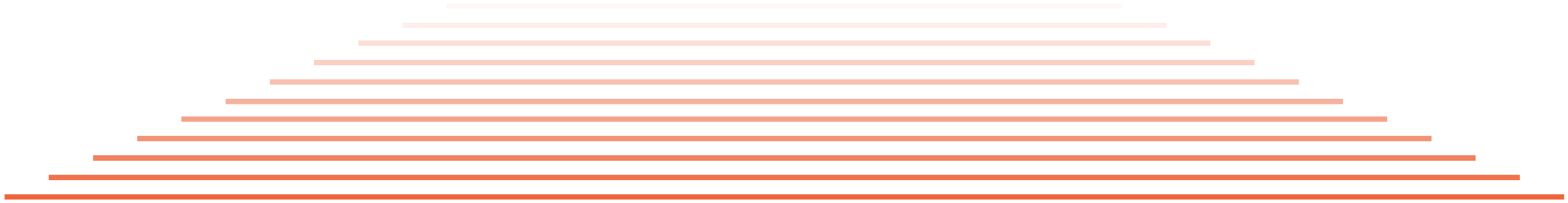
Conquering Time with FRP



A practical introduction to
Functional Reactive Programming

by sergi mansilla | @sergimansilla

@sergimansilla

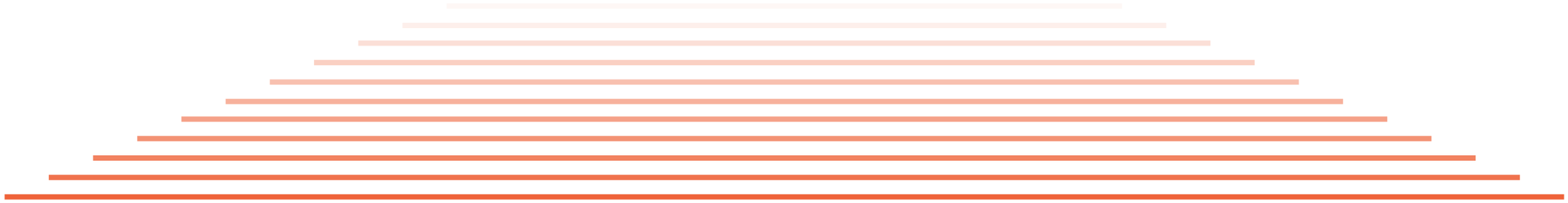
http://github.com/sergi

Alumni

telenor

Shameless linkbaiting

Conquering Time with FRP



A practical introduction to
Functional Reactive Programming

by sergi mansilla | @sergimansilla

Tame your async code with this one weird trick!

A practical introduction to
Functional Reactive Programming

by sergi mansilla | @sergimansilla

- Linkbaiting

- Time

Human beings
have hard-wired **time**
in their brain

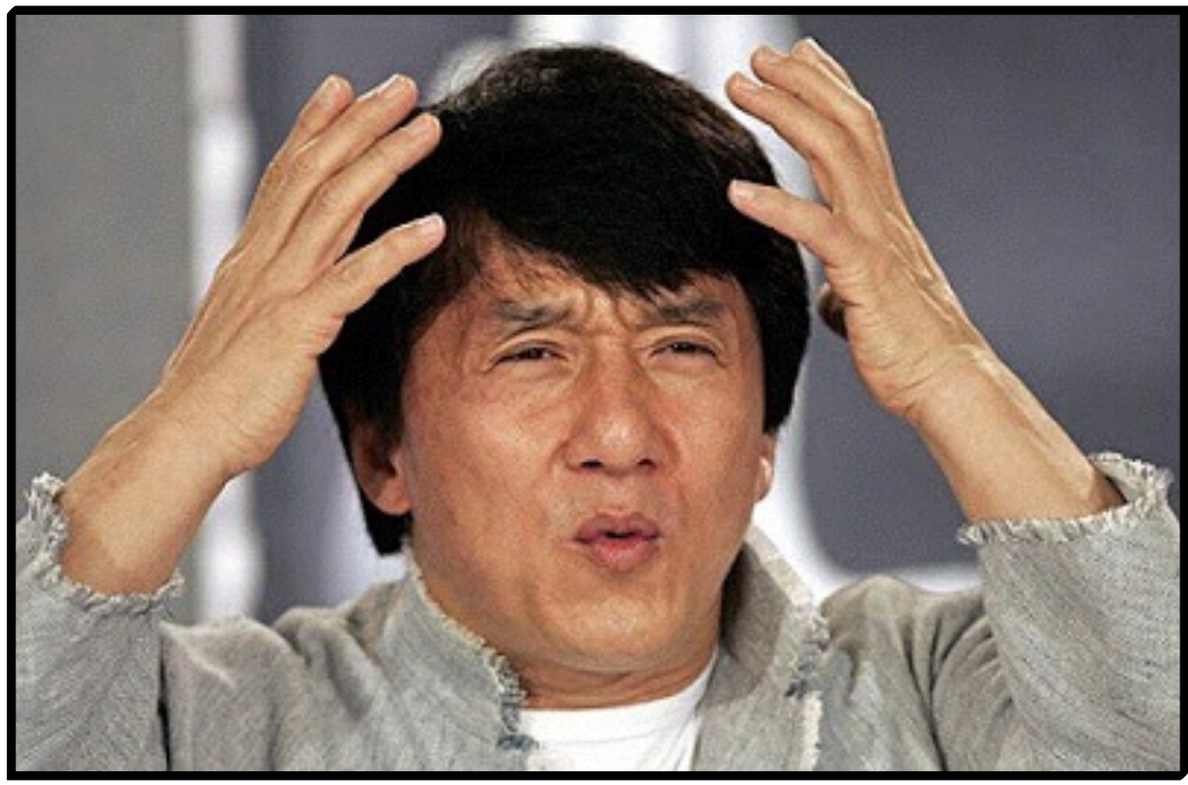
JS developers have hard-wired async in their brain

