'll ever learn.

RxMarbles

Interactive diagrams of Rx Observables

TRANSFORMING OPERATORS

[delay](#)

[delayWithSelector](#)

[findIndex](#)

[map](#)

[scan](#)

[throttle](#)

[throttleWithSelector](#)

COMBINING OPERATORS

[combineLatest](#)

[concat](#)

[merge](#)

[sample](#)

[startWith](#)

[zip](#)

FILTERING OPERATORS

[distinct](#)

[distinctUntilChanged](#)

[elementAt](#)

[filter](#)



merge



SWEETEN YOUR JAVASCRIPT

[Compile](#)[Eval](#)[Step 0](#) readable names auto-compile
 macro highlighting[vim](#)
[emacs](#)
[default](#)

```
1 /*  
2  * welcome to sweet.js!  
3  
4  You can play around with macro writing here on the left side and  
5  your code will automatically be compiled on the right. This page  
6  will also save your code to localStorage on every successful  
7  compile so feel free to close the page and come back later!  
8 */  
9  
10 // Here is a really simple identity macro to get started.  
11  
12 // The `macro` keyword is used to create and name new macros.  
13 macro id {  
14     rule {  
15         // after the macro name, match:  
16         // (1) a open paren  
17         // (2) a single token and bind it to '$x'  
18         // (3) a close paren  
19         ($x)  
20     } => {  
21         // just return the token we bound to '$x'  
22         $x  
23     }  
24 }  
25 id ( 42 );  
26  
27 // Note that a single token to sweet.js includes matched  
28
```

```
1 42;  
2 // Note that a single token to sweet.js includes matched  
3 // delimiters not just numbers and identifiers. For example,  
4 // an array with all of its elements counts as one token:  
5 [  
6     1,  
7     2,  
8     3  
9 ];  
10 // one of the really important things sweet.js does is protect  
11 // macros from unintentionally binding or capturing variables they  
12 // weren't supposed to. This is called hygiene and to enforce hygiene  
13 // sweet.js must carefully rename all variable names.  
14 var x;  
15 var foo = 100;  
16 var bar = 200;  
17 var tmp = 'my other temporary variable';  
18 var tmp$2 = bar;  
19 bar = foo;  
20 foo = tmp$2;
```

