

Alex Munda

Twitter: @mundaalex

GitHub: alexmunda

alexmunda.com

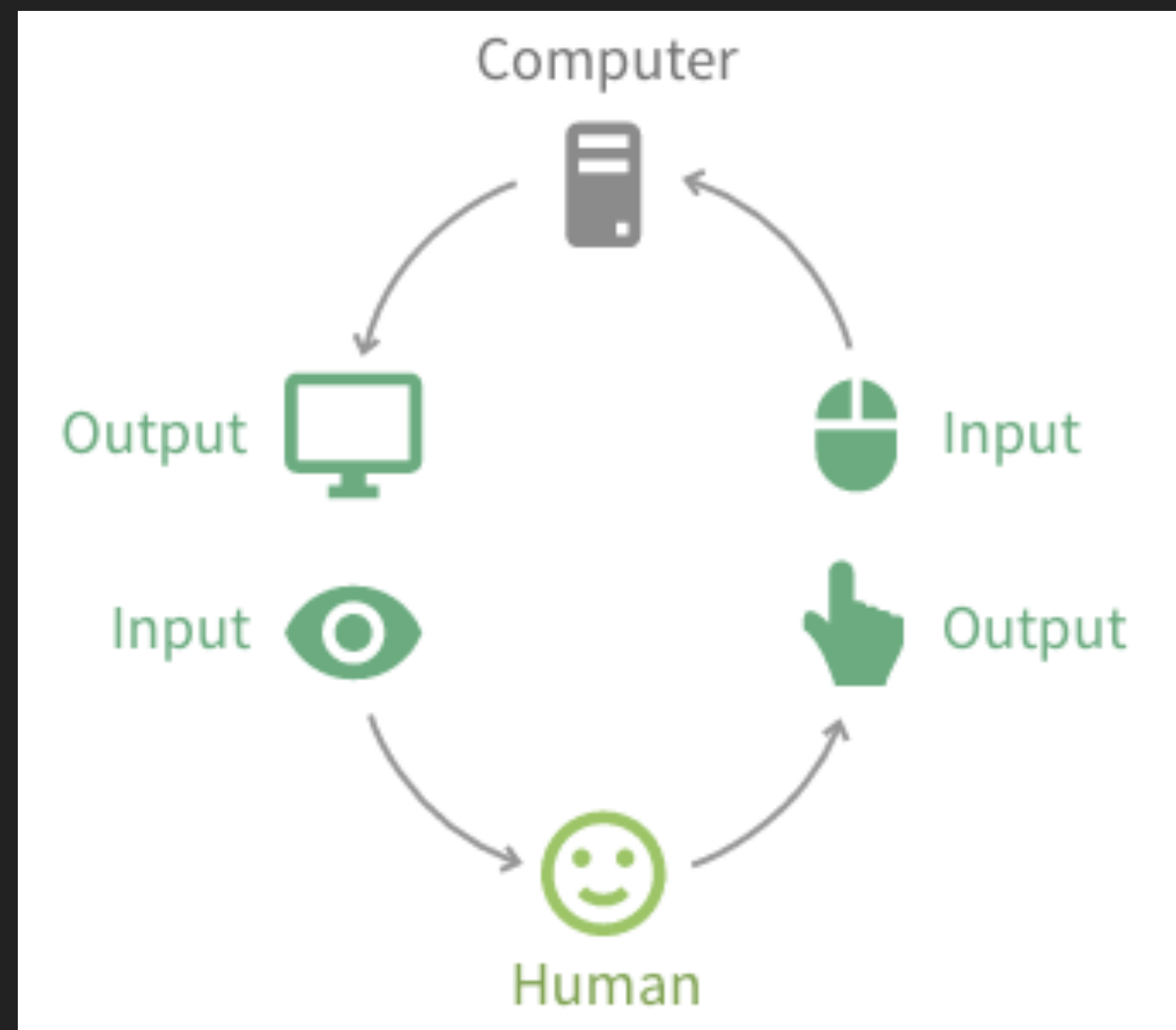


All images and
quotations from
<http://cycle.js.org/>
unless otherwise
noted.

