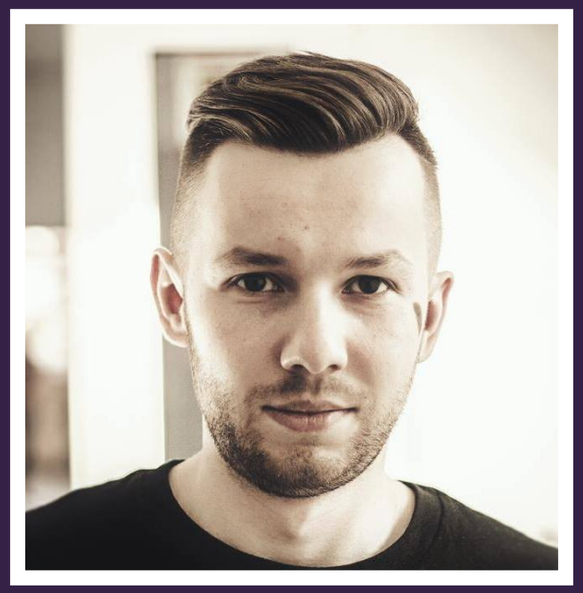
REACT JS



PRZEMYSŁAW WISZOWATY

HELLO!

SOFTware HUT

TENDERHUT GROUP

meet.js
białystok

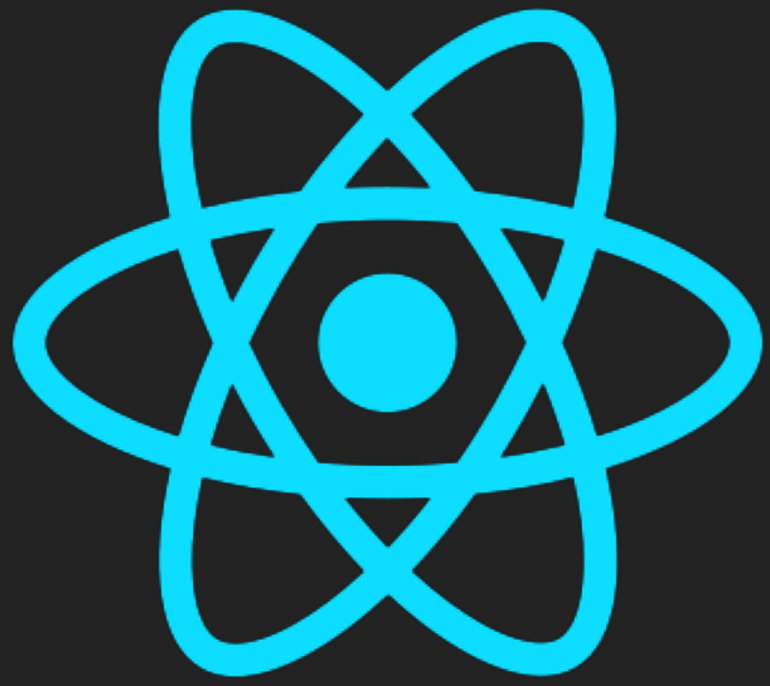
REACT IS EVERYWHERE

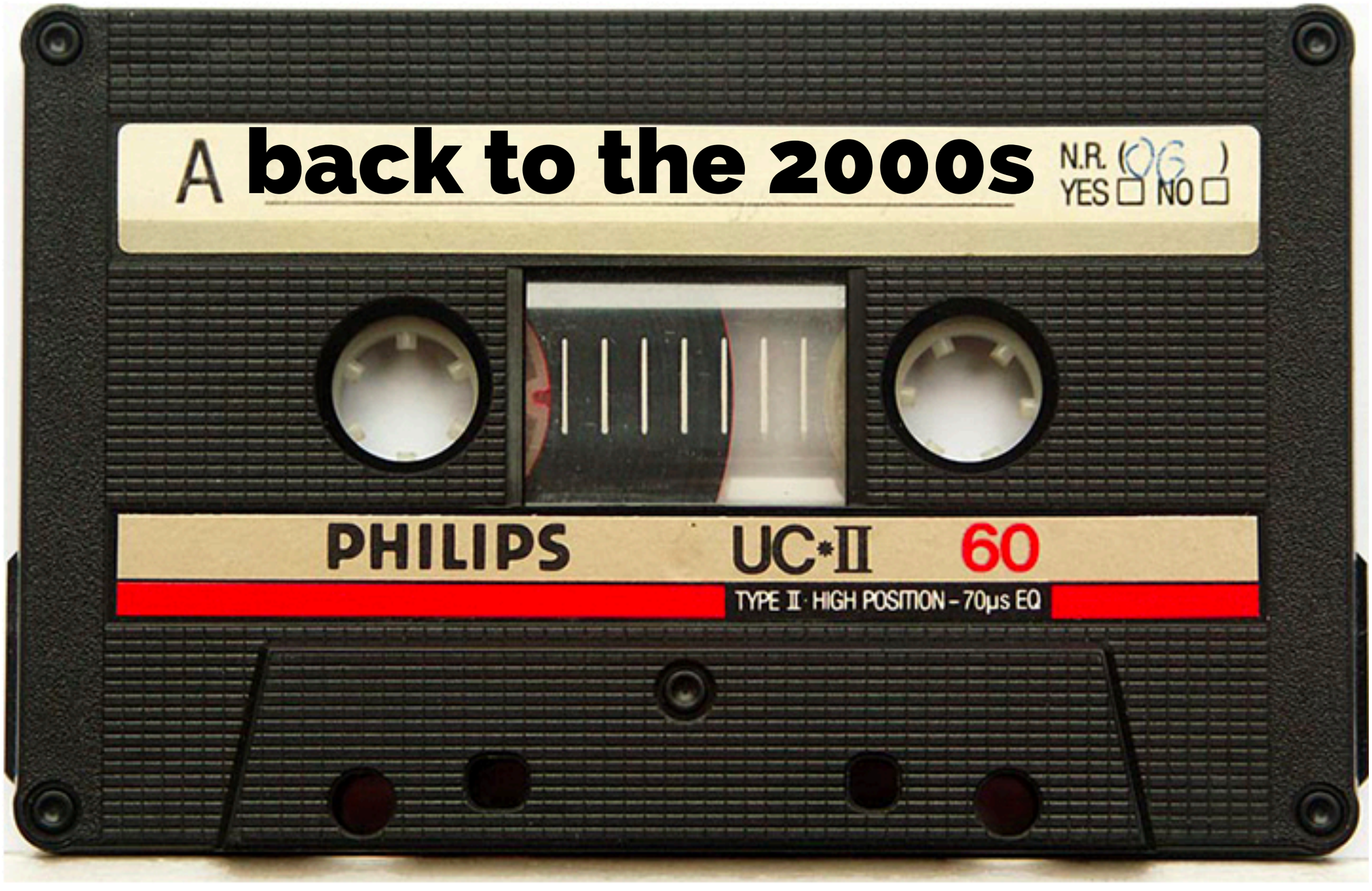
„Learn Once, Write Anywhere“

FRONTEND

BACKEND

MOBILE




A back to the 2000s
N.R. (06)
YES ☐ NO ☐
PHILIPS
UC*II
60
TYPE II · HIGH POSITION – 70µs EQ

The Geocities-izer

ge Look Like It Was Made By A 13 Year-Old In 1996

Type any URL in the box below and click Submit to see how it would look as a Geocities page.

