REACT JS

React is the **new cool technology** developers want to use to build their web apps

React was developed at **Facebook** and is used extensively on facebook.com

Instagram is entirely built on React

React is a very recent technology and interest in it has only grown as developers discover its utility

React was first launched in May, 2013

So.... what is React?

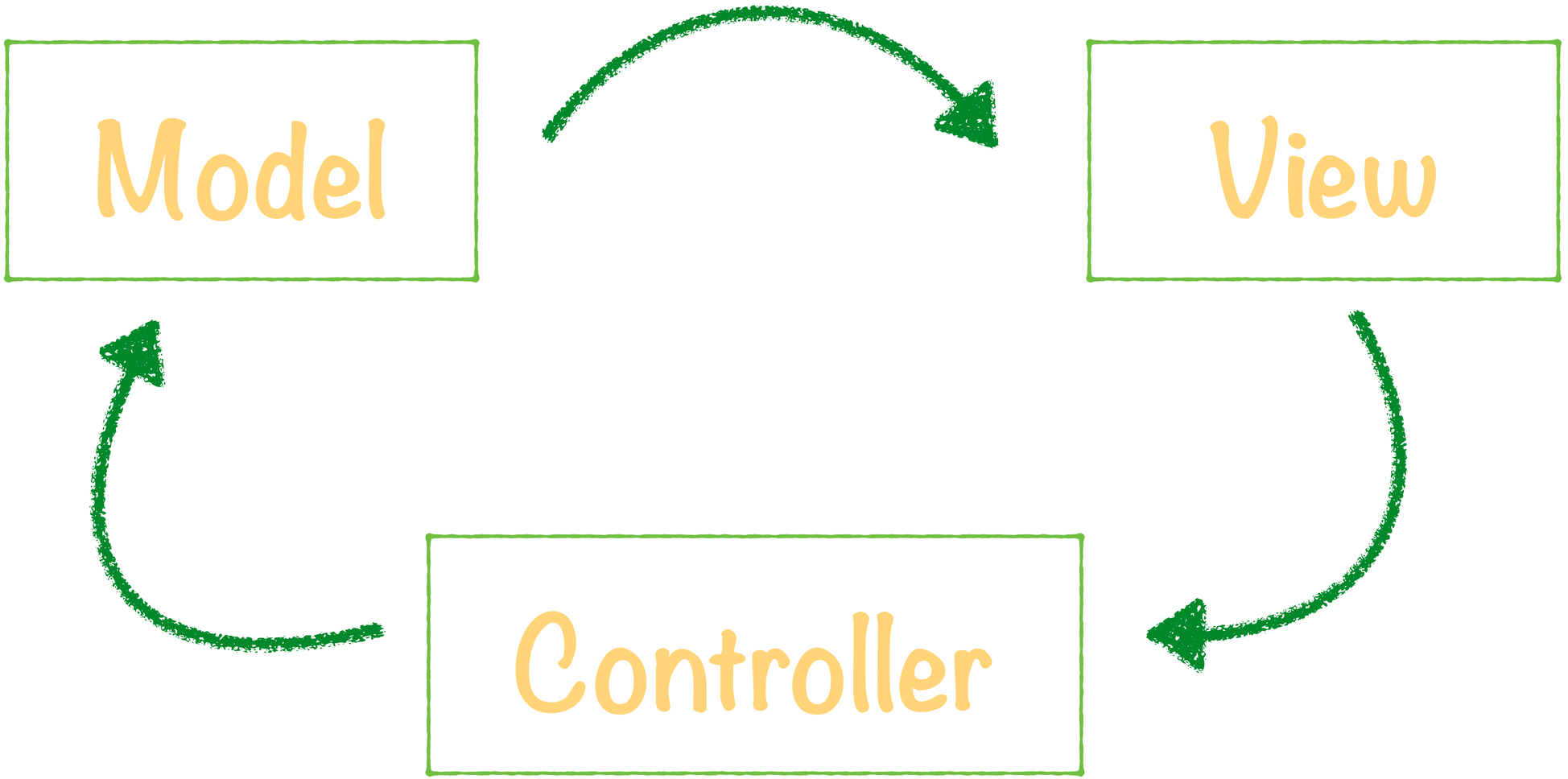
Let's start with what it is not - React is NOT a framework