Callbacks

Promises

Generators

Events



We use **events**
to deal with
asynchronous tasks

```
var clicks = 0;
document.addEventListener('click', function register(e) {
  if (clicks < 10) {
    if (e.clientX > innerWidth / 2) {
      console.log(e.clientX, e.clientY);
      clicks += 1;
    }
  } else {
    document.removeEventListener('click', register);
  }
});
```

Why are we still micromanaging code?

```
var clicks = 0;
document.addEventListener('click', function register(e) {
  if (clicks < 10) {
    if (e.clientX > innerWidth / 2) {
      console.log(e.clientX, e.clientY);
      clicks += 1;
    }
  } else {
    document.removeEventListener('click', register);
  }
});
```

```
var clicks = 0;
document.addEventListener('click', function register(e) {
  if (clicks < 10) {
    if (e.clientX > innerWidth / 2) {
      console.log(e.clientX, e.clientY);
      clicks += 1;
    }
  } else {
    document.removeEventListener('click', register);
  }
});
```

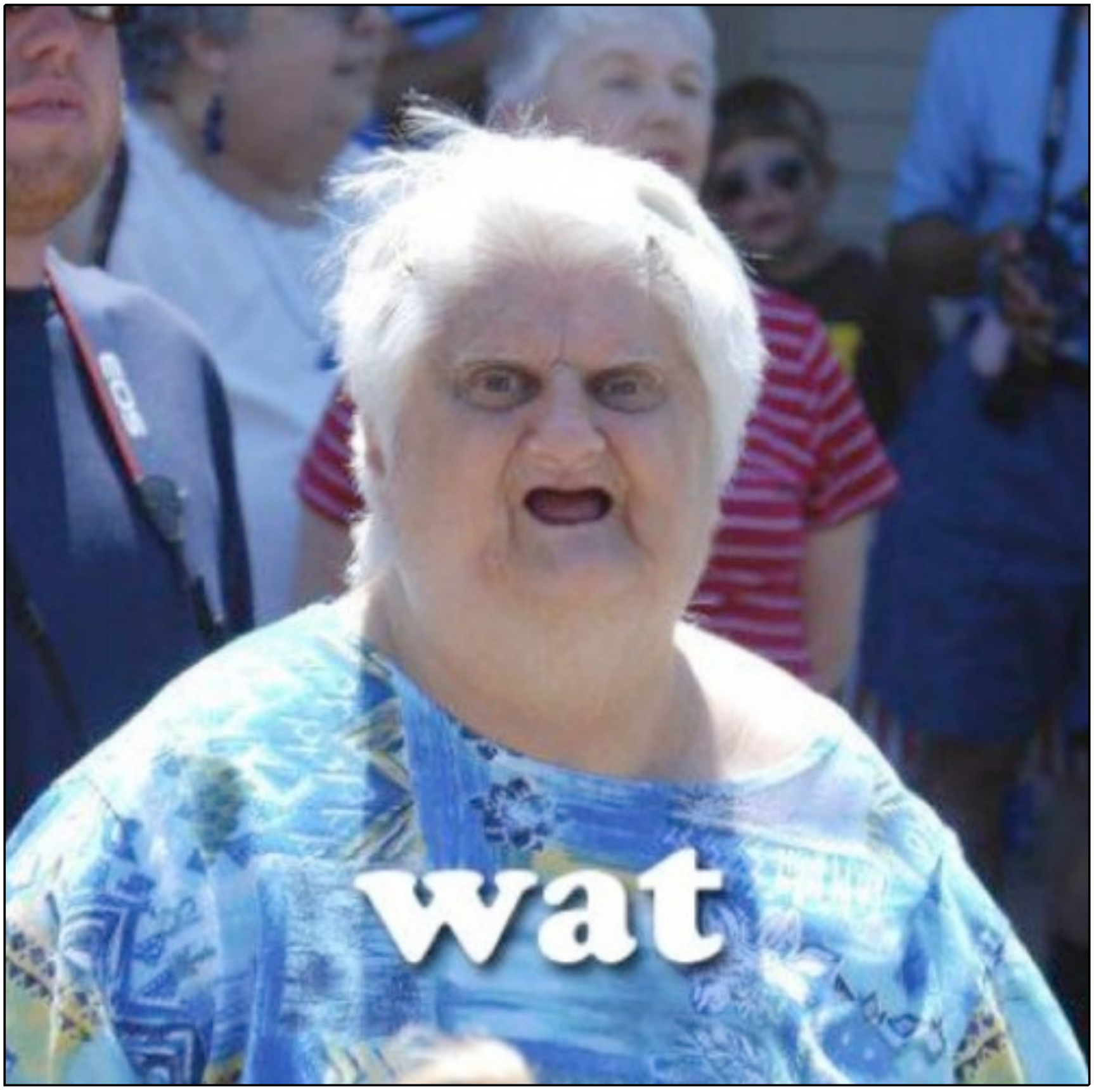
```
var clicks = 0;
document.addEventListener('click', function register(e) {
  if (clicks < 10) {
    if (e.clientX > innerWidth / 2) {
      console.log(e.clientX, e.clientY);
      clicks += 1;
    }
  } else {
    document.removeEventListener('click', register);
  }
});
```

```
var clicks = 0;
document.addEventListener('click', function register(e) {
  if (clicks < 10) {
    if (e.clientX > innerWidth / 2) {
      console.log(e.clientX, e.clientY);
      clicks += 1;
    }
  } else {
    document.removeEventListener('click', register);
  }
});
```

```
var clicks = 0;
document.addEventListener('click', function register(e) {
  if (clicks < 10) {
    if (e.clientX > innerWidth / 2 && isAPressed) {
      console.log(e.clientX, e.clientY);
      clicks += 1;
    }
  } else {
    document.removeEventListener('click', register);
  }
});

var isAPressed = false;
document.addEventListener('keydown', e => {
  isAPressed = e.keyCode === 65;
}, false);

document.addEventListener('keyup', e => {
  isAPressed = false;
}, false);
```