Or Try one of these: The New York Times YouTube BoingBoing

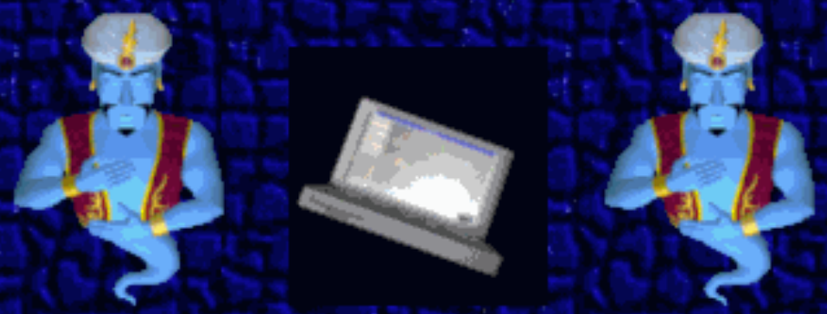
http://

Submit

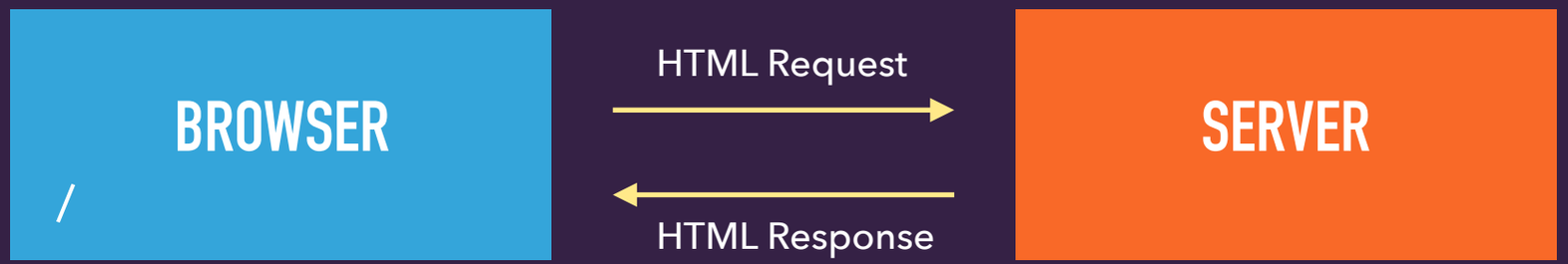
Some pages may work very slowly or not all. Many webapps are just too advanced for Geocities.

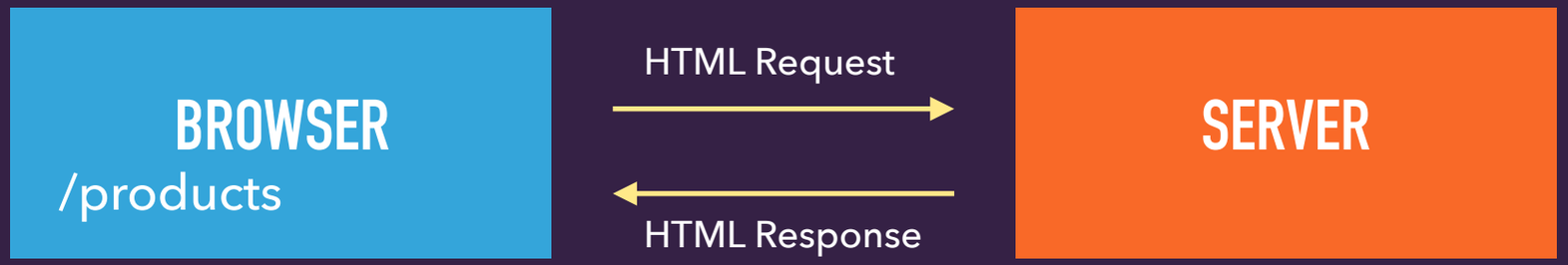
Turn your sound up for the full effect.

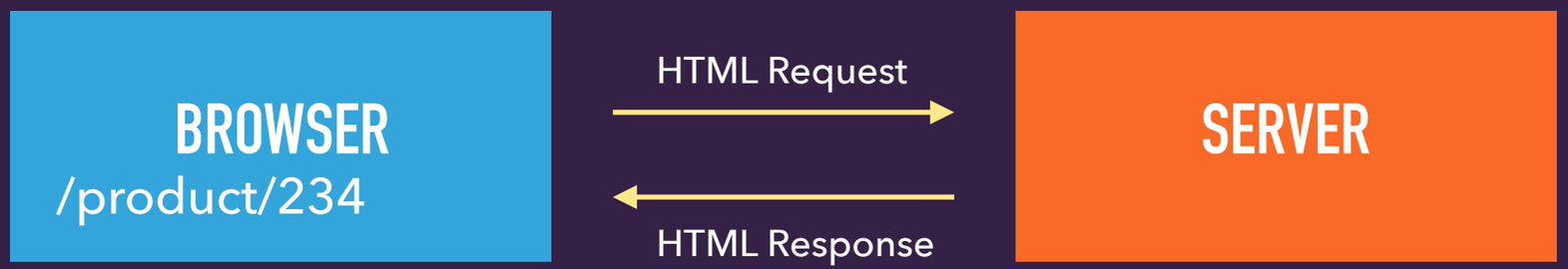
Created by Mike Lacher



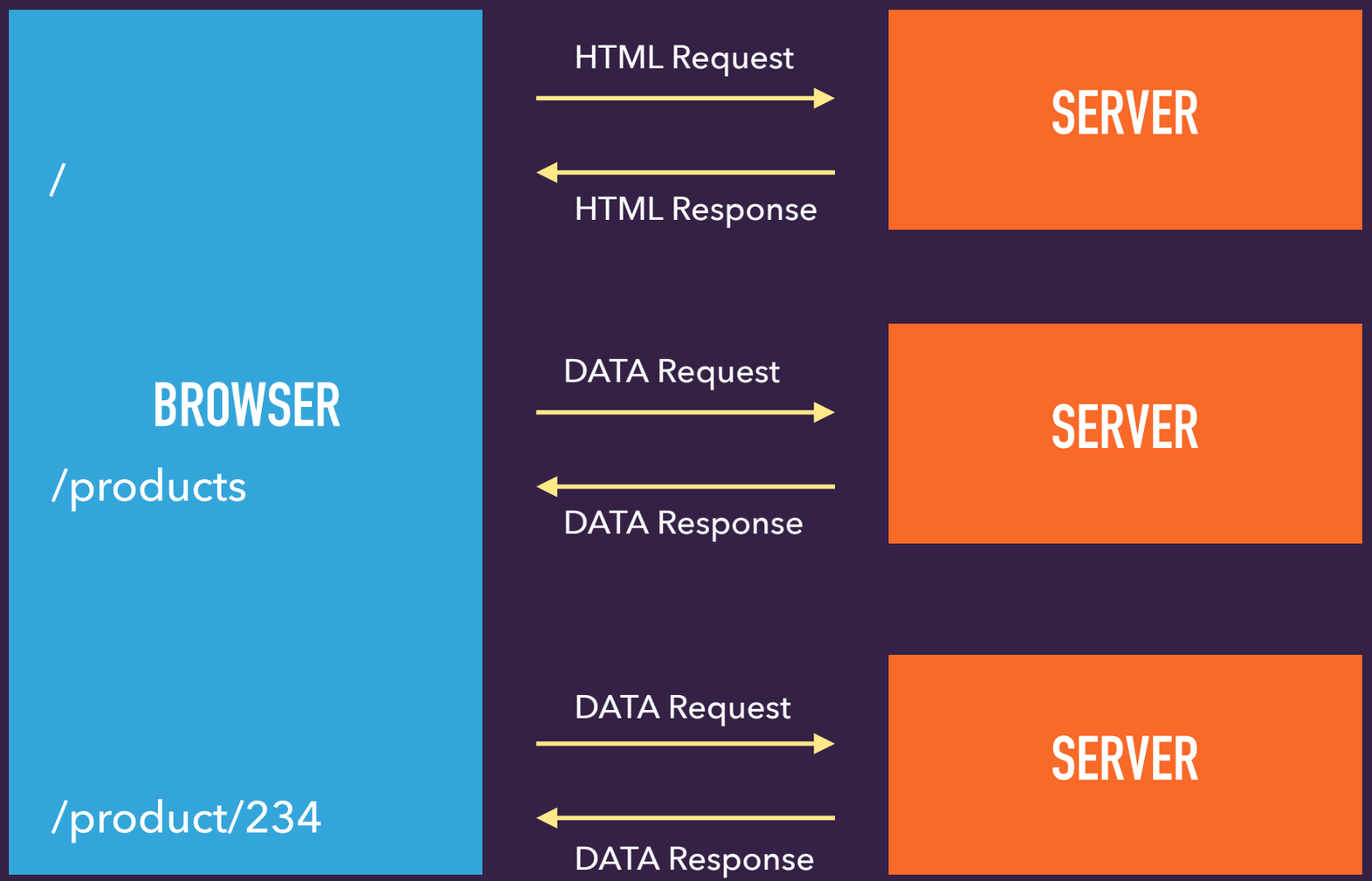
TRADITIONAL WEBSITE







SINGLE PAGE APPLICATION



Chrome Plik Edytuj Widok Historia Zakładki Osoby Okno Pomoc

TenderHut S.A. - Technologicz

https://tenderhut.com

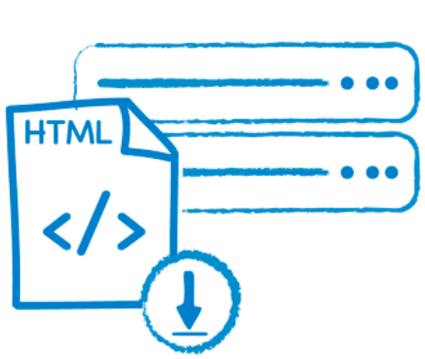
414 × 736

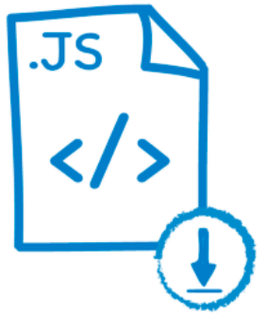
Elements Console Sources Network Performance Memory Application Security Audits