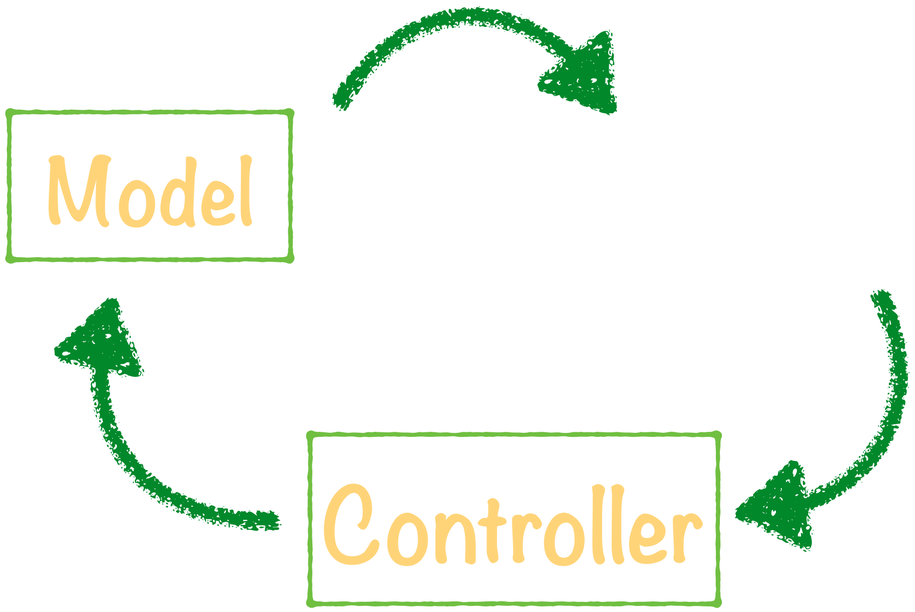
React is NOT a framework

React is a UI library

It allows you to build reusable, stateful and interactive components

Consider the MVC programming paradigm



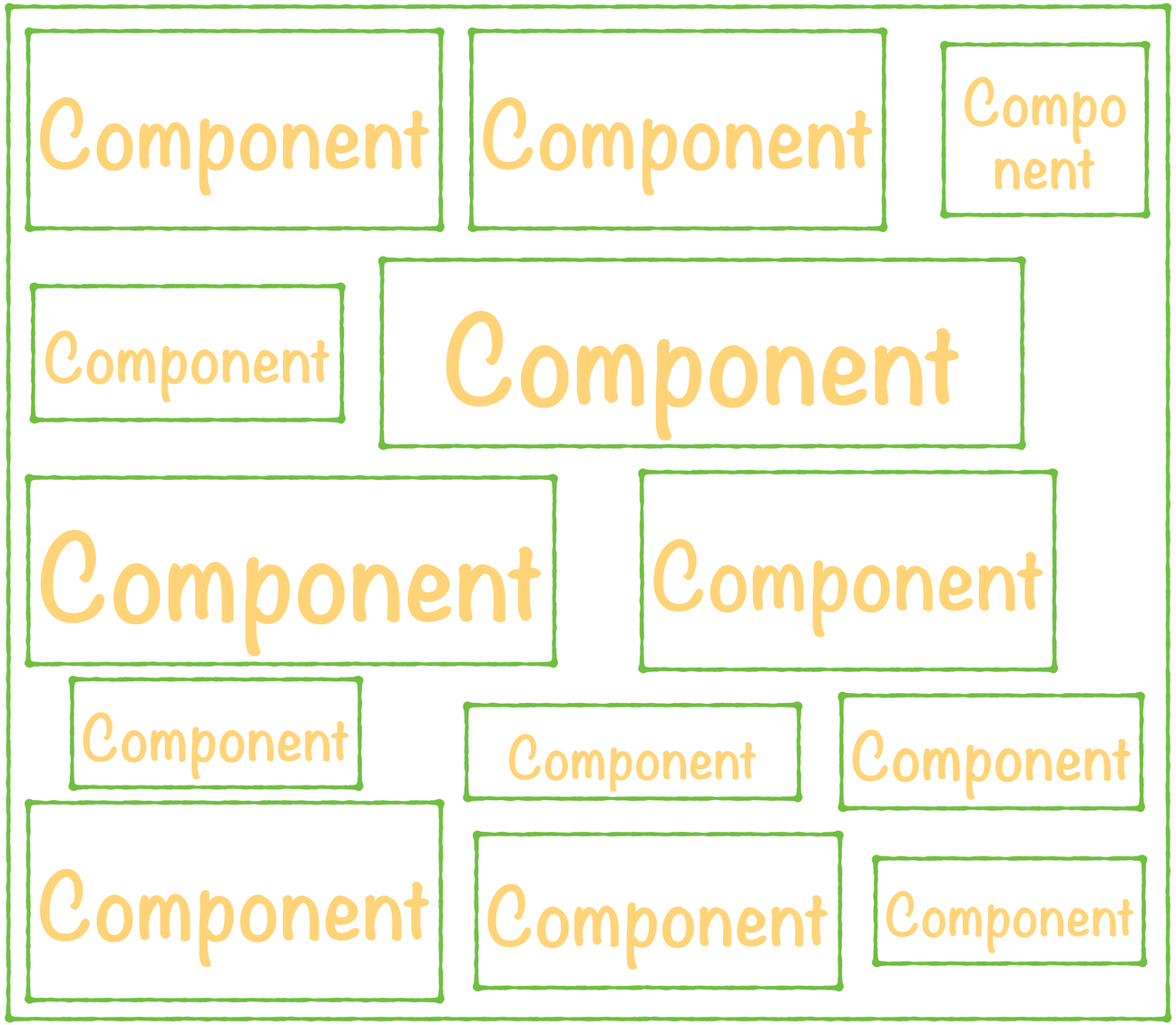


React focuses on the View!