We still code the **how**
instead of the **what**

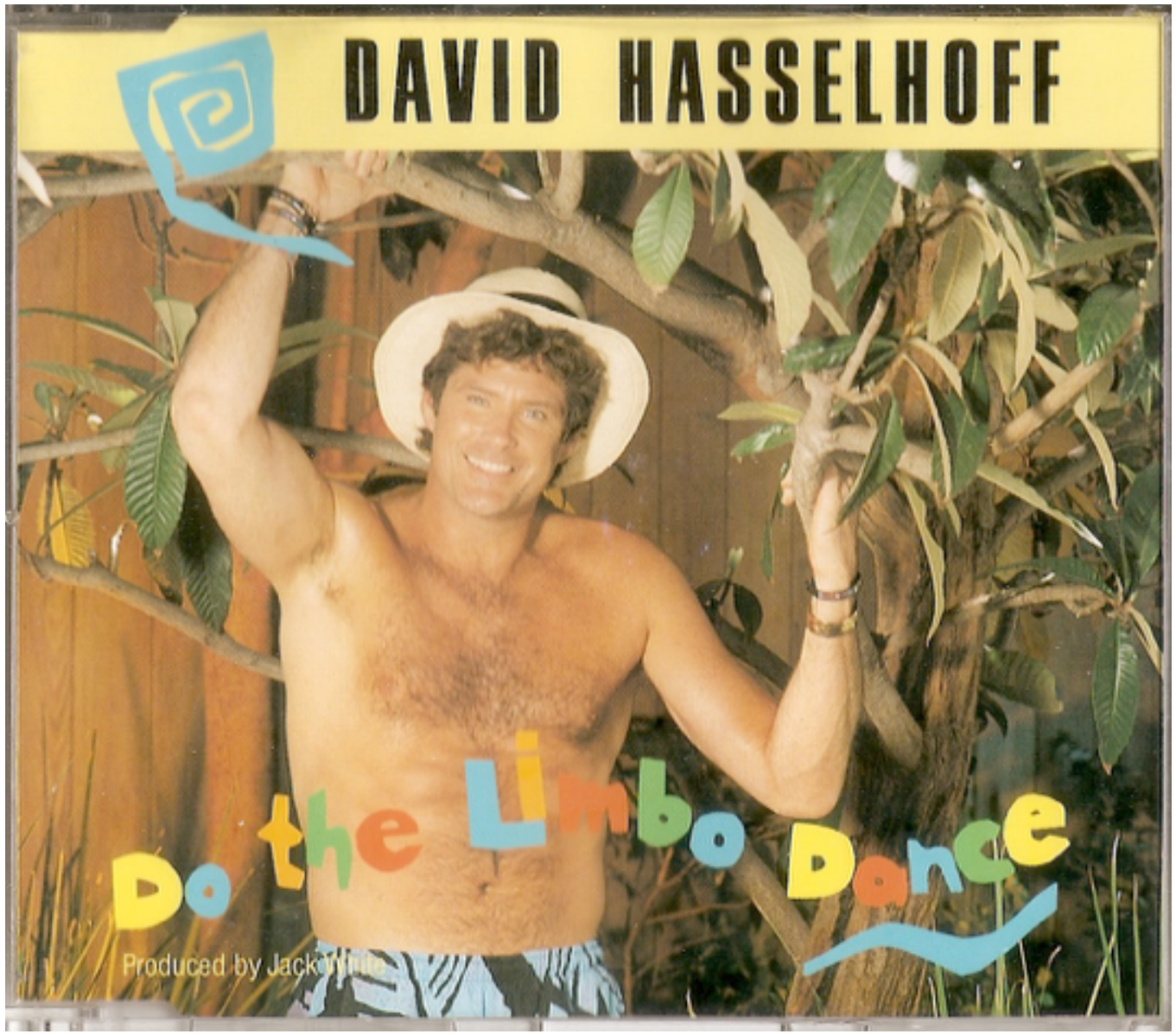
Programming should be more about the **what**


wat

State is dangerous

```
var clicks = 0;
document.addEventListener('click', function register(e) {
  if (clicks < 10) {
    if (e.clientX > innerWidth / 2 && isAPressed) {
      console.log(e.clientX, e.clientY);
      clicks += 1;
    }
  } else {
    document.removeEventListener('click', register);
  }
});

var isAPressed = false;
document.addEventListener('keydown', e => {
  isAPressed = e.keyCode === 65;
}, false);

document.addEventListener('keyup', e => {
  isAPressed = false;
}, false);
```

Event limbo



```
var clicks = 0;
document.addEventListener('click', function register(e) {
  if (clicks < 10) {
    if (e.clientX > innerWidth / 2) {
      console.log(e.clientX, e.clientY);
      clicks += 1;
    }
  } else {
    document.removeEventListener('click', register);
  }
});
```

Isn’t that the problem promises try to solve?

Sync

```
var y = f(x);
var z = g(y);
```

Promises

```
fAsync(x).then(...);
gAsync(x).then(...);
```

Sync

```
res =
  stocks
  .filter(q => q.symbol == 'FB')
  .map(q => q.quote)

res.forEach(x =>
  ...
```

RxJS

```
res =
  stocks //async retrieval
  .filter(q => q.symbol == 'FB')
  .map(q => q.quote)

res.subscribe(x =>
  ...
```


Click!
Click!
Click!