INTRO TO CYCLE.JS

HUMAN COMPUTER INTERACTION CYCLE

- ▶ Think of both the Human and Computer as functions



OBSERVABLES

- ▶ “Lazy event streams which can emit zero or more events, and may or may not finish”
- ▶ Observables can be listened to

```
1  var royals = ['Moose', 'Hosmer', 'Salvy', 'Gordo'];
2
3  var royals$ = Rx.Observable.from(royals);
4
5  royals$.subscribe(player => console.log(player));
6
7  // Outputs:
8  // Moose
9  // Hosmer
10 // Salvy
11 // Gordo
```

ANATOMY OF A CYCLE.JS APPLICATION

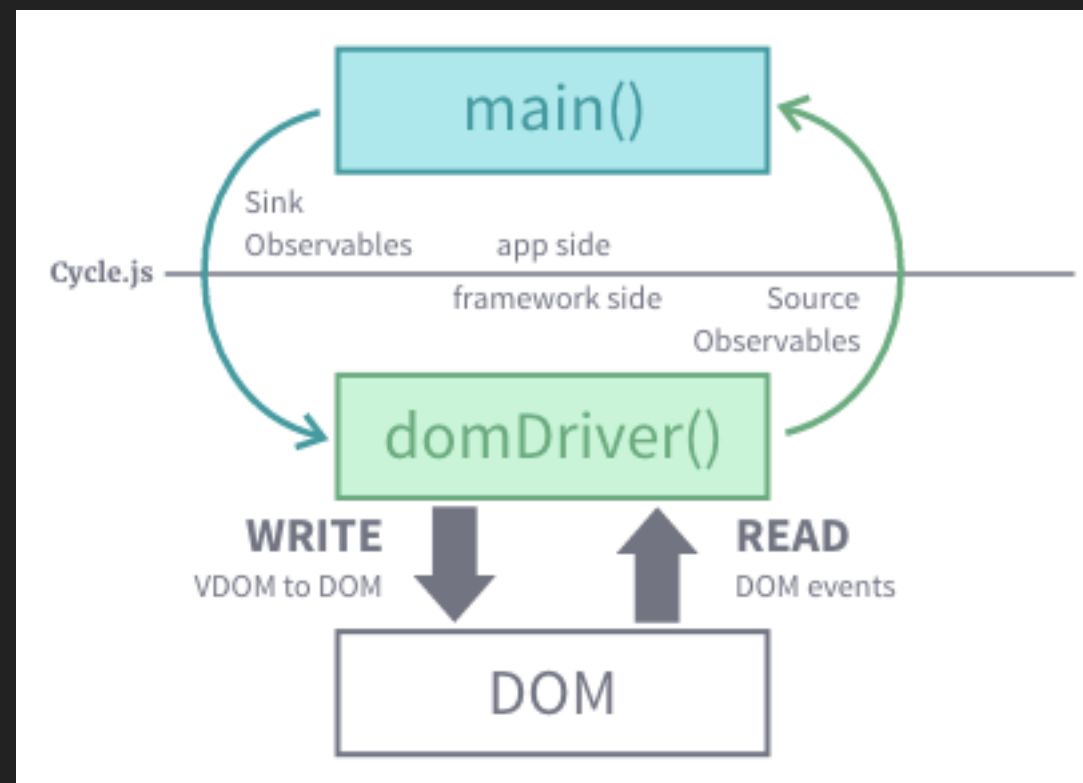
```
1  function main(sources){ // Takes in sources (observables), returns sinks (observables)
2
3  const sinks = {
4    DOM: sources.DOM.select('.my-cool-button')
5      .events('click')
6      // Do more cool things to the DOM with RxJS observable operators
7      // Ex: .filter(...).map(...)
8  }
9
10 return sinks;
11 // Sink and driver object keys must match so Cycle.run can match up the sources
12 // and sink observables.
13 }
14
15 const drivers = { // Listen to Observable sinks and do cool things
16   DOM: makeDOMDriver('#app')
17 };
18
19 Cycle.run(main, drivers); //Connects main and drivers circularly
```

SOURCES AND SINKS

```
1  function main(sources){
2    //An HTTP request observable from a button click
3    const getGameDetails$ = sources.DOM.select('.get-game-details').events('click')
4      .map(() => {
5        return {
6          url: STATS_URL,
7          method: 'GET'
8        };
9      });
10
11   //HTTP responses are sources
12   const gameDetails$ = sources.HTTP
13     .filter(res$ => res$.request.url.indexOf(STATS_URL) === 0)
14     .mergeAll()
15     .map(res => res.body)
16     .startWith(null)
17
18   const sinks = {
19     // DOM: $gameDetails.map(...)
20     HTTP: getGameDetails$ // HTTP requests are sinks
21   }
22
23   return sinks;
24 }
25
26 const drivers = {
27   //DOM: makeDOMDriver('#app'),
28   HTTP: makeHTTPDriver()
29 };
30
31 Cycle.run(main, drivers);
```