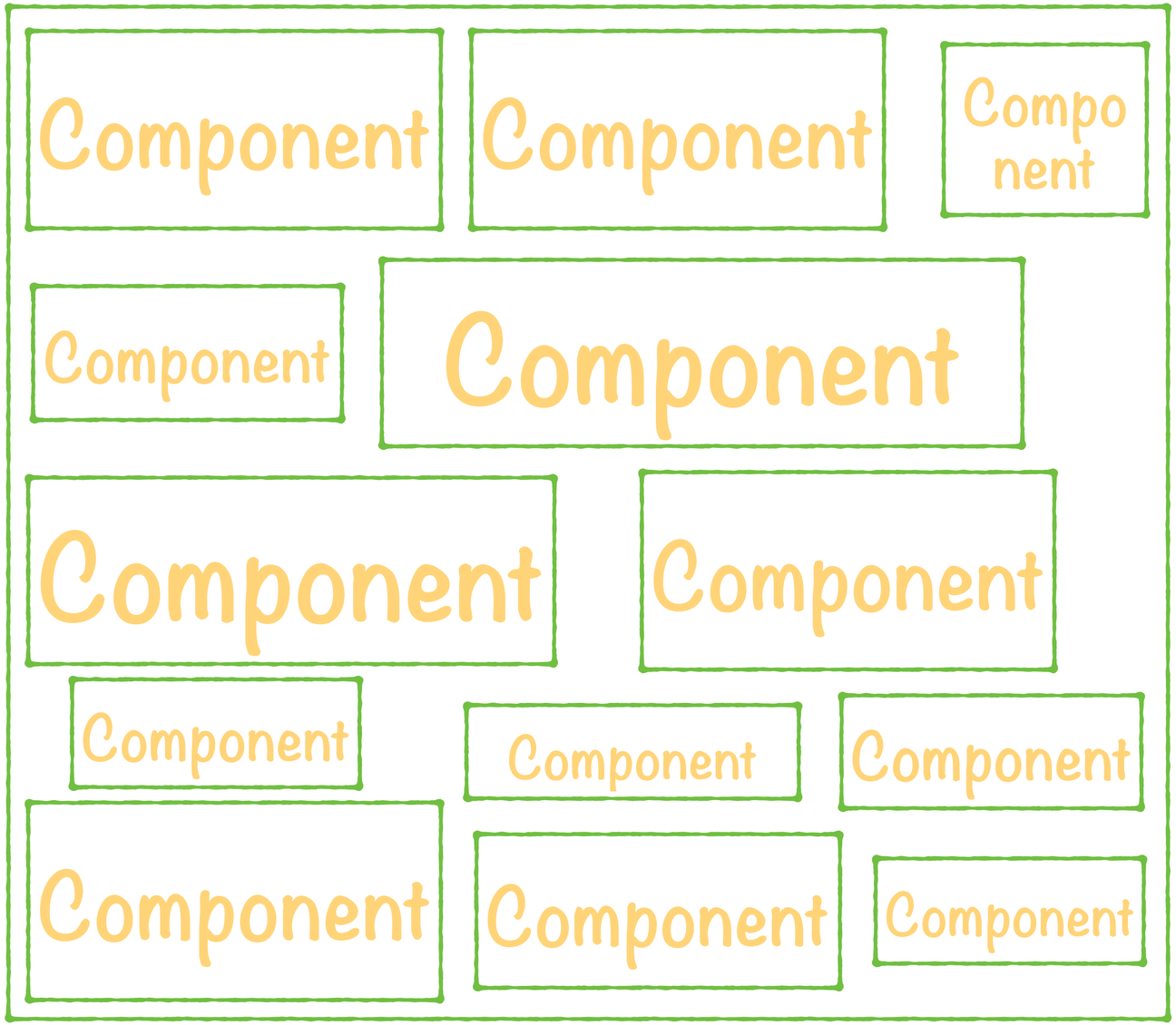
View

As you build UI,
break stuff down into
smaller logical units

And make each of
these units a
reusable component

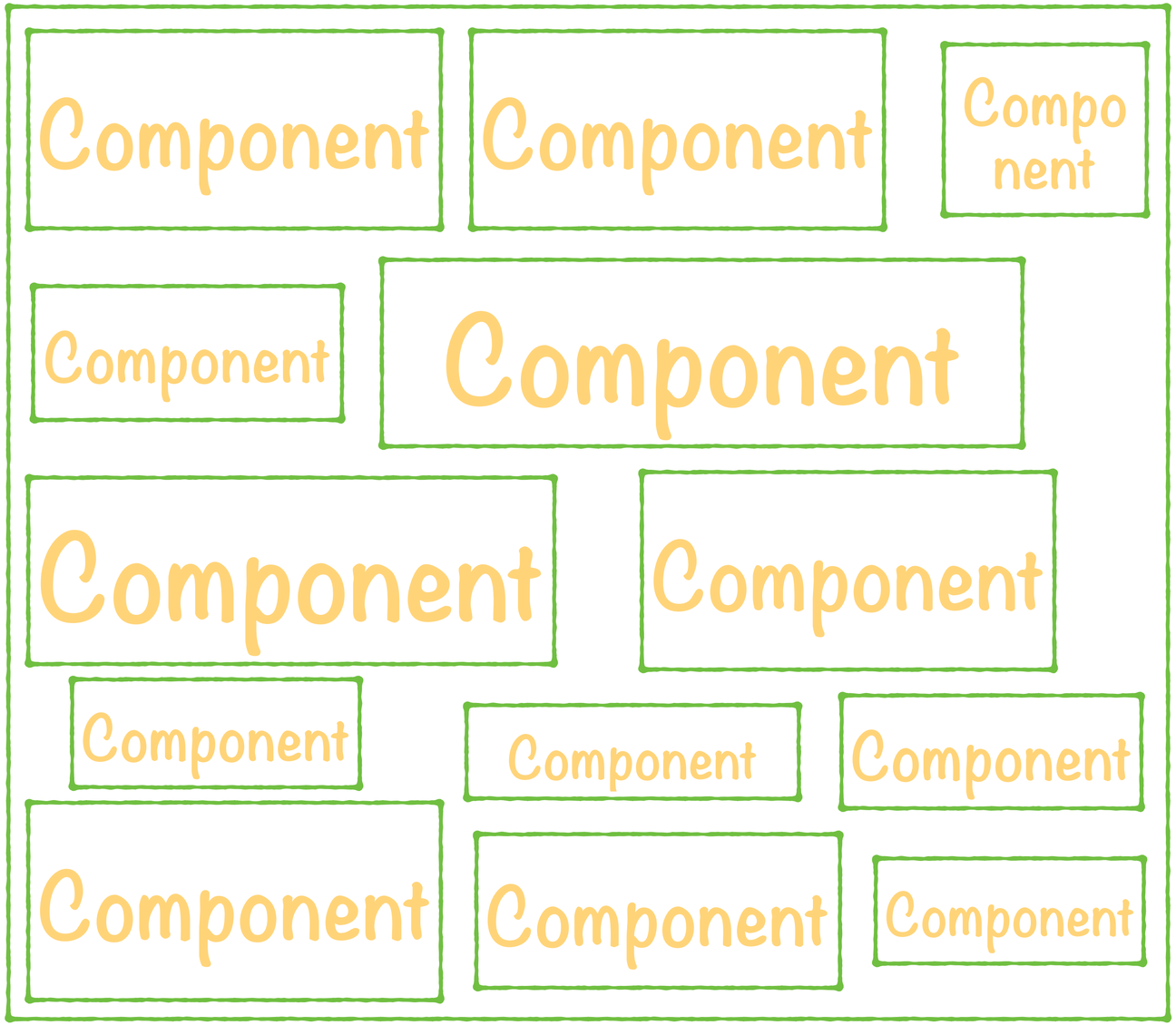


React imagines every view as made up of components

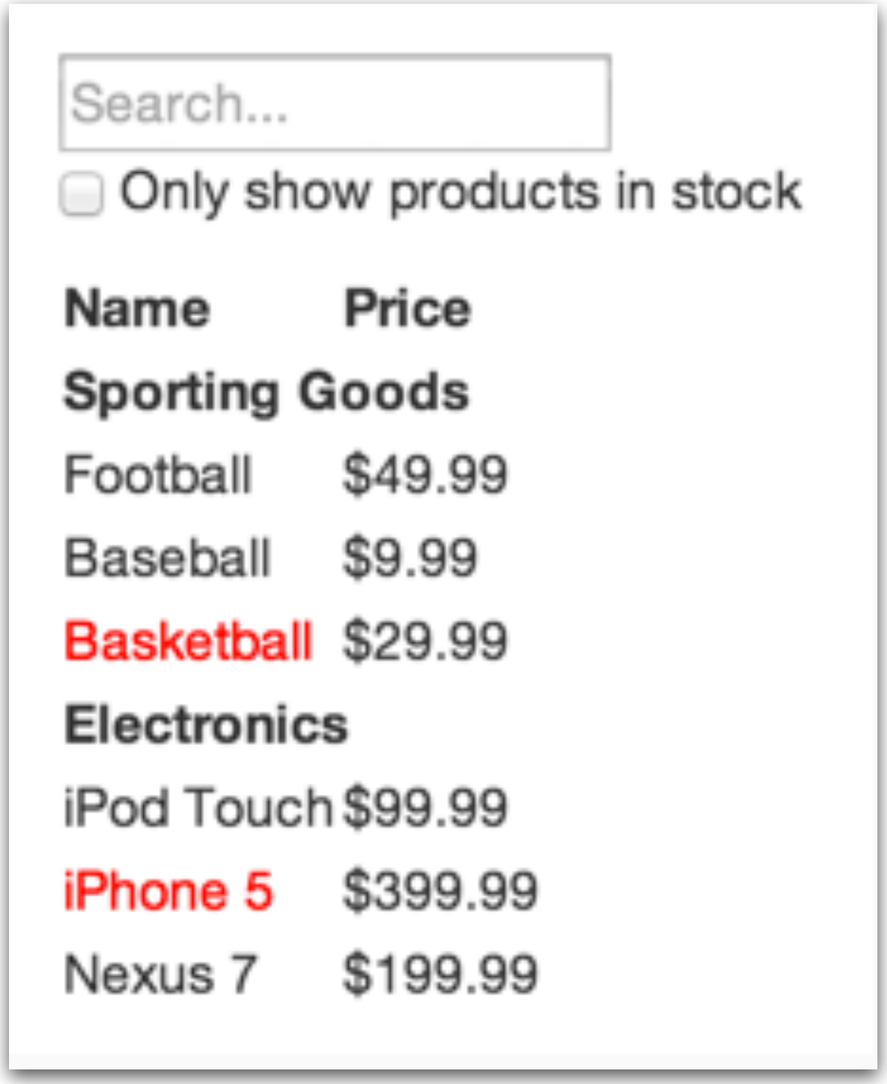


Components may **build** on other components

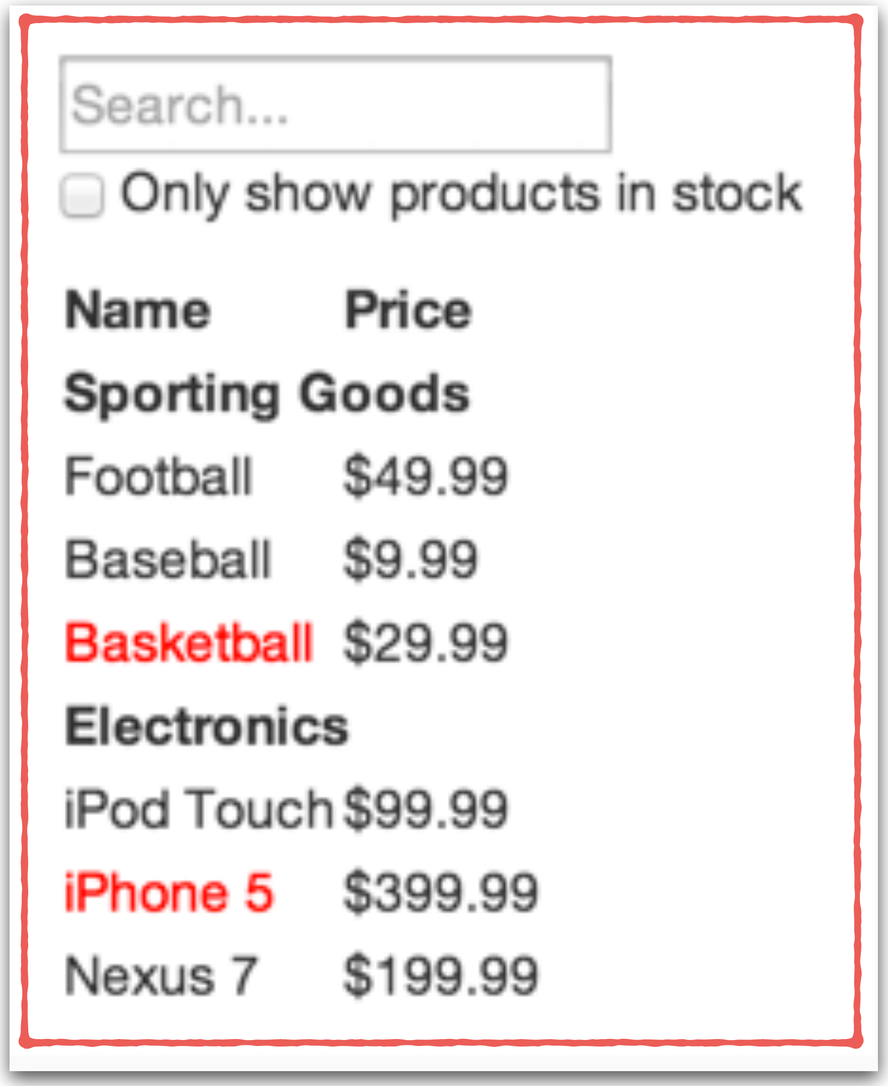