[
Click!
, 
Click!
, 
Click!
]


λ
λ
THE FUNCTIONAL WAY IS THE RIGHT WAY

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
  .filter(n => n % 2)
  .map(n => 'item ' + n)
  .forEach(n => console.log(n))

// "item 1"
// "item 3"
// "item 5"
// "item 7"
// "item 9"
```

Final Resting Place

Fantasy Role Playing

Functional Reactive Programming

	A	B	C
1			
2	12	20	=A2+B2
3			

	A	B	C
1			
2	10	20	30
3			

Deal with values that change **over time**

RxJS helps us compose **asynchronous** and event-based programs

Limit to 10

Filter

```
var clicks = 0;
document.addEventListener('click', function register(e) {
  if (clicks < 10) {
    if (e.clientX > innerWidth / 2) {
      console.log(e.clientX, e.clientY);
      clicks += 1;
    }
  } else {
    document.removeEventListener('click', register);
  }
});
```

Print the coordinates

```
fromEvent(document, 'click')
  .filter(c => c.clientX > innerWidth / 2 })
  .take(10)
  .subscribe(c => console.log(c.clientX, c.clientY) })
```

Create Observable

```
fromEvent(document, 'click')
  .filter(c => c.clientX > innerWidth / 2 })
  .take(10)
  .subscribe(c => console.log(c.clientX, c.clientY) })
```

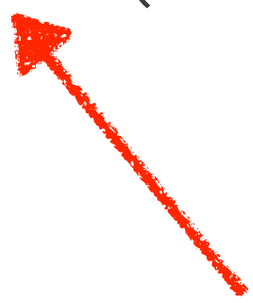
Create filtered Observable
from the first one

```
fromEvent(document, 'click')
  .filter(c => c.clientX > innerWidth / 2 })
  .take(10)
  .subscribe(c => console.log(c.clientX, c.clientY) })
```

Create final Observable
taking only first 10 results

```
fromEvent(document, 'click')
  .filter(c => c.clientX > innerWidth / 2 })
  .take(10)
  .subscribe(c => console.log(c.clientX, c.clientY) })
```

```
fromEvent(document, 'click')
  .filter(c => c.clientX > innerWidth / 2 })
  .take(10)
.subscribe(c => console.log(c.clientX, c.clientY) })
```



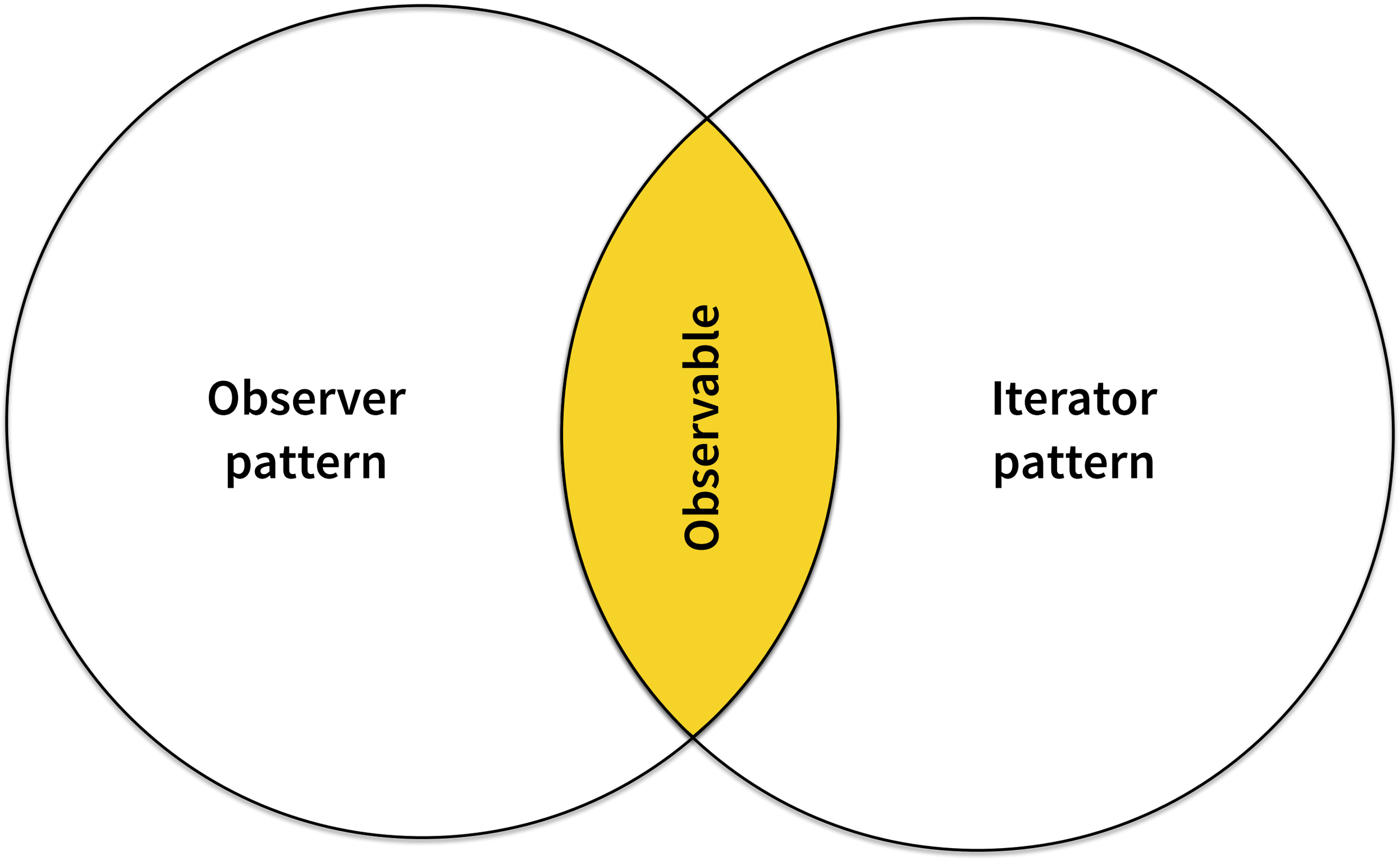
Actually kick off computation

Rx.Observable

Rx.Observer

Rx.Observer

- OnNext()
- OnError()
- OnComplete()


Observer pattern
Observable
Iterator pattern

