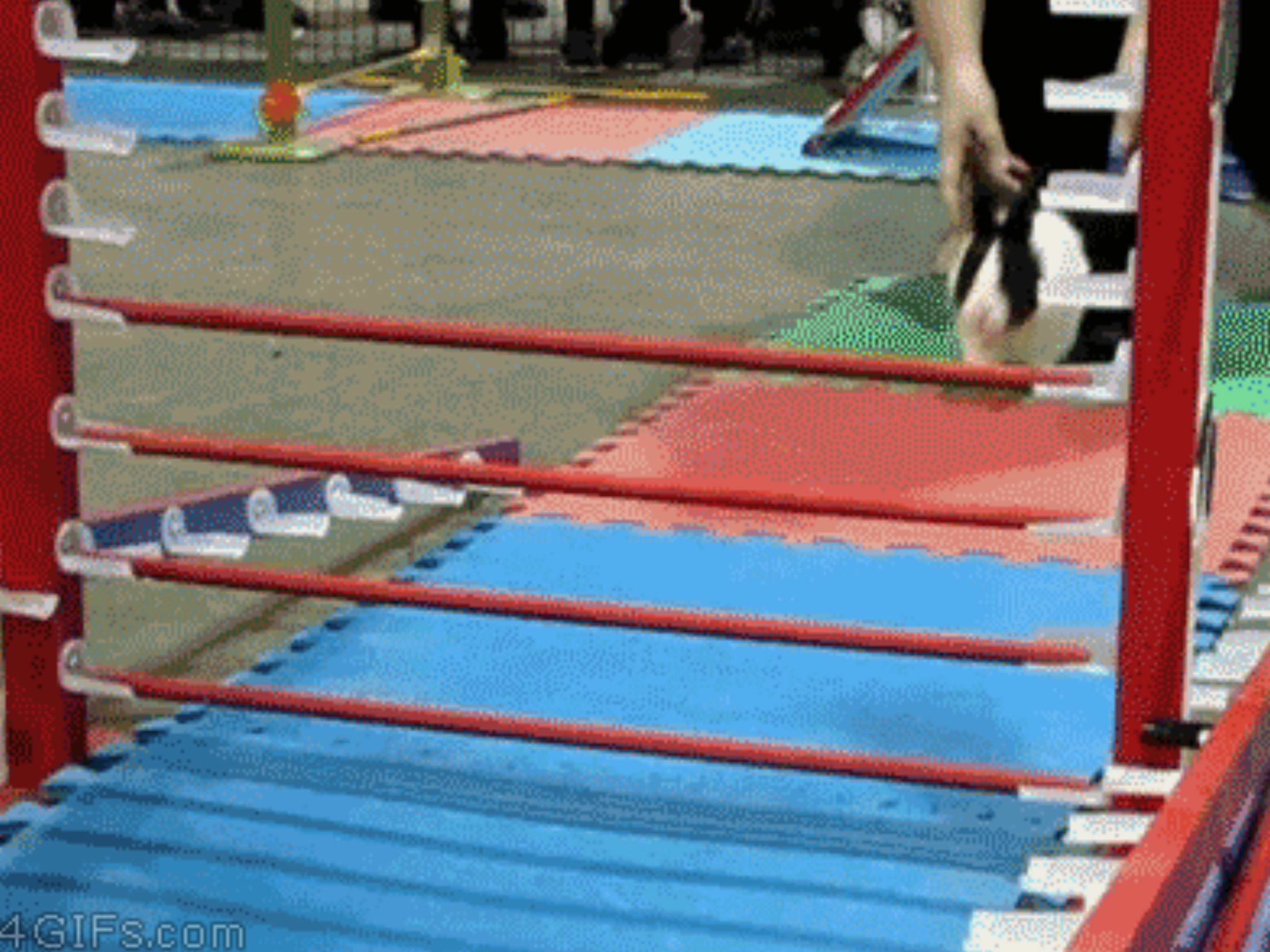




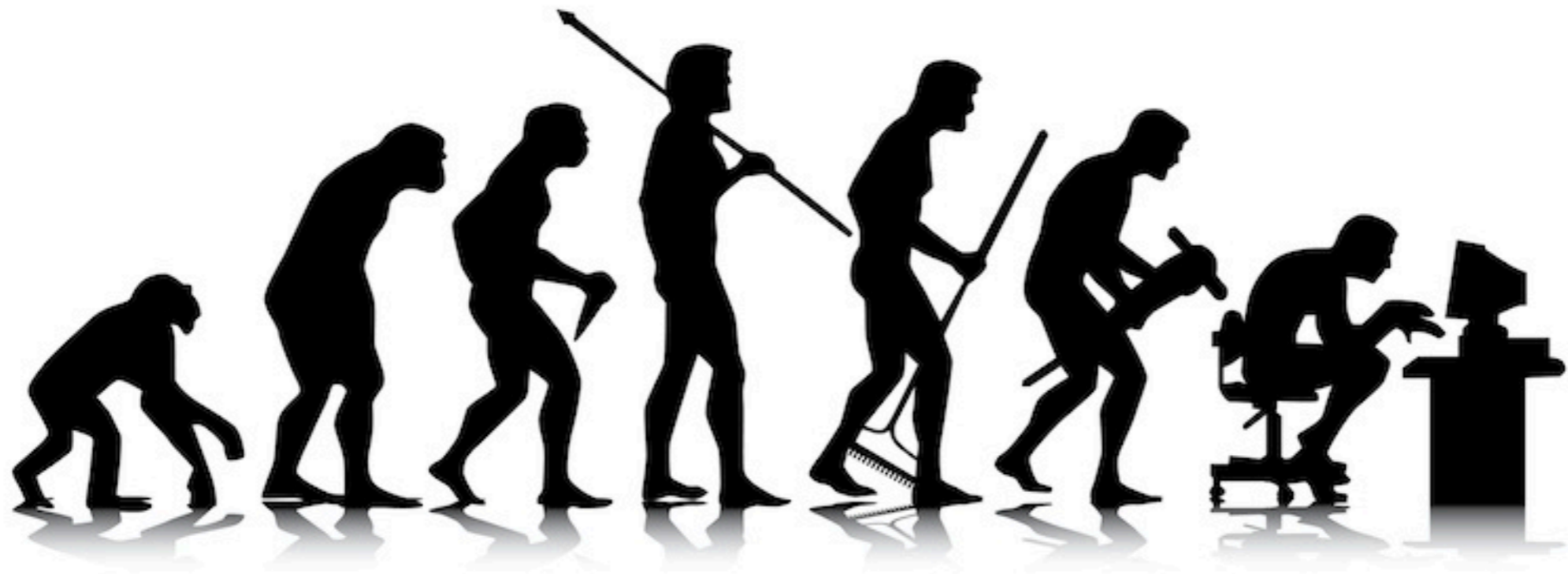
# Redux



# WHAT THE **HECK** IS REDUX? \* †

\* Not the same thing as React

† Doesn't have to be used with React



**DEMO**

**WHAT THE HECK  
IS REDUX?**

# A “state container”

A library that allows you to manage your front-end app's state at any scale.



# **The store**

# **Actions**

# **Reducers**



# The store

```
{  
  counter: 3,  
  dropdown: {  
    status: 'open',  
  },  
}
```

# Actions

```
{ type: INCREMENT }  
  { type: DECREMENT }  
{ type: TOGGLE_DROPDOWN, status: 'open' }
```

# Reducers

## Pure functions

1. when given the same input, always return the same output
2. cause no side effects

# Example: online orders

```
var orders = [10, 20, 30];

function addOrderToRunningTotal(currentRunningTotal, order) {
  return currentRunningTotal + order;
}

var runningTotal = orders.reduce(addOrderToRunningTotal, 0);

console.log(runningTotal); // => 60;
```