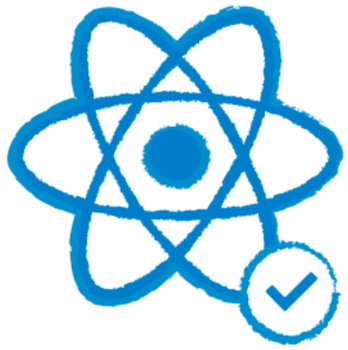
(index) :formatted

```
<!DOCTYPE html>
<html lang="en" prefix="og: http://ogp.me/ns#">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,initial-scale=1,shrink-to-fit=no">
        <link rel="apple-touch-icon" sizes="180x180" href="/icons/apple-touch-icon.png?v=694AzX47gQ">
        <link rel="icon" type="image/png" sizes="32x32" href="/icons/favicon-32x32.png?v=694AzX47gQ">
        <link rel="icon" type="image/png" sizes="16x16" href="/icons/favicon-16x16.png?v=694AzX47gQ">
        <link rel="mask-icon" href="/icons/safari-pinned-tab.svg?v=694AzX47gQ" color="#5bbad5">
        <link rel="shortcut icon" href="/icons/favicon.ico?v=694AzX47gQ">
        <meta name="msapplication-TileColor" content="#2d89ef">
        <meta name="msapplication-config" content="/icons/browserconfig.xml?v=694AzX47gQ">
        <meta name="theme-color" content="#ffffff">
        <meta name="google-site-verification" content="NCJvpBpTbp9pUJdDk7t6dhIisOj4kZJEphgntRU-TbM"/>
        <link rel="manifest" href="/manifest.json">
        <title>TenderHut</title>
        <link href="/static/css/main.5ca042e5.css" rel="stylesheet">
    </head>
    <body>
        <noscript>You need to enable JavaScript to run this app.</noscript>
        <div id="app-root"></div>
        <div id="modal-root"></div>
        <script type="text/javascript" src="/static/js/main.dc65024f.js"></script>
    </body>
</html>
```

CSR





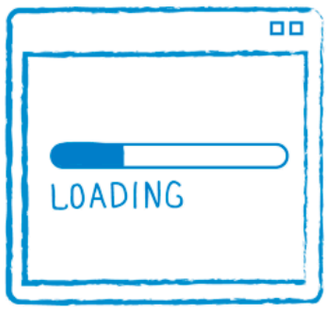


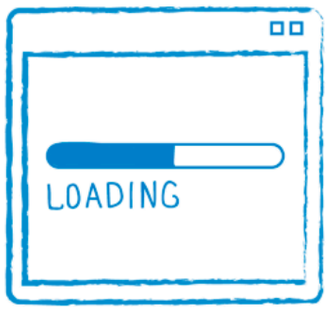
Server Sending Response to Browser

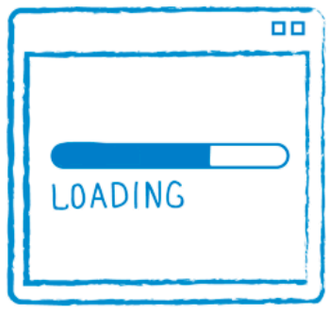
Browser Downloads JS

Browser executes React

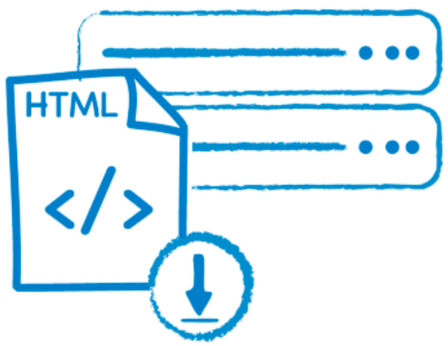
Page Now **Viewable** and **Interactable**



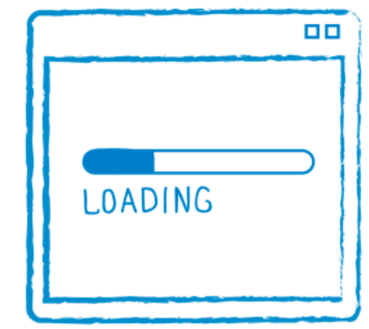


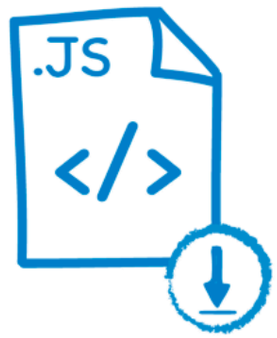


SSR

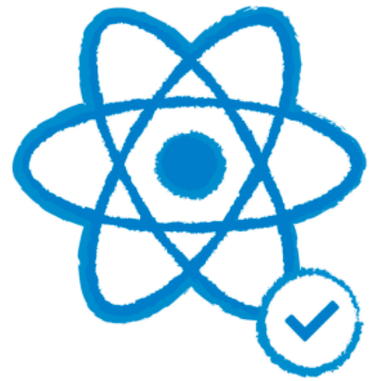


Server Sending Ready to be rendered HTML Response to Browser





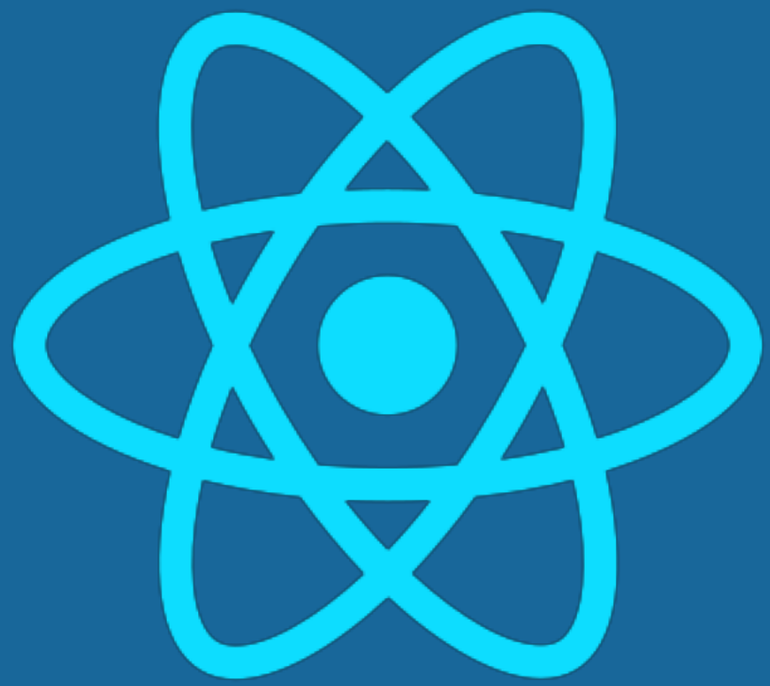
Browser Renders the page, **Now Viewable**, and Browser Downloads JS



Browser executes React

Page Now **Interactable**

WHAT IS REACT?