React is a form of **component driven development**



Let's break down this UI into a component hierarchy

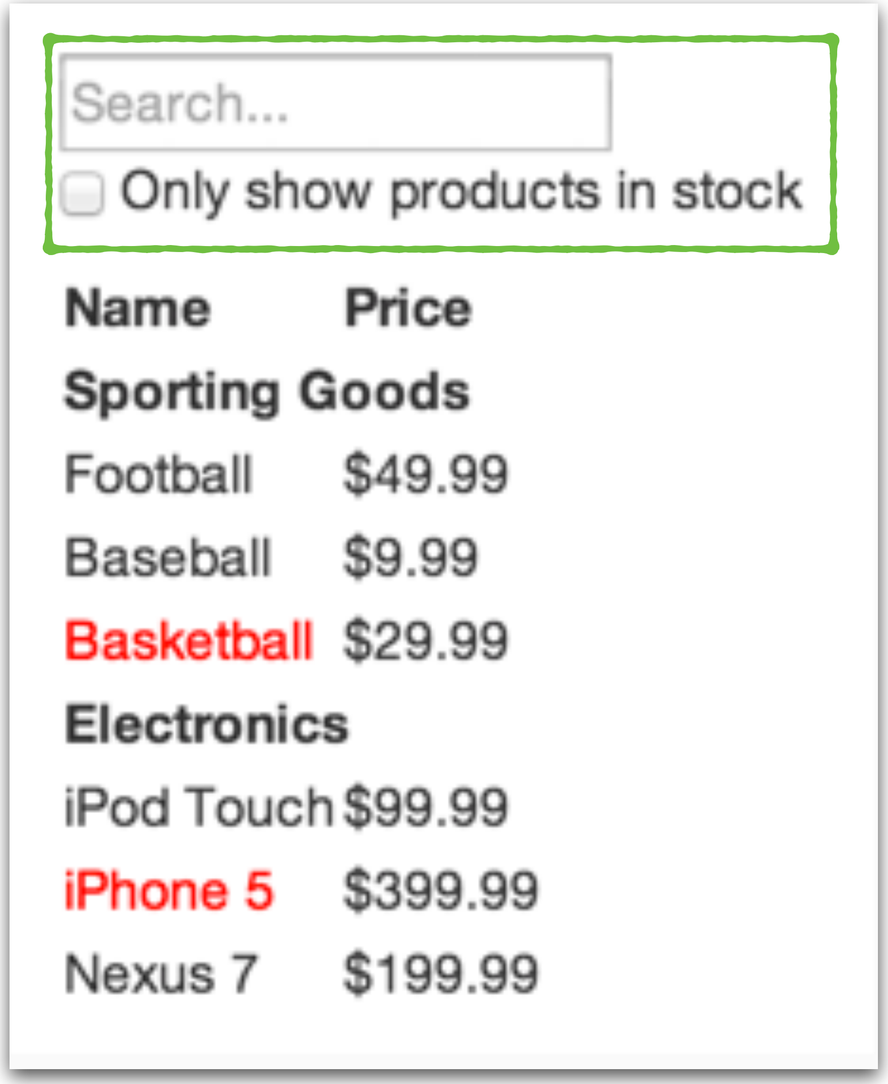


https://facebook.github.io/react/docs/thinking-in-react.html



FilteredProductsTable

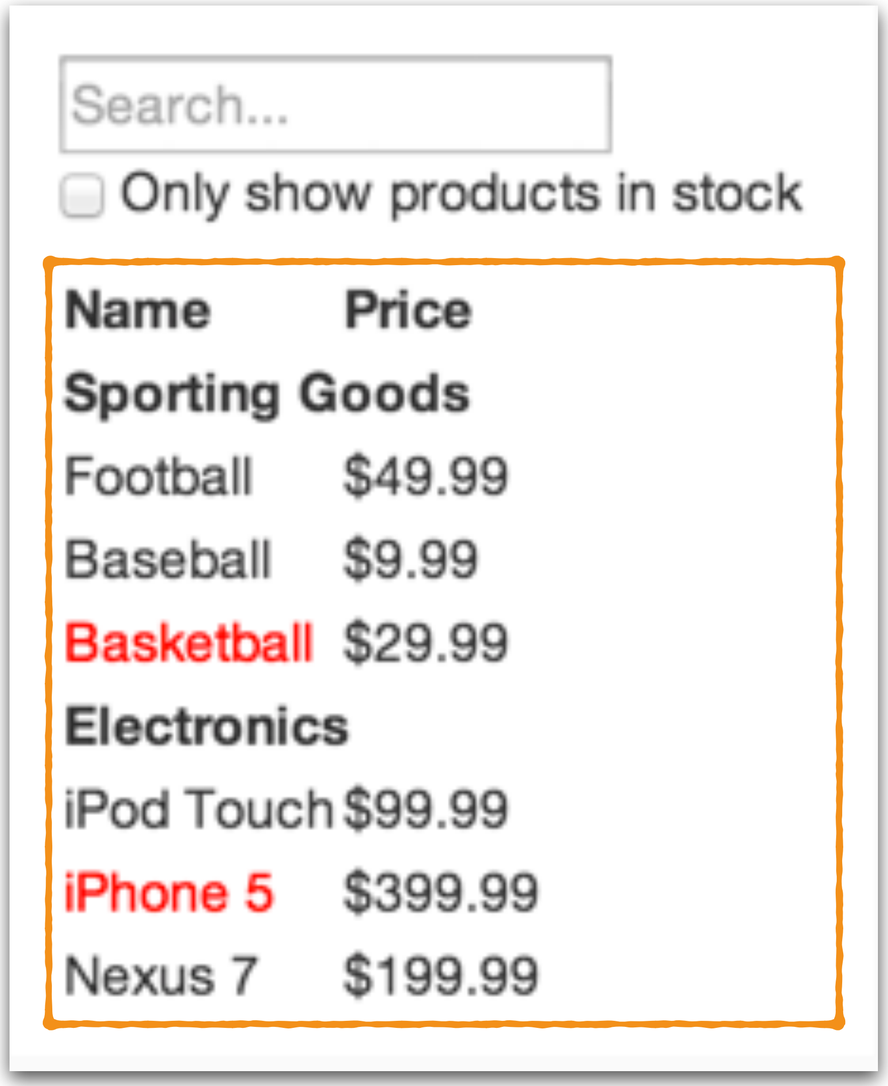
Displays all the products filtered on the basis of your search