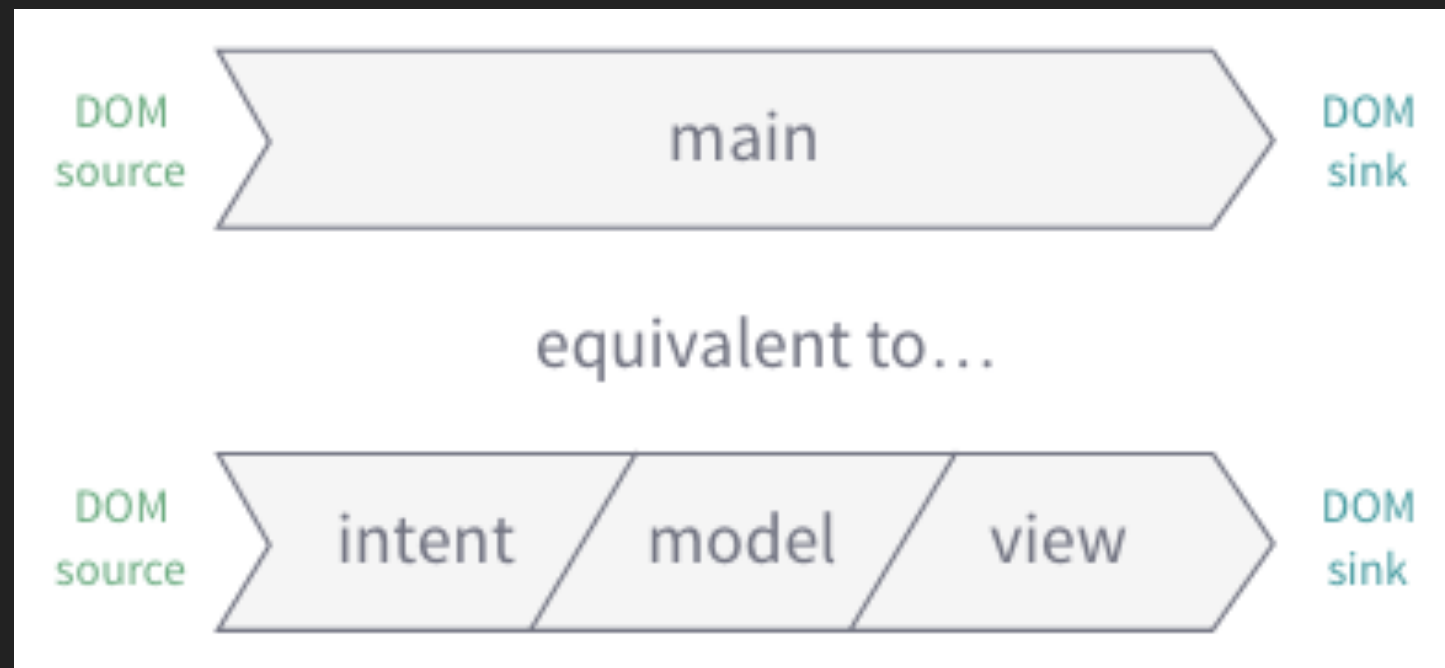
DRIVERS

- ▶ DOM Driver
- ▶ Other drivers: HTTP Driver, React Native Driver
- ▶ The DOM sink is a virtual-dom observable