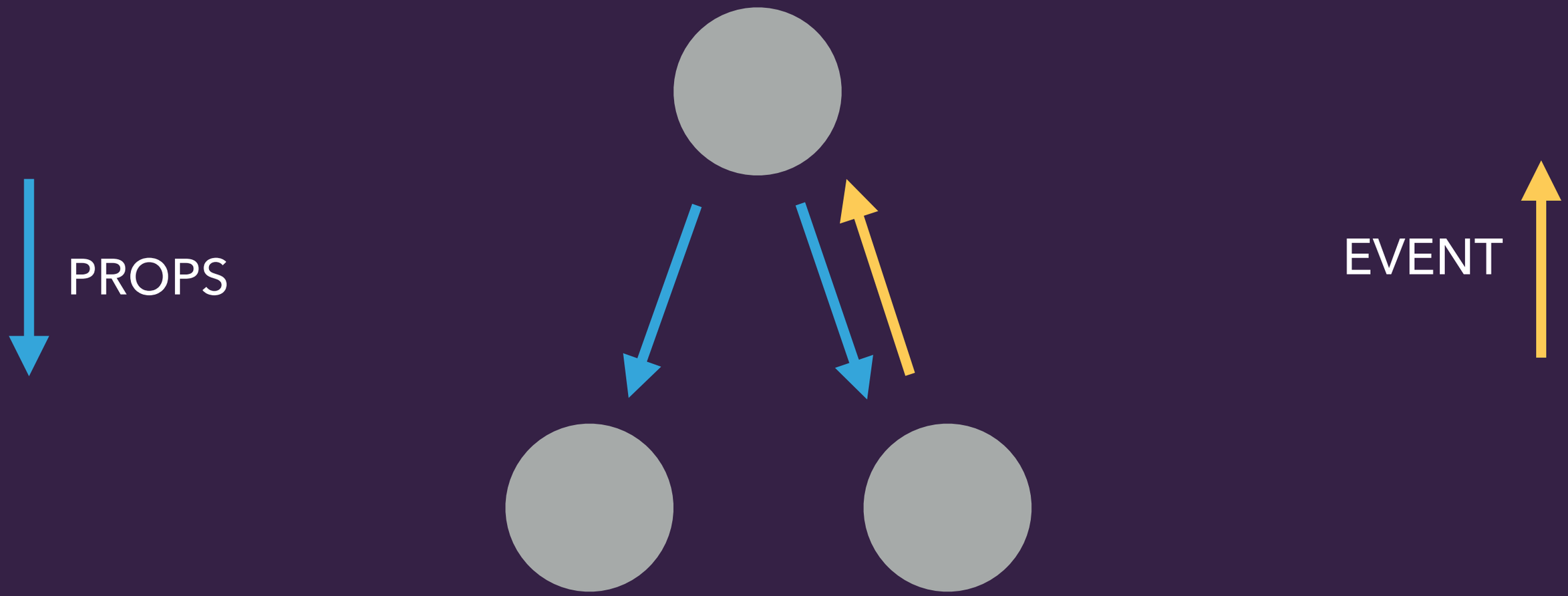


LIBRARY NOT ~~FRAMEWORK~~

LOW LEARNING CURVE

ONE WAY DATA FLOW

ONE WAY DATA FLOW



NO CONTROLLERS

NO MODELS

NO DIRECTIVES

NO GLOBAL EVENT LISTENER

JUST COMPONENT

COMPONENT

ISOLATED

REUSABLE

TESTABLE

COMPONENT TYPES

FUNCTIONAL

```
function Welcome(props) {
    return <h1>Hello, {props.name}</h1>;
}
```

FUNCTIONAL

```
const Welcome = props => {
    return <h1>Hello, {props.name}</h1>;
};
```

FUNCTIONAL

```
const Welcome = ({name}) => {
    return <h1>Hello, {name}</h1>;
};
```

CLASS

```
class Welcome extends React.Component {
    render() {
        const {name} = this.props;
        return (
            <h1>Hello, {name}</h1>
        )
    }
}
```

EVERYTHING IS A COMPONENT

CardComponent

CardHeaderComponent

CardBodyComponent

UserPhotoComponent

```jsx
import React from 'react';

const FacebookComponent = () => (
    <CardComponent>
        <CardHeaderComponent>
            <SelectorComponent />
            <SelectorComponent />
            <SelectorComponent />
        </CardHeaderComponent>

        <CardBodyComponent>
            <UserPhotoComponent />
        </CardBodyComponent>

        <CardFooterComponent />
    </CardComponent>
);

export default FacebookComponent;

```

JSX

```jsx
const Welcome = () => {
  return <h1>Hello, World</h1>;
};
```

JSX

```js
const Welcome = function Welcome() {
    return React.createElement(
        type: "h1",
        props: null,
        children: "Hello, World"
    );
};
```