**DEMO**

# Example: online orders #2

```
1 var actions = [  
2   { type: 'PLACE_ORDER', orderTotal: 10 },  
3   { type: 'PLACE_ORDER', orderTotal: 20 },  
4   { type: 'PLACE_ORDER', orderTotal: 30 },  
5 ];  
6  
7 function addOrderToRunningTotal(currentRunningTotal, action) {  
8   return currentRunningTotal + action.orderTotal;  
9 }  
10  
11 var runningTotal = actions.reduce(addOrderToRunningTotal, 0);  
12  
13 console.log(runningTotal); // => 60;
```

**DEMO**

```
1 export class MyComponent implements OnInit {
2   private runningTotal: number;
3
4   ngOnInit() {
5     this.runningTotal = this.getRunningTotal(this.runningTotal, { type: '@@INIT' });
6   }
7
8   private getRunningTotal(currentRunningTotal = 0, action) {
9     if (action.type === 'PLACE_ORDER') {
10      return currentRunningTotal + action.orderTotal;
11    }
12    return currentRunningTotal;
13  }
14
15  private handlePlaceOrderButtonClicked(orderTotal: number): void {
16    this.runningTotal = this.getRunningTotal(this.runningTotal, {
17      type: 'PLACE_ORDER',
18      orderTotal: orderTotal,
19    });
20  }
21 }
```



```
1 <button (click)="handlePlaceOrderButtonClicked(10)">
2   Place $10 order
3 </button>
4
5 <button (click)="handlePlaceOrderButtonClicked(20)">
6   Place $20 order
7 </button>
8
9 <button (click)="handlePlaceOrderButtonClicked(30)">
10  Place $30 order
11 </button>
```

**DEMO**

# The store

```
1 import { createStore } from 'redux';
2 import { runningTotal } from './reducers'
3
4 const store = createStore(runningTotal);
5
6 console.log(store.getState()); // => 0
7
8 store.dispatch({ type: 'PLACE_ORDER', orderTotal: 10 });
9
10 store.subscribe(function() {
11   console.log(store.getState()); // => 10
12 });
```

```
1 export class MyModule {
2   constructor(
3     devToolsExtension: DevToolsExtension,
4     ngRedux: NgRedux<IAppState>
5   ) {
6     const enhancers = [];
7
8     if (devToolsExtension.isEnabled()) {
9       enhancers.push(devToolsExtension.enhancer());
10    }
11
12    ngRedux.configureStore(rootReducer, {}, [], enhancers);
13  }
14 }
```

```
1 import { combineReducers } from 'redux';
2
3 function runningTotal(currentRunningTotal: number = 0, action) {
4   if (action.type === 'PLACE_ORDER') {
5     return currentRunningTotal + action.orderTotal;
6   }
7   return currentRunningTotal;
8 }
9
10 export const rootReducer = combineReducers({
11   runningTotal,
12 })
store.getState(); // => 0
store.getState(); // => { runningTotal: 0 }
```

```
1 import { createStore } from 'redux';
2 import { NgRedux } from 'ng2-redux';
3
4 export class MyComponent {
5   private disconnect: Function;
6
7   constructor(private ngRedux: NgRedux<IAppState>) {
8     this.disconnect = ngRedux.connect(this.mapStateToProps, this.mapDispatchToProps)(this);
9   }
10
11  ngOnDestroy() {
12    this.disconnect();
13  }
14
15  private mapStateToProps(state: IAppState) {
16    const { runningTotal } = state;
17    return { runningTotal };
18  }
19
20  private mapDispatchToProps(dispatch) {
21    return {
22      handlePlaceOrderButtonClicked: function(orderTotal: number) {
23        dispatch({ type: 'PLACE_ORDER', orderTotal });
24      },
25    };
26  }
27 }
```

**DIEMO**

```
dispatch( { type: 'FINISH_TALK' } );
```