```
// Creates an observable sequence of 5 integers
var source = Rx.Observable.range(1, 5)

// Prints out each item
var subscription = source.subscribe(
    x => { console.log('onNext: ' + x) },
    e => { console.log('onError: ' + e.message) },
    () => { console.log('onCompleted') })

// => onNext: 1
// => onNext: 2
// => onNext: 3
// => onNext: 4
// => onNext: 5
// => onCompleted
```

```
var mousemove = fromEvent(document, 'mousemove');

var mouseCoords = mousemove.map(e => ({
  left: e.clientX,
  top: e.clientY
}))
```

```
var mouseSide = mousemove.map(e =>
  (e.clientX > window.innerWidth / 2 ? 'right' : 'left'))

mouseCoords.subscribe(pos =>
  coords.innerHTML = pos.top + 'px ' + pos.left + 'px')

mouseSide.subscribe(s => side.innerHTML = s);
```

```
// Search Wikipedia for a given term
function searchWikipedia(term) {
  var cleanTerm = global.encodeURIComponent(term);
  var url = ‘http://en.wikipedia.org/w/api.php...'
    + cleanTerm + '&callback=JSONPCallback';
  return Rx.Observable.getJSONPRequest(url);
}
```

```
var input = document.querySelector('#searchtext'),
  results = document.querySelector('#results');

// Get all distinct key up events from the input and
var keyup = fromEvent(input, 'keyup')
  .map(e => e.target.value)
  .where(text => text.length > 2) // Longer than 2 chars
  .throttle(200) // Pause for 200ms
  .distinctUntilChanged(); // Only if the value has changed
```

```
var searcher = keyup
  .map(text => searchWikipedia(text)) // Search wikipedia
  .switchLatest() // Ensure no out of order results
  .where(data => (data.length === 2)); // Where we have data

searcher.subscribe(data => {
  // Append the results (data[1])
}, error => {
  // Handle any errors
});
```

Start Typing

```
// Search Wikipedia for a given term
function searchWikipedia(term) {
  var cleanTerm = global.encodeURIComponent(term);
  var url = 'http://en.wikipedia.org/w/api.php?
action=opensearch&format=json&search='
    + cleanTerm + '&callback=JSONPCallback';
  return Rx.Observable.getJSONPRequest(url);
}
```

```
var input = document.querySelector('#searchtext'),
  results = document.querySelector('#results');

// Get all distinct key up events from the input and
var keyup = fromEvent(input, 'keyup')
  .map(e => e.target.value)
  .where(text => text.length > 2) // Longer than 2 chars
  .throttle(200) // Pause for 200ms
  .distinctUntilChanged(); // Only if the value has changed
```

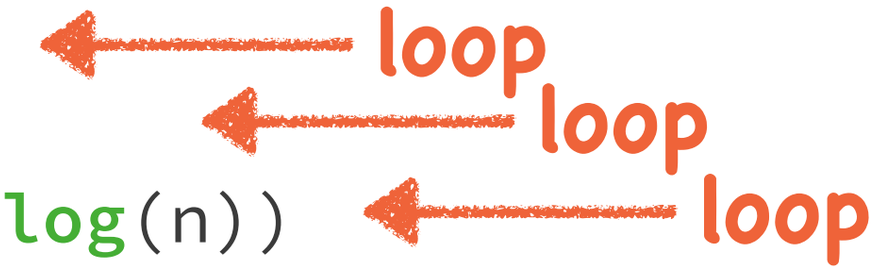
```
// Search Wikipedia for a given term
function searchWikipedia(term) {
  return fromArray(['JavaScript',
    'JavaServer Pages',
    'JavaSoft',
    'JavaScript library',
    'JavaScript Object Notation',
    'JavaScript engine',
    'JavaScriptCore']);
}

var input = document.querySelector('#searchtext'),
  results = document.querySelector('#results');