JS

REACT DOESN'T REQUIRE USING JSX

VIRTUAL DOM

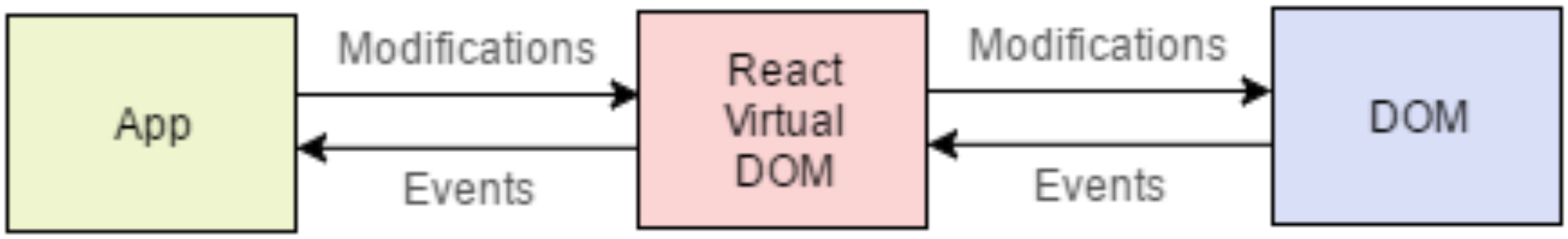
IT'S FAST

IT'S PURE

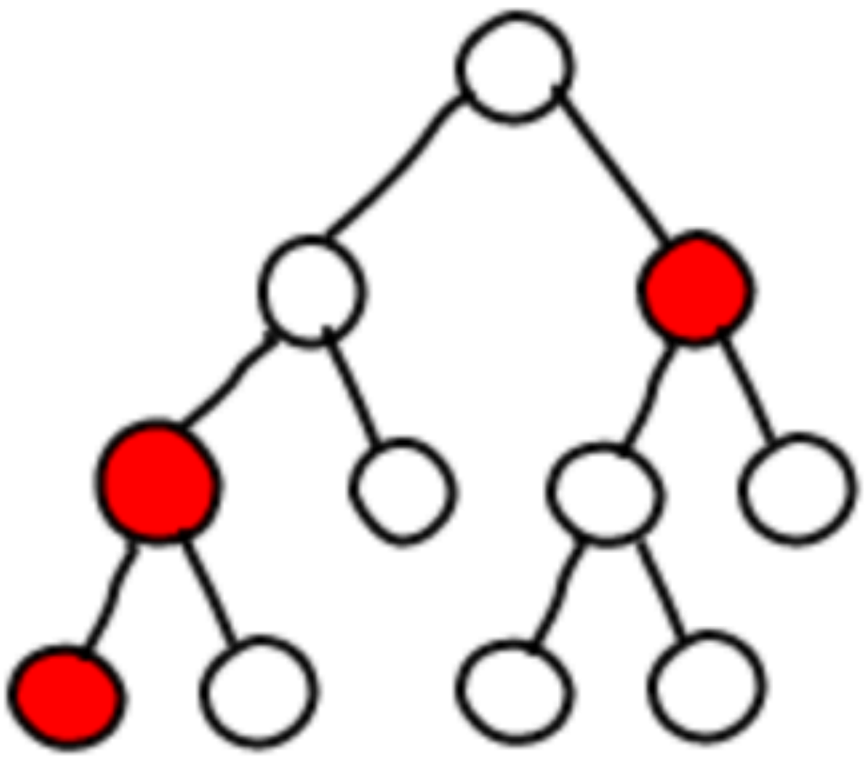
IT WORKS

Traditional Web Application

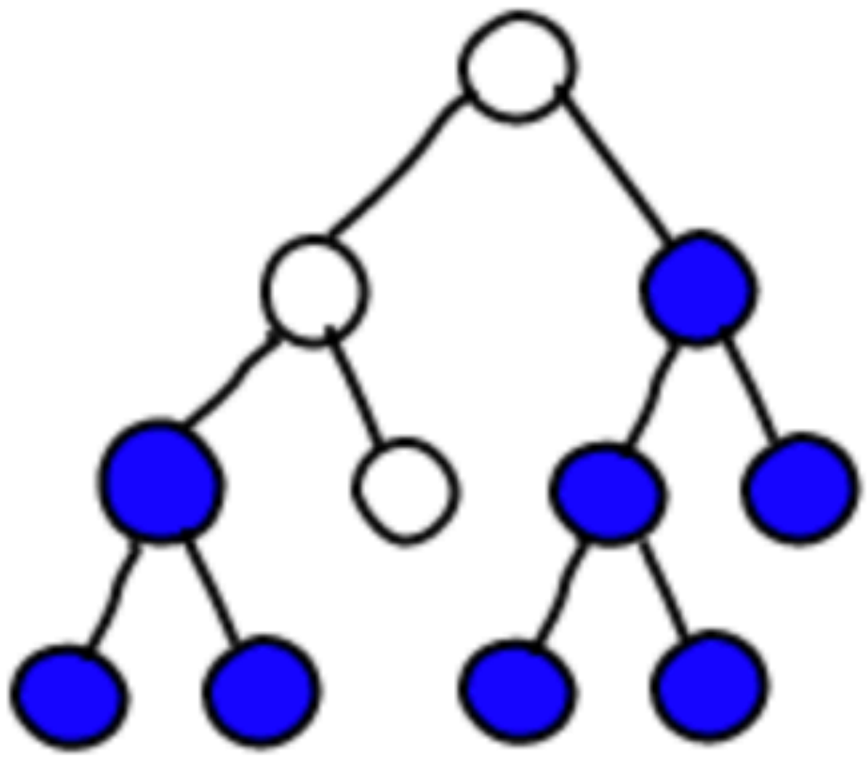
React.js



Dirty

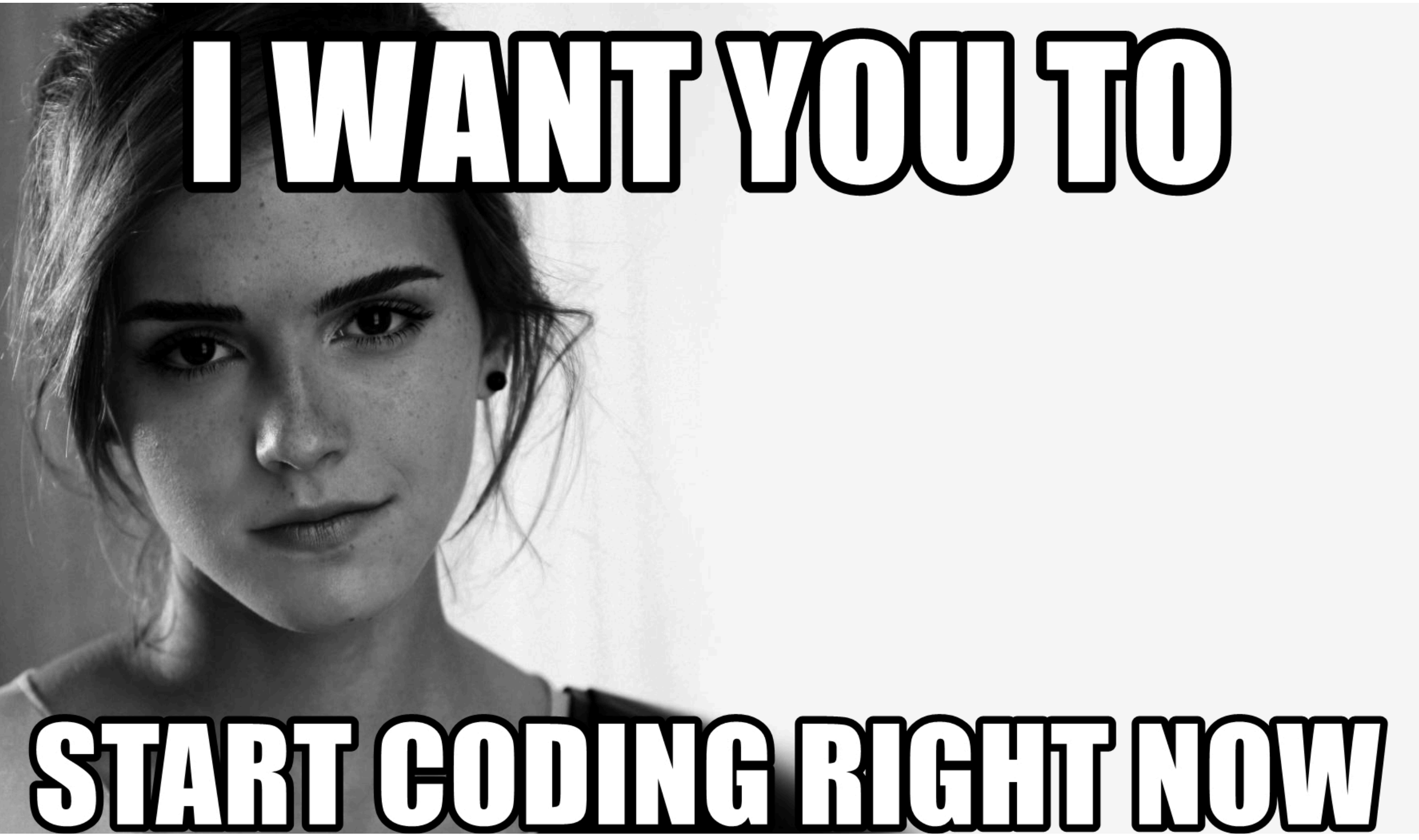


Re-rendered



Hello, world!

It is 12:26:46 PM.


I WANT YOU TO
START CODING RIGHT NOW

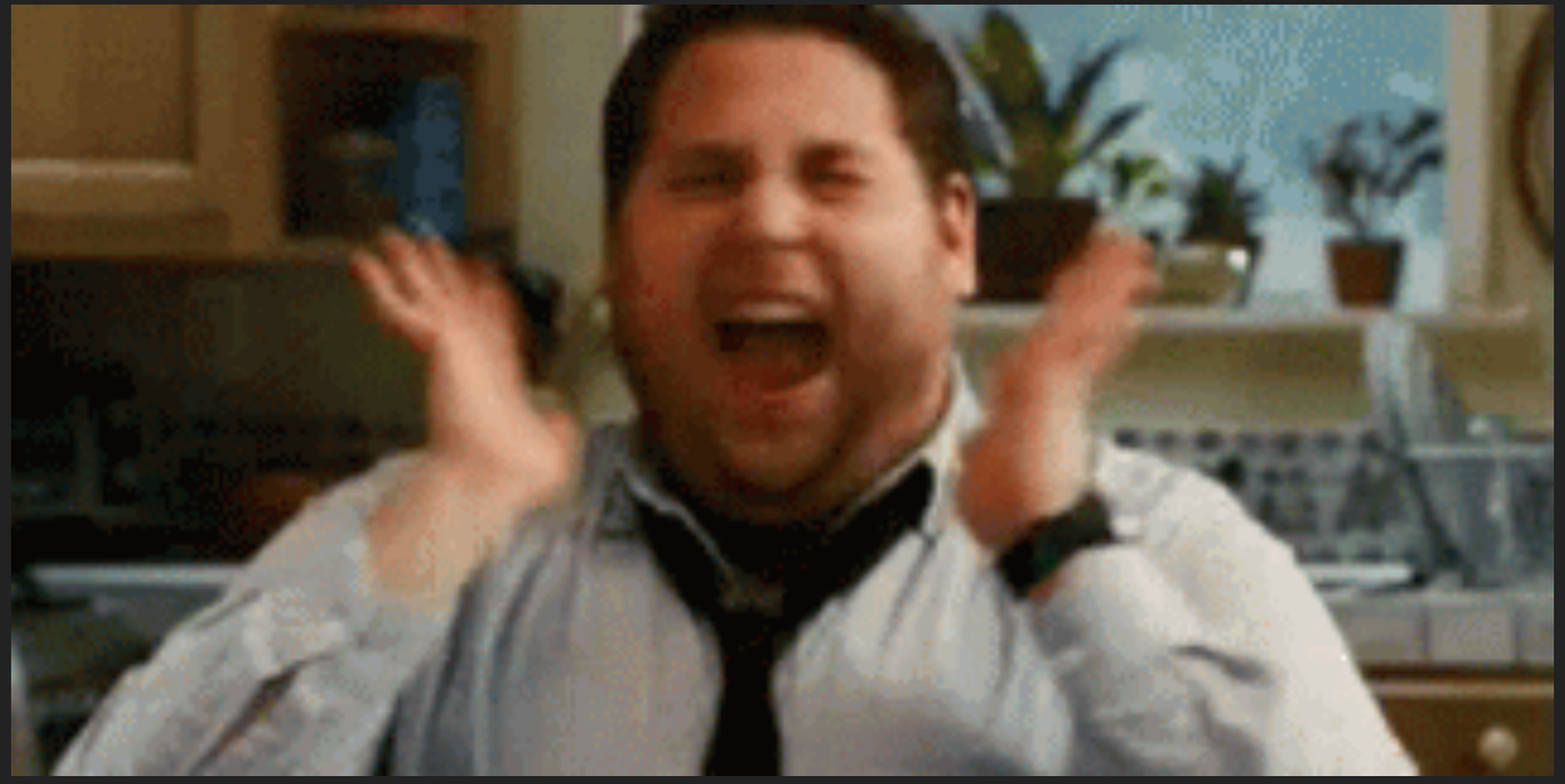
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>React App</title>
    <script src="https://unpkg.com/react@16/umd/react.production.min.js" crossorigin></script>
    <script src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js" crossorigin></script>
  </head>
  <body>
    <div id="root"></div>

