

A Javascript Library for Building User Interfaces

# Who are we?

Luke Chui  
ACM Hack  
lukechui.com

Vic Yeh  
ACM Hack  
@vicohyeh

**SLAVE = WEB  
FRONTEND**

**MASTER = BACKEND**

[memegenerator.net](http://memegenerator.net)



# Traditional MVC Frameworks

Model

structure of data

Controller

logic that connects  
the model and the  
view

View

elements in the  
user interface

**Is there a more efficient, declarative, and JS-friendly alternative?**

# ReactJS

“Just the V in MVC”

- Built upon the idea that DOM manipulation is expensive
- Super fast for app that has a lot of state changes
- Allow full power of JS in UI
- Functional reactive programming

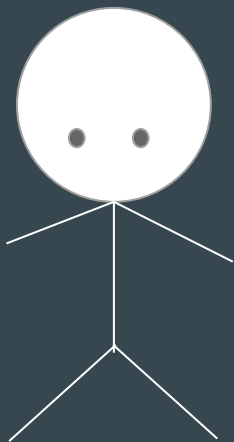
# Key Concepts

- Re-render everything on every change
- Virtual DOM
- Composable UI Components

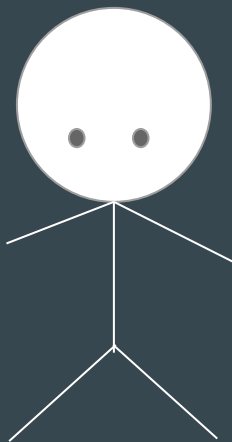


# Virtual DOM

- In-memory data structure
- Minimize DOM changes -> increase performance



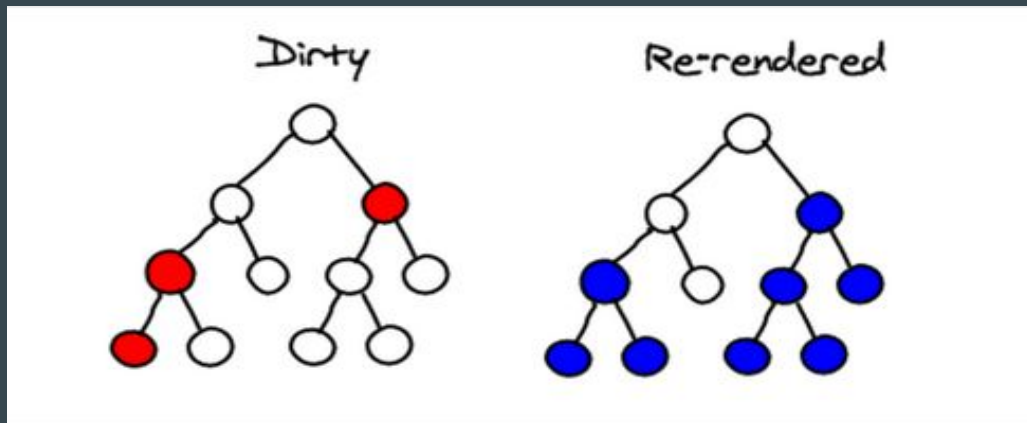
Real



Virtual

# Re-render everything on every change?

- 1) Build new Virtual DOM tree
- 2) Diff with the old one
- 3) Compute minimal set of changes required
- 4) Batch render all updates



# JSX

- No required
- Mixture of HTML and JS

ex.

```
var HelloMessage = React.createClass({  
  render: function() {  
    return <div>Hello, ACM Hack</div>;  
  }  
});
```

# Composable UI Components

- React is all about building reusable UI components

Dashboard

TodoList

Search

JSX

```
<div>  
  <TodoList />  
  <Search />  
</div>;
```

# Composable UI Components

```
var TodoList = React.createClass({
  render: function() {
    var createItem = function(item) {
      return <div key={item.id}>{item.text}</div>;
    }
    return <ul>{this.props.items.map(createItem)}</ul>;
  }
});

var Dashboard = React.createClass({
  render: function() {
    return <div>
      <TodoList items={this.state.items}/>
    </div>;
  }
});
```

DEMO