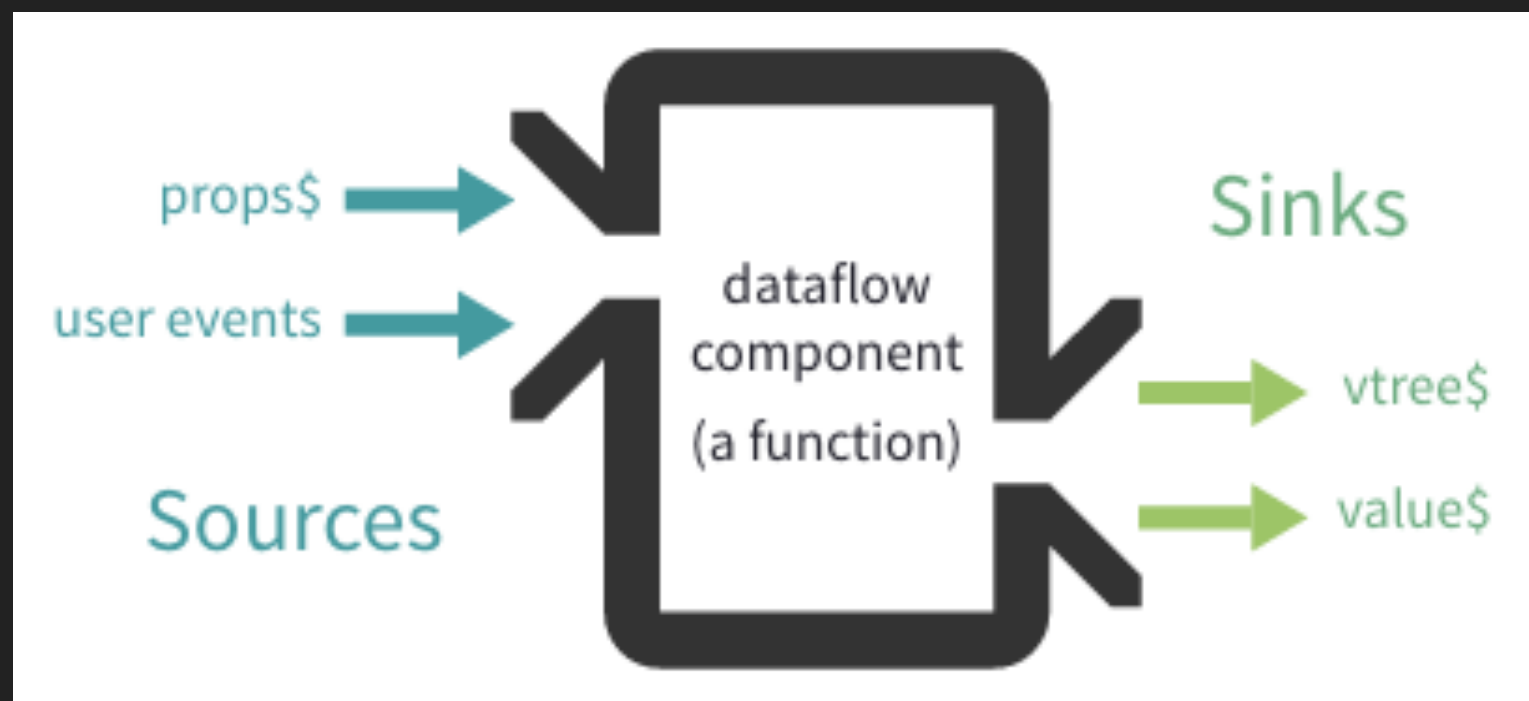
MODEL - VIEW - INTENT

- ▶ Intent - listen to user inputs
- ▶ Model - business logic / state
- ▶ View - show output to user



COMPONENTS

- ▶ Any "main" function can be used as a component in another "main" function
- ▶ Call `isolate(component)()` on component functions that will be used multiple times



DEMO

► <http://alexmunda.com/royals5>

LEARN MORE ABOUT CYCLE.JS

- ▶ Cycle.js Site

- ▶ <http://cycle.js.org/getting-started.html>

- ▶ Cycle.js Repos

- ▶ <https://github.com/cyclejs>

- ▶ Cycle.js Fundamentals

- ▶ <https://egghead.io/series/cycle-js-fundamentals>

LEARN MORE ABOUT RXJS

- ▶ Learn RxJS

- ▶ <http://reactivex.io/learnrx/>

- ▶ The introduction to Reactive Programming you've been missing

- ▶ <https://gist.github.com/staltz/868e7e9bc2a7b8c1f754>

- ▶ Async JavaScript with Reactive Extensions

- ▶ <https://www.youtube.com/watch?v=XRYN2xt11Ek>

- ▶ RxJS Repo Docs

- ▶ <https://github.com/Reactive-Extensions/RxJS/tree/master/doc>

- ▶ RxMarbles

- ▶ <http://rxmarbles.com/>