    <script type="text/javascript">
        var Welcome = function Welcome() {
            return React.createElement("h1", null, "Hello, World");
        };

        ReactDOM.render(Welcome, document.getElementById("root"));
    </script>
  </body>
</html>
```

CREATE REACT APP

npx create-react-app myapp



DEMO

REACT DEV TOOLS

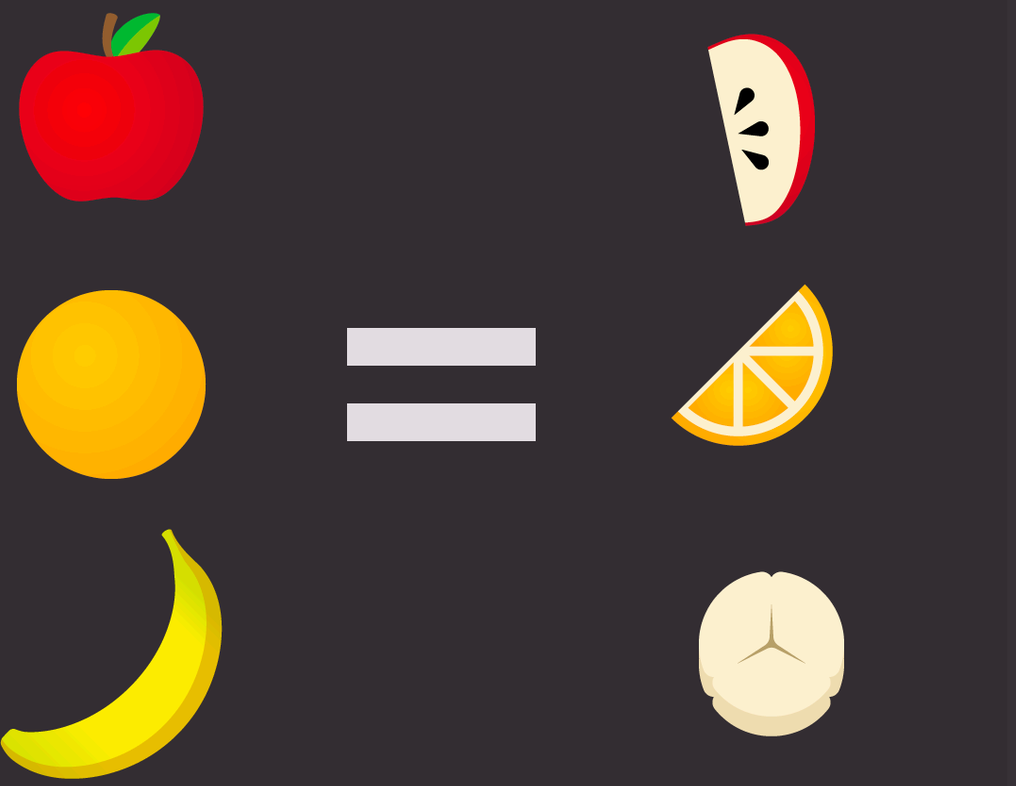
DEMO

THANKS

REACT JS

FUNCTIONAL PROGRAMMING

MAP



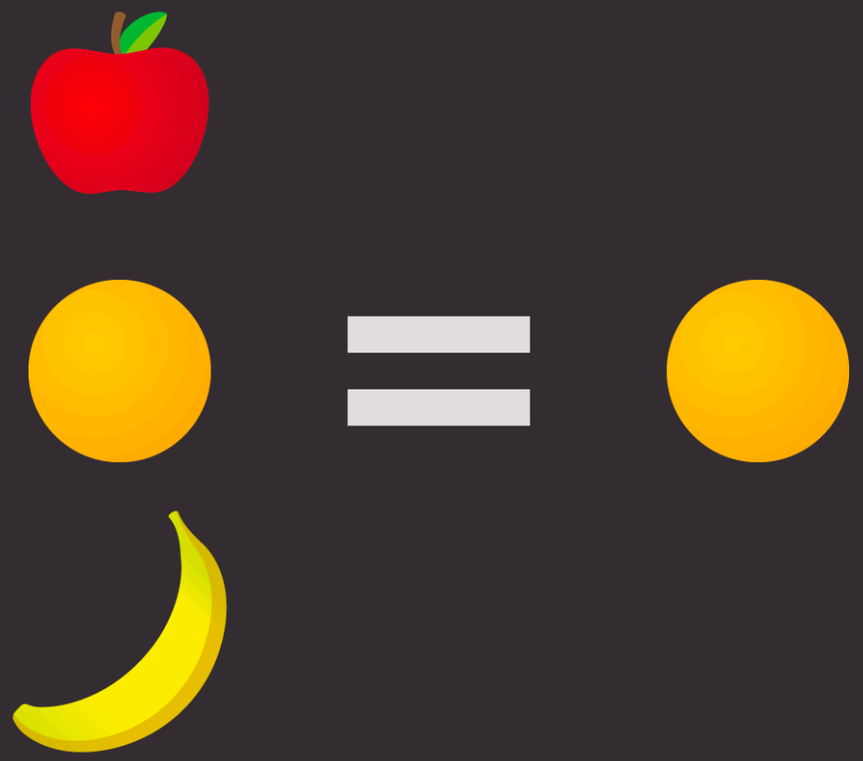
MAP

```
const array = [1,2,3,4,5];
```

```
const newArray = array.map(element => element + 1);
```

```
> newArray[2,,,,]
> newArray[2,3,,,]
> newArray[2,3,4,,]
> newArray[2,3,4,5,]
> newArray[2,3,4,5,6]
```

FILTER



FILTER

```
const array = [1,2,3,4,5];
const newArray = array.map(element => element > 3);
```

```
> newArray[]
> newArray[]
> newArray[]
> newArray[4]
> newArray[4,5]
```

RENDER MULTIPLE COMPONENT

RENDER MULTIPLE COMPONENT

```jsx
const App = () => (
  <div>
    <h4>Asia</h4>
    <h4>Paweł</h4>
    <h4>Gosia</h4>
    <h4>Darek</h4>
  </div>
);
```

```jsx
const App = () => (
  <div>
  {[
    <h4>Asia</h4>,
    <h4>Paweł</h4>,
    <h4>Gosia</h4>,
    <h4>Darek</h4>
  ]}
  </div>
);
```

RENDER MULTIPLE COMPONENT

```
const names = [
    "Asia",
    "Paweł",
    "Gosia",
    "Darek"
];
```

map()

```
{[
    <h4>Asia</h4>,
    <h4>Paweł</h4>,
    <h4>Gosia</h4>,
    <h4>Darek</h4>
]}
```

RENDER MULTIPLE COMPONENT

```
const names = ["Asia", "Paweł", "Gosia", "Darek"];
```

```
names.map(name => <h4>{name}</h4>);
```

```
{[
  <h4>Asia</h4>,
  <h4>Paweł</h4>,
  <h4>Gosia</h4>,
  <h4>Darek</h4>
]}
```

RENDER MULTIPLE COMPONENT

```
const names = ["Asia", "Paweł", "Gosia", "Darek"];

const App = () => (
  <div>
    {names.map(name => (
      <h4>{name}</h4>
    ))}
  </div>
);
```

```
00:35:03.447 ▶Warning: Each child in an array or iterator should have  index.js:1452
a unique "key" prop.