SearchBar

User input specified for search filters

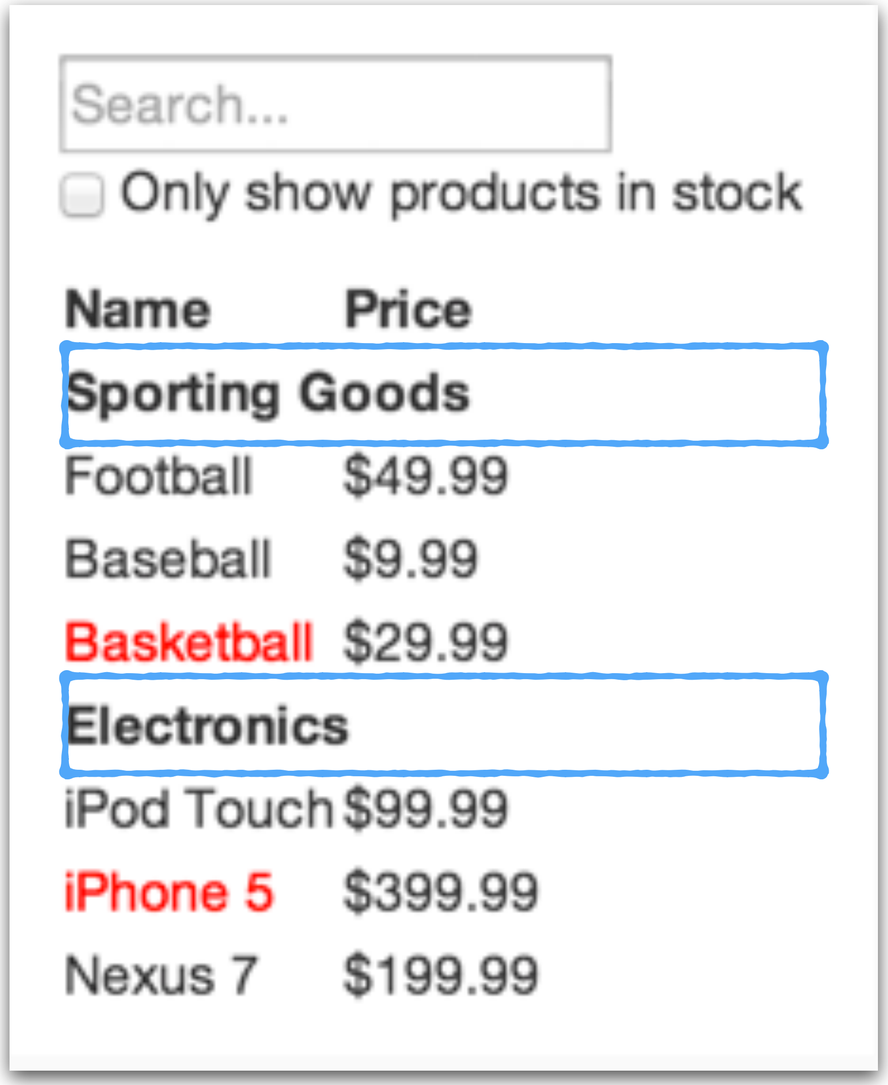
FilteredProductsTable
SearchBar



ProductTable

The actual display of products available

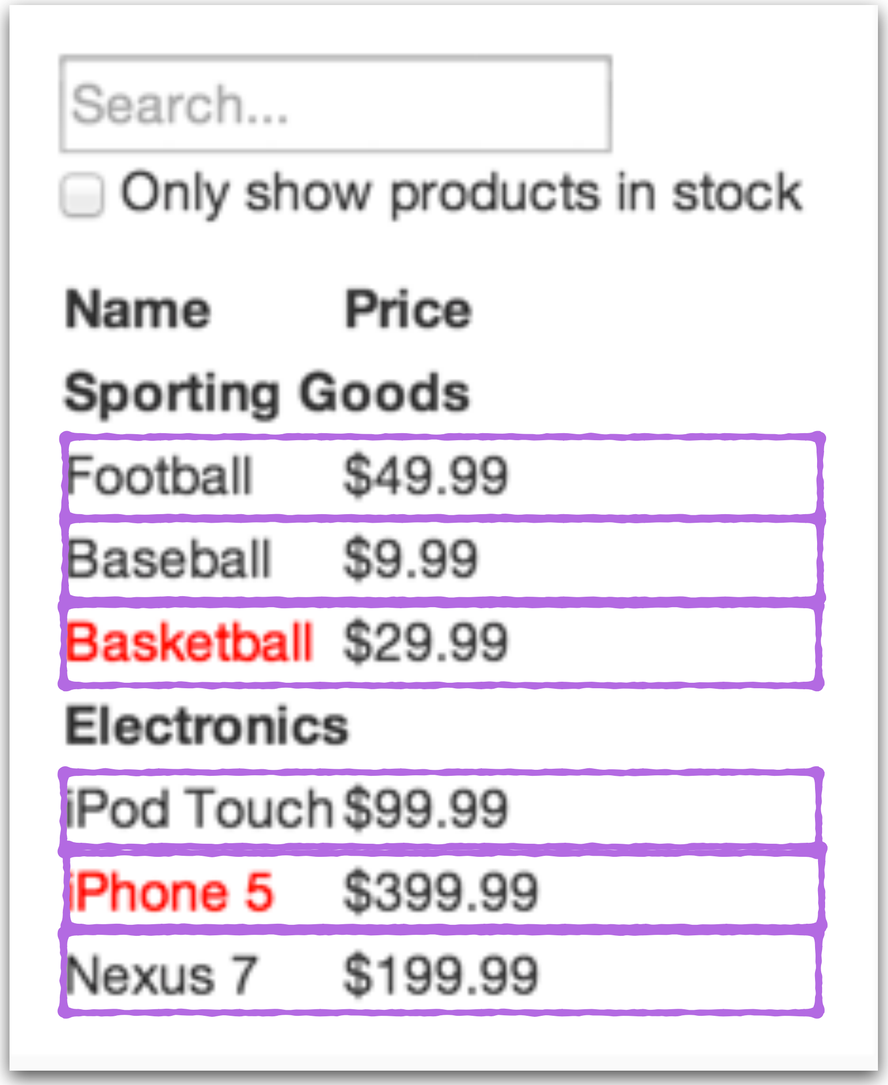
FilteredProductsTable
SearchBar
ProductTable