// Get all distinct key up events from the input and
var keyup = fromEvent(input, 'keyup')
  .map(e => e.target.value)
  .where(text => text.length > 2) // Longer than 2 chars
  .throttle(200) // Pause for 200ms
  .distinctUntilChanged(); // Only if the value has changed
```

fromArray
fromCallback
fromEvent
fromEventPattern
fromIterable
fromNodeCallback
fromPromise

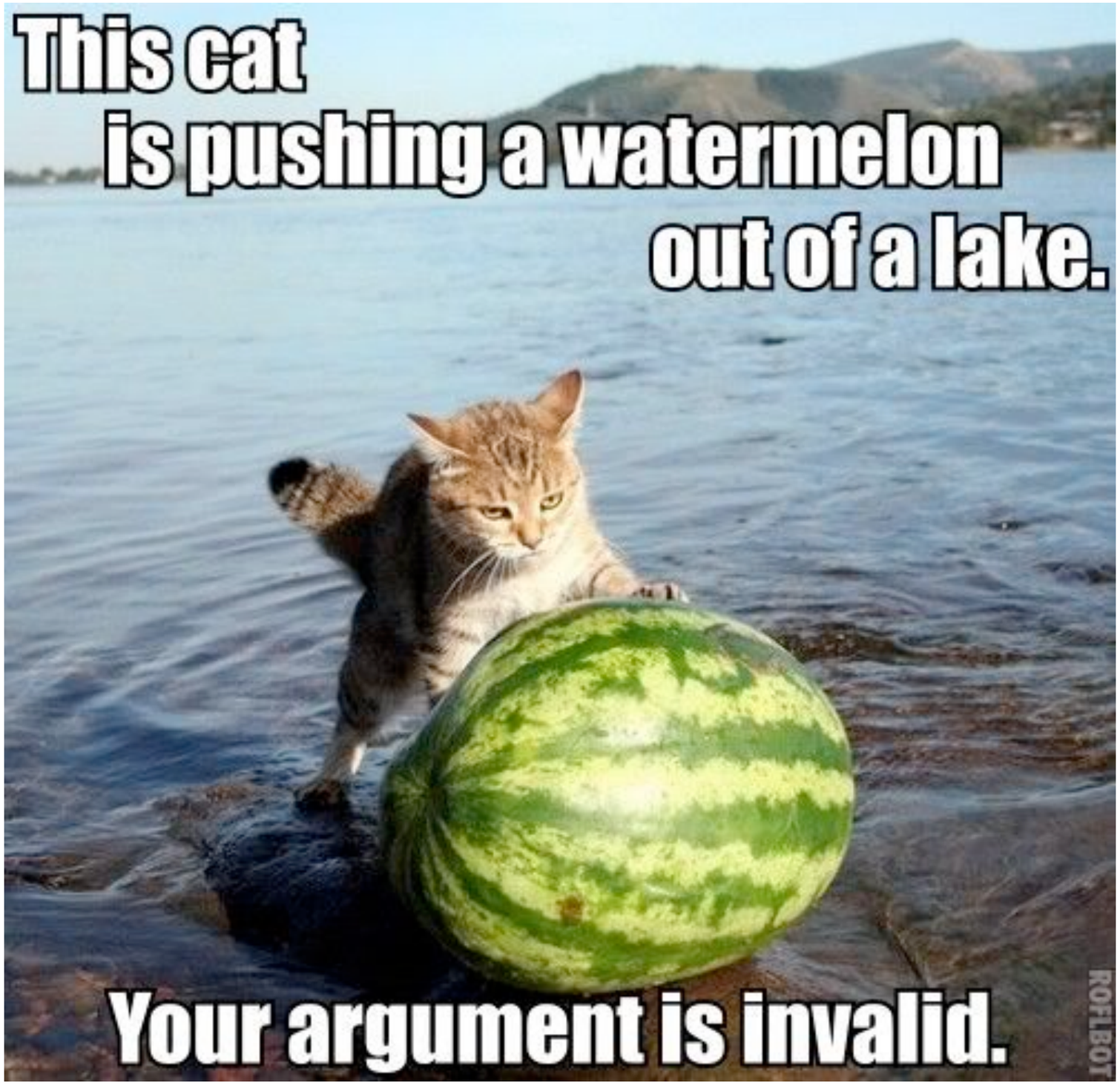
```
var mouseup = fromEvent(dragTarget, 'mouseup');
var mousemove = fromEvent(document, 'mousemove');
var mousedown = fromEvent(dragTarget, 'mousedown');