Check the render method of `App`. See https://fb.me/react-warning-keys
for more information.
    in h4 (at App.js:10)
    in App (at src/index.js:7)
```

KEY SHOULD BE UNIQUE

NOT

~~TIMESTAMP~~

~~RANDOM~~

GOOD

```
▼ <App> == $r
  ▼ <div>
    ▼ <div className="content">
        <h4 key="0">Asia</h4>
        <h4 key="1">Paweł</h4>
        <h4 key="2">Gosia</h4>
        <h4 key="3">Darek</h4>
      </div>
    ▼ <div className="sidebar">
        <h4 key="0">Skoda</h4>
        <h4 key="1">VW</h4>
        <h4 key="2">BMW</h4>
        <h4 key="3">Porsche</h4>
      </div>
    </div>
</App>
```

ANTI PATTERN

KEY={INDEX}

BUT.....

DEMO

KEY ISN'T PASSED TO COMPONENT

PROPS

PROPS = PRIMITIVE VALUES, REACT ELEMENTS FUNCTIONS

PROPS
DEFAULT IS TRUE

```jsx
<MyComponent dark />
```

```jsx
<MyComponent dark={true} />
```

PROPS
SPREAD ATTRIBUTES

```
const postData = {
  title: "Post Title",
  image: "https://pic.com/1.jpg",
  text: "Lorem ipsum dolor sit."
};

<Post {...postData} />
```

=

```
<Post
    title="Post Title"
    image="https://pic.com/1.jpg"
    text="Lorem ipsum dolor sit."
/>
```

NOTHING RENDER

```
<div />
```

```
<div></div>
```

```
<div>{false}</div>
```

```
<div>{null}</div>
```

```
<div>{undefined}</div>
```

```
<div>{true}</div>
```

CONDITIONAL RENDERING

CONDITIONAL RENDERING

```
const App = () => {
  if (post) {
    return <Post {...post} />;
  } else {
    return null;
  }
};
```

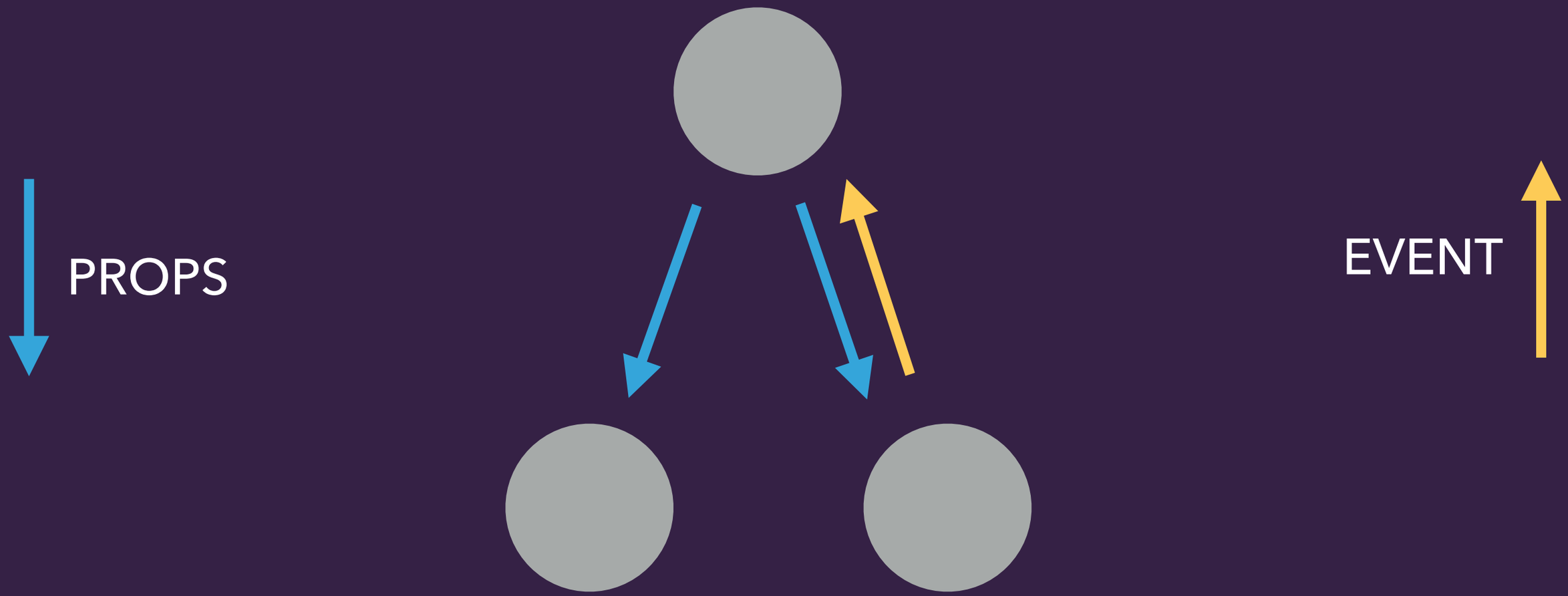
```
const App = () => {
  return post ? <Post {...post} /> : null;
};
```

CONDITIONAL RENDERING

```
const App = () => {
  return post && <Post {...post} />;
};
```

```
true && expression = expression
false && expression = false
```

ONE WAY DATA FLOW



REACT

EVENTS

event props of React element

```
// ReactElement
<button onClick={ this.handleClick }>Click!</button>
<input type="text" defaultValue="" onBlur={ this.handleBlur } />

// HTML DOM element
<button onclick="handle_click();">Click!</button>
<input type="text" value="" onblur="handle_blur();" />
```

event attributes of HTML DOM element

REACT
EVENTS

```
const App = () => {
  return <Button name="Save" />;
};


const Button = ({ name }) => (
  <button onClick={() => alert("Click Button”)}>
    {name}
  </button>
);
```

REACT
EVENTS

```
const onButtonClick = () => alert("Click Button");

const App = () => {
  return <Button name="Save" onClick={onButtonClick}/>;
};

const Button = ({ name, onClick }) => (
  <button onClick={onClick}>{name}</button>
);
```

PROPTYPES

PROPTYPES

```
import PropTypes from "prop-types";

const Hello = ({ name }) => <h1>Hello, {name}</h1>;

Hello.propTypes = {
  name: PropTypes.string
};

Hello.defaultProps = {
  name: 'unknown'
};
```

PROPTYPES

```
22:52:11.651 ▶Warning: Failed prop type: Invalid prop `name` of type `array`
              supplied to `Hello`, expected `string`.
                  in Hello (at App.js:6)
                  in App (at src/index.js:7)
```

PROPTYPES

https://github.com/facebook/prop-types

CONTEXT API

CONTEXT

```
const App = () => (
  <Wrapper lang="pl">
    <Content lang={props.lang}>
      <BlogPosts lang={props.lang}>
        <Post lang={props.lang} />
      </BlogPosts>
    </Content>
    <Sidebar lang={props.lang}>
      <SomeComponent lang={props.lang}>
        <NewsletterForm lang={props.lang} />
      </SomeComponent>
    </Sidebar>
  </Wrapper>
);
```

CONTEXT

```
const Context = React.createContext();
```

```
<Context.Provider />
```

```
<Context.Consumer />
```

<PROVIDER>

<CONSUMER>

```jsx
const LanguageContext = React.createContext();

const App = () => (
  <LanguageContext.Provider value="pl">
    <Wrapper>
      <Content>
        <BlogPosts>
          <LanguageContext.Consumer>
            {lang => <Post lang={lang} />}
          </LanguageContext.Consumer>
        </BlogPosts>
      </Content>
      <Sidebar>
        <SomeComponent>
          <LanguageContext.Consumer>
            {lang => <NewsletterForm lang={lang} />}
          </LanguageContext.Consumer>
        </SomeComponent>
      </Sidebar>
    </Wrapper>
  </LanguageContext.Provider>
);
```

THANKS

REACT JS

HOC

HIGH ORDER COMPONENT

HOC

```
component => componentOnSteroids;
```

HOC

```
const EnhancedComponent = higherOrderComponent(WrappedComponent);
```

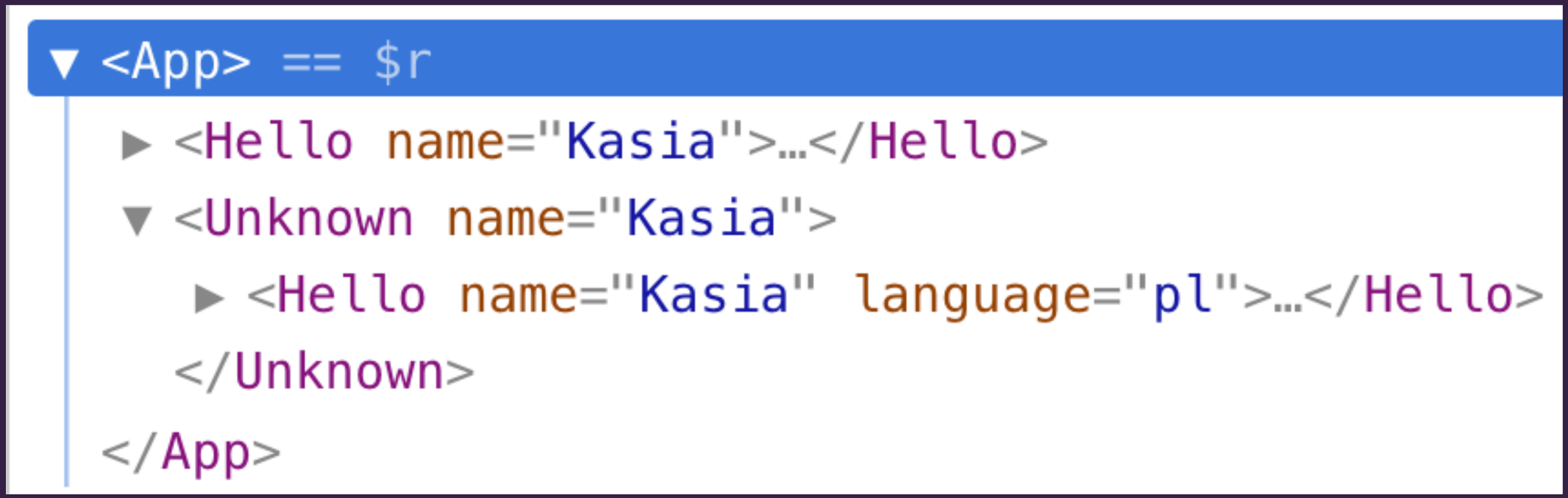
HOC