Category

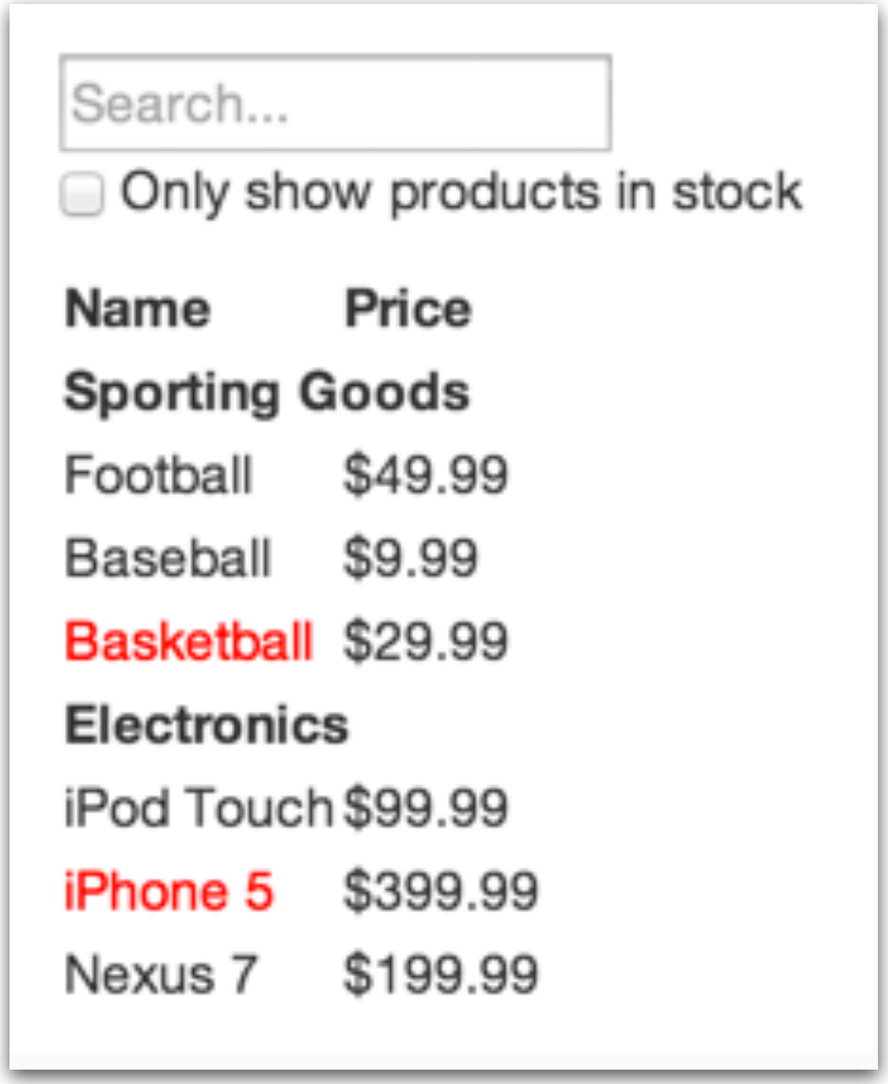
The category of products available

FilteredProductsTable
SearchBar
ProductTable
Category



Product

The actual products in rows



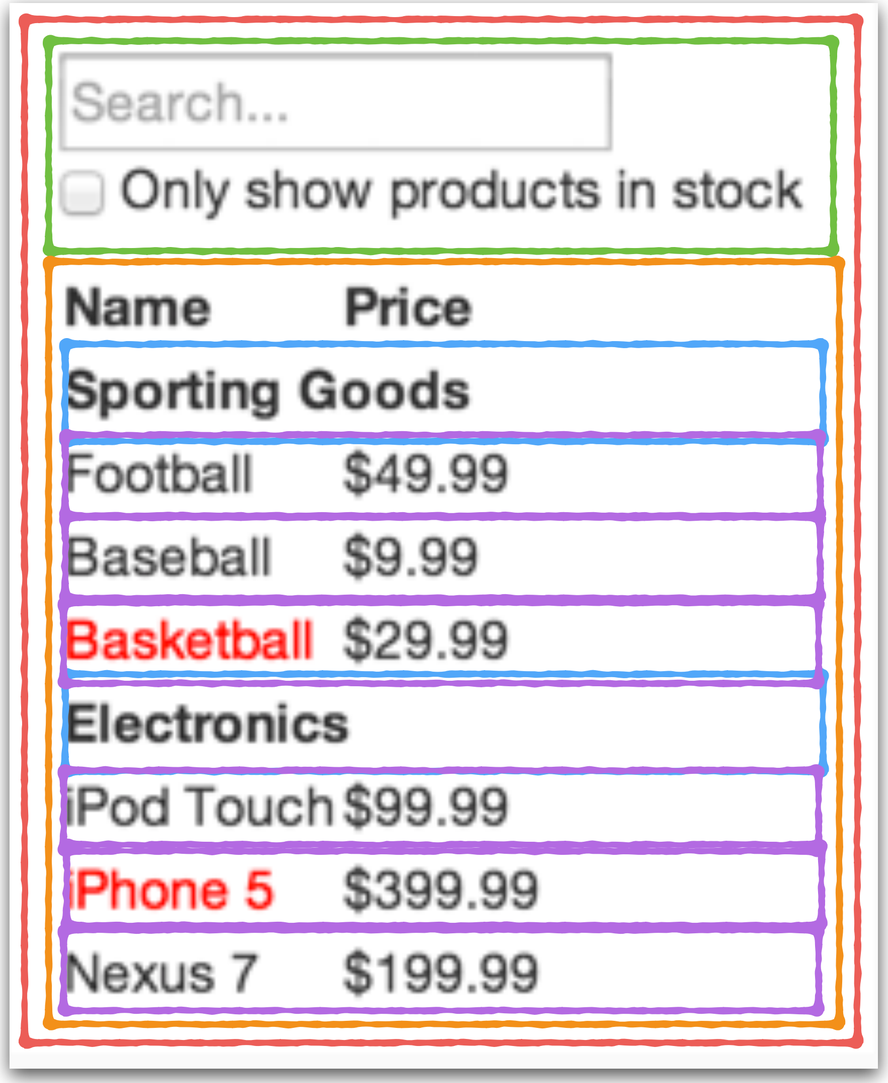
FilteredProductsTable

SearchBar

ProductTable

Category

Product



Components are well designed when each component has a **single responsibility**

React is NOT a framework

React is a concept and a library
built using that concept

REACT JS

What makes it cool?

What makes React cool?

components built with logic and behavior

fast updates using Virtual dom

Server side rendering

communication using one-way data flow

using properties and states

REACT JS

components built with logic and behavior

View components in React have both the logic and behavior of the view

REACT JS

components built with logic and behavior

This eliminates all the bridging code where UI changes are **relayed** to the JS code

REACT JS

components built with logic and behavior

This eliminates all the bridging
code where UI changes are
relayed to the JS code

And the JS code checks the
logic and then **updates** the DOM

REACT JS

components built with logic and behavior

This eliminates all the bridging code where UI changes are **relayed** to the JS code

And the JS code checks the logic and then **updates** the DOM

Grouping them together makes them manageable and more intuitive to maintain

What makes React cool?

components built with logic and behavior

fast updates using Virtual dom

Server side rendering

communication using one-way data flow

using properties and states

fast updates using Virtual dOM

DOM rendering tends to be what makes web applications really slow

fast updates using Virtual dOM

React speeds this up by maintaining a mirror copy of every DOM element

fast updates using Virtual dom

React speeds this up by
maintaining a mirror copy of
every DOM element

Imagine every DOM node recreated in-memory

This is the Virtual DOM!

fast updates using Virtual dom

This is the Virtual DOM!