var mousedrag = mousedown.selectMany(md => {
    var startX = md.clientX + window.scrollX,
        startY = md.clientY + window.scrollY,
        startLeft = parseInt(md.target.style.left, 10) || 0,
        startTop = parseInt(md.target.style.top, 10) || 0;

    // Calculate delta with mousemove until mouseup
    return mousemove.map(mm => {
        mm.preventDefault();

        return {
            left: startLeft + mm.clientX - startX,
            top: startTop + mm.clientY - startY
        };
    }).takeUntil(mouseup);
});

subscription = mousedrag.subscribe(pos => {
    dragTarget.style.top = pos.top + 'px';
    dragTarget.style.left = pos.left + 'px';
});
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
  .filter(n => n % 2)
  .map(n => 'item ' + n)
  .forEach(n => console.log(n))

// "item 1"
// "item 3"
// "item 5"
// "item 7"
// "item 9"
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
  .filter(n => n % 2)
  .map(n => 'item ' + n)
  .forEach(n => console.log(n))
```

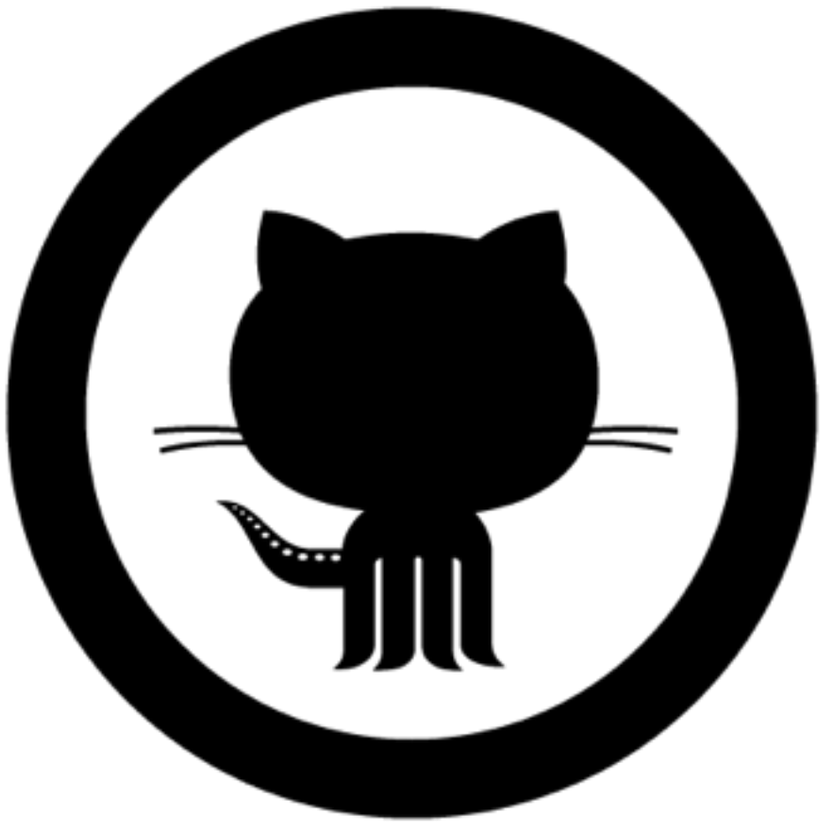



```
// "item 1"
// "item 3"
// "item 5"
// "item 7"
// "item 9"
```


This cat
is pushing a watermelon
out of a lake.
Your argument is invalid.
ROFLBOT

```
fromArray([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
  .filter(n => n % 2)
.map(n => n * 100)
.map(n => 'item ' + n)
.subscribe(n => console.log(n))
```

First Class events
Composability
Encapsulation




NETFLIX


Microsoft


SOUNDCLOUD

Thanks!

@sergimansilla

goto; conference

Please evaluate this talk via the mobile app!

Follow us @gotoamst