The virtual DOM is an **abstraction** which allows us to make UI changes very **fast**

fast updates using Virtual dom

This is the Virtual DOM!

Any DOM changes you make are first made to the virtual DOM

fast updates using Virtual dom

This is the Virtual DOM!

Any DOM changes you make are
first made to the virtual DOM

React checks the difference between
the current DOM and the new DOM

fast updates using Virtual dom

This is the Virtual DOM!

Any DOM changes you make are first made to the virtual DOM

React checks the difference between the current DOM and the new DOM

And only the changed properties are applied to the real DOM!

What makes React cool?

components built with logic and behavior

fast updates using Virtual dom

Server side rendering

communication using one-way data flow

using properties and states

Server side rendering

The Virtual DOM also enables rendering complete UIs **without** a browser

Server side rendering

React allows you to create the entire web page on the server

And simply send it down to the client

Server side rendering

The initial page load becomes super fast!

What makes React cool?

components built with logic and behavior

fast updates using Virtual dom

Server side rendering

communication using one-way data flow

using properties and states

communication using one-way data flow

React uses **composition** to build individual view components

Components **contain** other components within them to build up complex views

communication using one-way data flow

Data passed to React components flows down the hierarchy - from parent to child

communication using one-way data flow

Data passed to React components flows down the hierarchy - from parent to child

This streamlines communications and makes React very intuitive

What makes React cool?

components built with logic and behavior

fast updates using Virtual dom

Server side rendering

communication using one-way data flow

using properties and states

Using properties and states

React components accept **immutable properties** which can be used to determine how the component renders

Using properties and states

React components also can have an **intrinsic state** which determines how the component looks and behaves

Using properties and states

This **separation** between properties and state is an important feature which makes the React pattern clear and easy to use

What makes React cool?

components built with logic and behavior

fast updates using Virtual dom

Server side rendering

communication using one-way data flow

using properties and states

REACT JS

npm set up

npm set up

This is a tool which makes it easy for Javascript developers to **share code** in the form of **packages**

A package is a small **building block** which usually solves a single problem well

npm set up

Let's install the **npm client** on our machine so we can get packages which are useful for this course

REACT JS

set up http-server

REACT JS

Accessing ReactJS Files

Accessing ReactJS Files

Let's see where you can find the latest React versions to reference

Accessing ReactJS Files

AccessReactJSFiles Video

EXAMPLE 1

HelloWorld.html

Show how we make directories where all the examples will live and start up http-server to run these examples

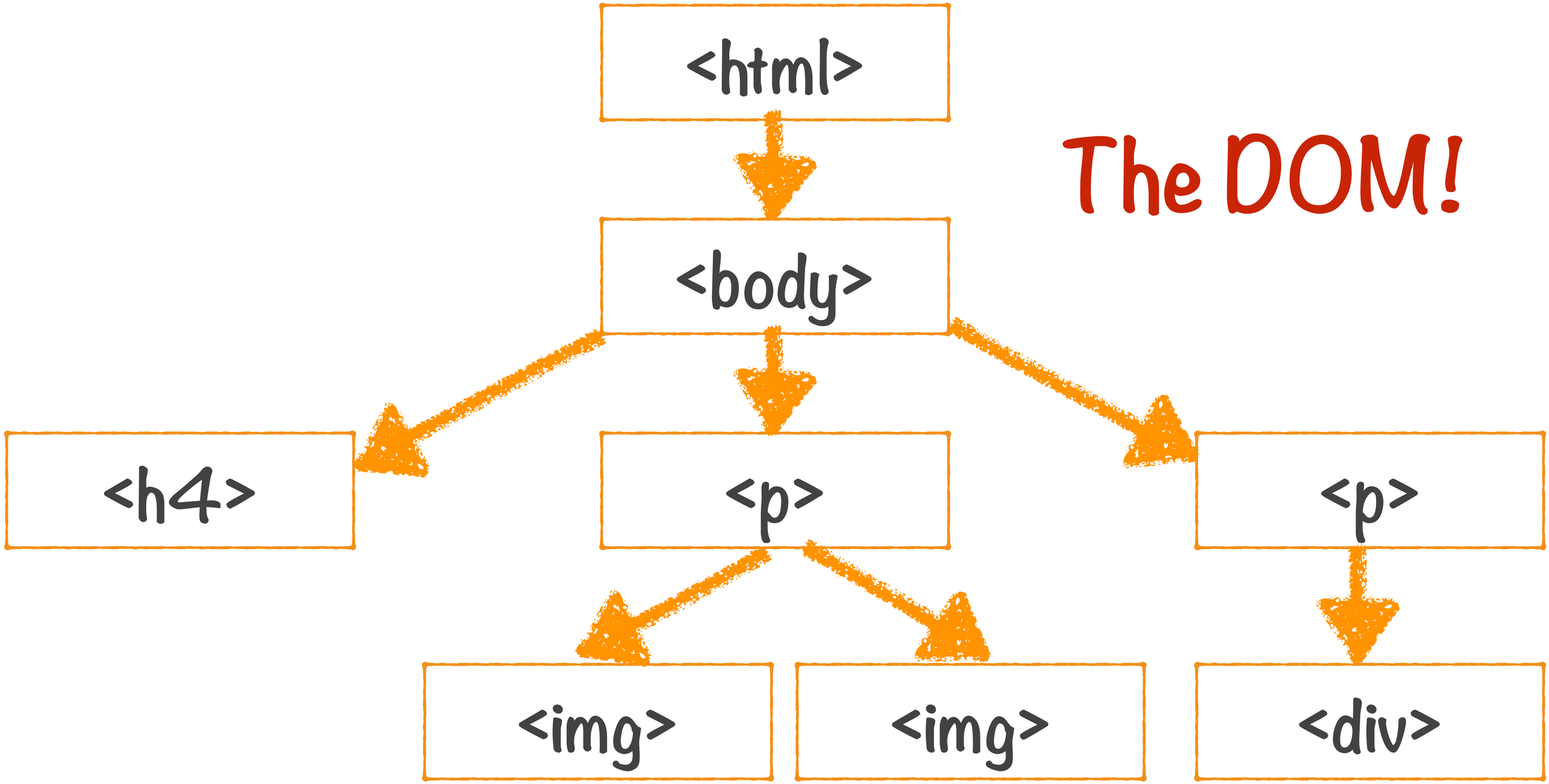
REACT JS

Virtual DOM

Virtual DOM

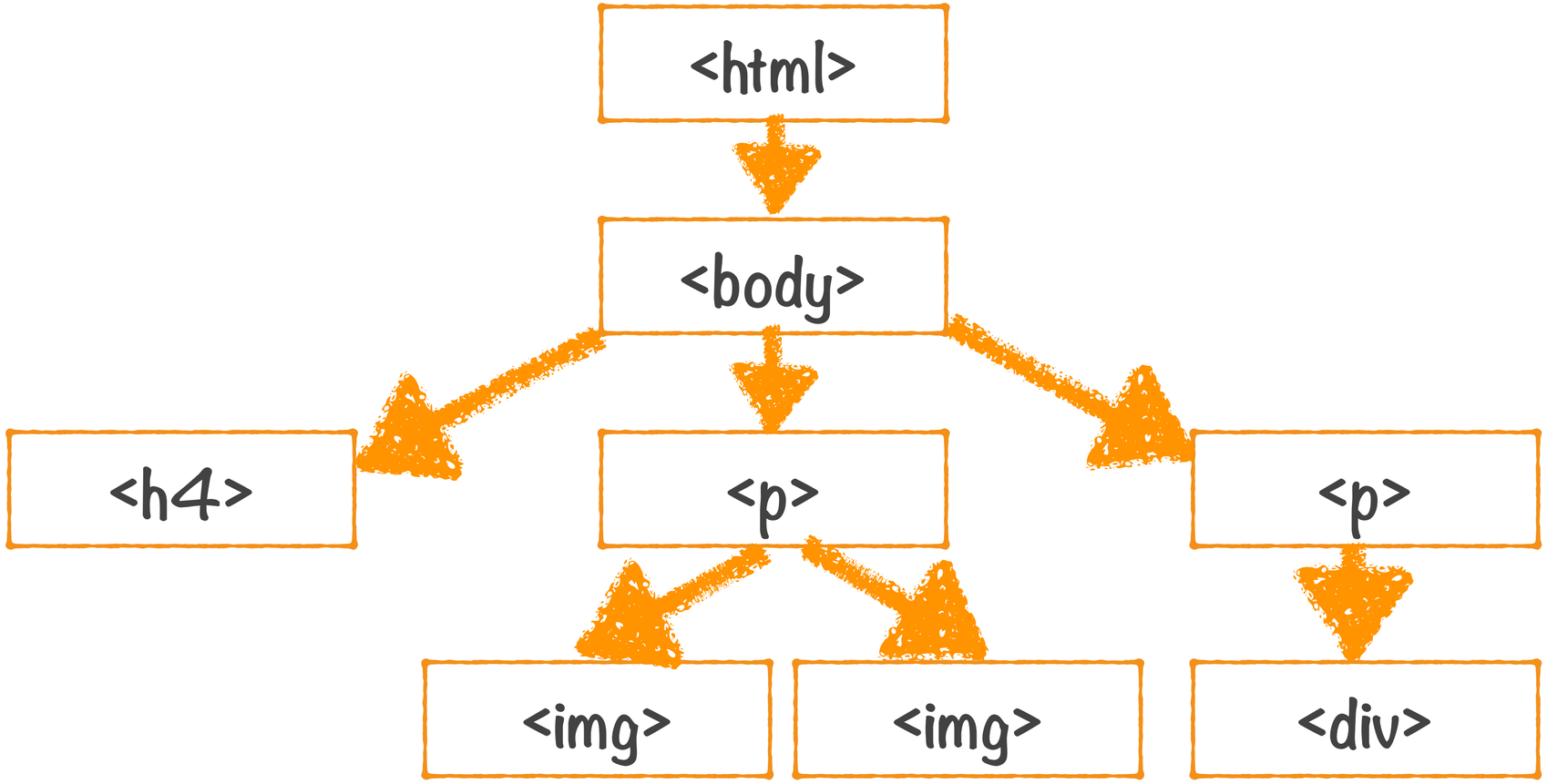
Every webpage can be represented as a **hierarchy** of elements

Virtual DOM



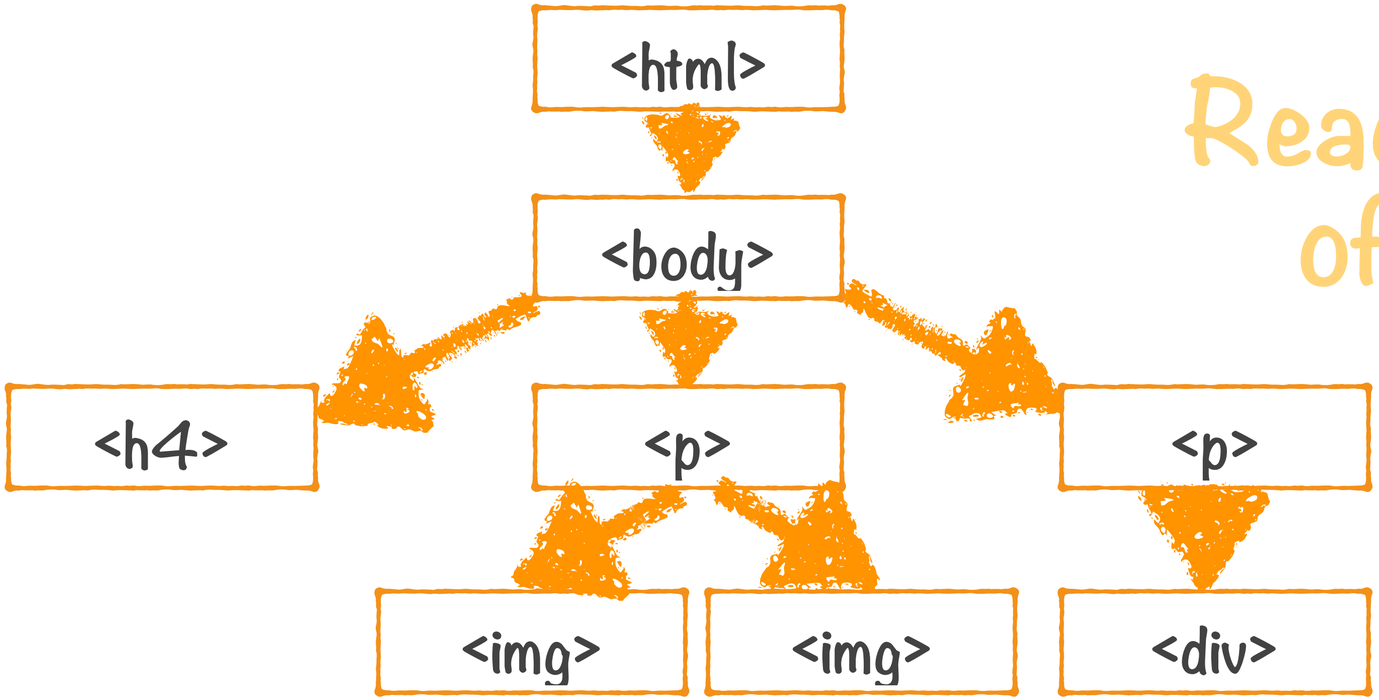
The DOM!

Virtual DOM



React stores an exact mirror of this DOM **in-memory**

Virtual DOM



React stores an exact mirror of this DOM **in-memory**

Webpages update the DOM in response to **user** events or **server** events

Virtual DOM

React stores an exact mirror
of this DOM **in-memory**

User events: Clicks, swipes, hover

Server events: Success and error
responses from the server

User events, Server events

These will change the state of the underlying **data model**

User events, Server events

These will change the state of
the underlying **data model**

React will first render these
changes on the **virtual DOM**

Virtual DOM

React will first render these changes on the **virtual DOM**

React will then calculate the **difference** between the previous virtual DOM and the current new virtual DOM

Virtual DOM

React will first render these changes on the **virtual DOM**

React will then calculate the **difference** between the previous virtual DOM and the current new virtual DOM

React only updates what is needed in the new DOM

Virtual DOM

React only updates what is needed in the new DOM

This process called **reconciliation** makes DOM updates very fast

Virtual DOM

This process called **reconciliation**
makes DOM updates very fast

React minimizes the number of DOM elements which should be updated

Virtual DOM

This process called **reconciliation**
makes DOM updates very fast

And is completely **abstracted** from
the user!

You simply write code as though the DOM were being updated in real-time!

EXAMPLE 2

NestedElements.html