```
const withLanguage = WrappedComponent => {
  return props => (
    <WrappedComponent {...props} language="pl" />
  );
};
```

```
const HelloWithLang = withLanguage(Hello);

const App = () => (
  <>
    <Hello name="Kasia" />
    <HelloWithLang name="Kasia" />
  </>
);
```

HOC


▼ <App> == $r
▶ <Hello name="Kasia">…</Hello>
▼ <Unknown name="Kasia">
▶ <Hello name="Kasia" language="pl">…</Hello>
</Unknown>
</App>

HOC

DON’T MUTATE THE ORIGINAL COMPONENT.

USE COMPOSITION.

HOC

```
function logProps(InputComponent) {
    InputComponent.prototype.componentWillReceiveProps = function(nextProps) {
        console.log('Current props: ', this.props);
        console.log('Next props: ', nextProps);
    };
    return InputComponent;
}

const EnhancedComponent = logProps(InputComponent);
```

HOC

```
function logProps(WrappedComponent) {
    return class extends React.Component {
        componentWillReceiveProps(nextProps) {
            console.log('Current props: ', this.props);
            console.log('Next props: ', nextProps);
        }
        render() {
            return <WrappedComponent {...this.props} />;
        }
    }
}
```

HOC

```
const EnhancedComponent = hoc(Component);

const EnhancedComponent = anotherHoc(Component, config);
```

```
const ConnectedComponent = connect(param1, param2)(Component);
const enhance = connect(param1, param2);
const ConnectedComponent = enhance(Component);
```

HOC

```
const EnhancedComponent = withRouter(connect(param)(WrappedComponent))
```

```
const enhance = compose(
    withRouter,
    connect(commentSelector)
)
const EnhancedComponent = enhance(WrappedComponent)
```

HOC

```
@withRouter
@connect(param)
export default class MyFancyComponent extends React.Component {

}
```

HOC

react-fns

https://github.com/jaredpalmer/react-fns

CLASS COMPONENT

CLASS COMPONENT

```
const Welcome = props => {
    return <h1>Hello, {props.name}</h1>;
}
```

```
class Welcome extends React.Component {
    render() {
        return <h1>Hello, {this.props.name}</h1>;
    }
}
```

CLASS COMPONENT

„When should I use a function and when a class?"

~undefined

CLASS COMPONENT DIFFERENCES

CLASS COMPONENT SYNTAX

CLASS COMPONENT DIFFERENCES

```
var Welcome = function Welcome(props) {
  return React.createElement(
        "h1",
        null,
        "Hello, ",
        props.name);
};
```

CLASS COMPONENT DIFFERENCES

```
var Welcome = (function(_React$Component) {
    _inherits(Welcome, _React$Component);

    function Welcome() {
        _classCallCheck(this, Welcome);

        return _possibleConstructorReturn(
            this,
            (Welcome.__proto__ || Object.getPrototypeOf(Welcome)).apply(
                this,
                arguments
            )
        );
    }

    _createClass(Welcome, [
        {
            key: "render",
            value: function render() {
                return React.createElement("h1", null, "Hello, ", this.props.name);
            }
        }
    ]);

    return Welcome;
})(React.Component);
```

CLASS COMPONENT STATE

CLASS COMPONENT STATE

STATE VS PROPS

CLASS
COMPONENT
STATE

COMMON

plain JS objects

render update

deterministic

CLASS
COMPONENT
STATE

PROPS

Component's configuration

Received from above

Immutable

CLASS
COMPONENT
STATE

STATE

Component's information

Created in component

Changeable

CLASS COMPONENT STATE

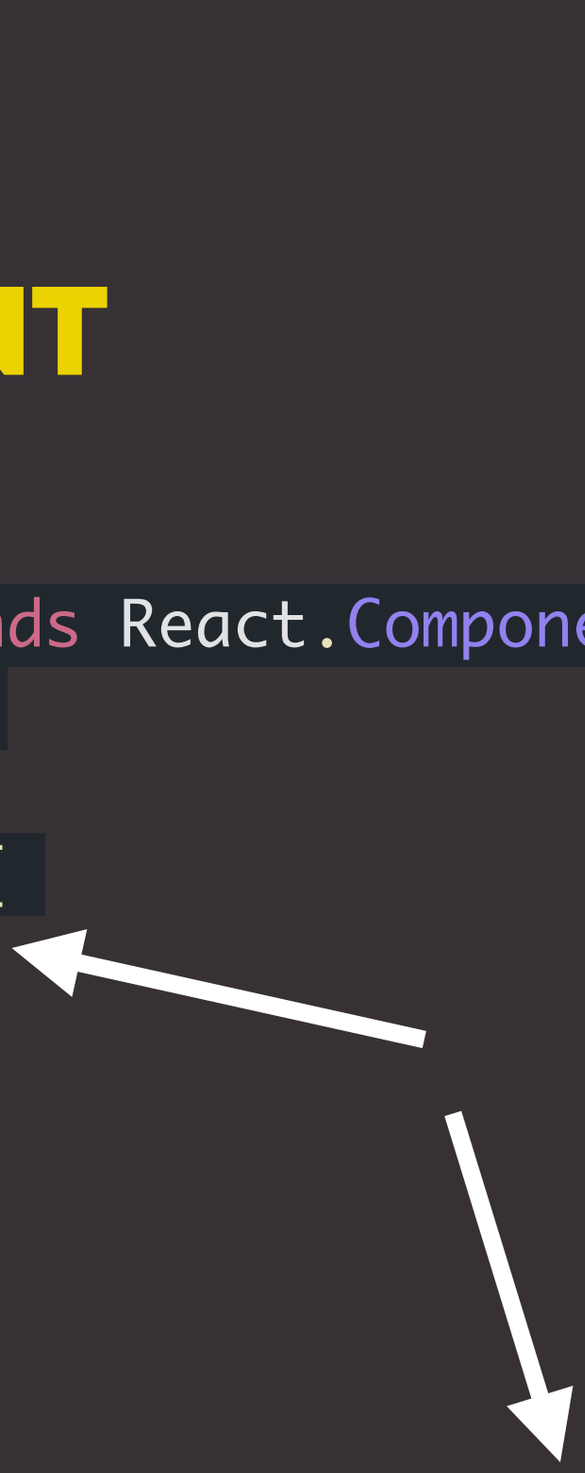
```
class Button extends React.Component {
    constructor() {
        super();
        this.state = {
            count: 0,
        };
    }

    render() {
      ……
    }
}
```

CLASS COMPONENT STATE

```
class Button extends React.Component {
  constructor() {
    super();
    this.state = {
      count: 0
    };
  }

  render() {
    return (
        <button>
            Clicked {this.state.count} times
        </button>
    );
  }
}
```



```
class Button extends React.Component {
  constructor() {
    super();
    this.state = {
      count: 0
    };
    this.updateCount = this.updateCount.bind(this)
  }

  updateCount() {
    this.setState((prevState, props) => {
      return { count: prevState.count + 1 };
    });
  }

  render() {
    return (
      <button onClick={this.updateCount}>
        Clicked {this.state.count} times
      </button>
    );
  }
}
```

DEMO

INITIALIZE STATE

```
this.state = {}
```

CHANGE STATE

```
this.setState()
```

setState()

```
this.setState({ count: 2 });
```

```
this.setState((state) => {
    return {count: state.count + 1};
});
```

setState()

ASYNCHRONOUS

```
constructor() {
    super();
    this.state = {
        count: 0;
    }
}

incrementCount() {
    this.setState({count: this.state.count + 1});
}

handleSomething() {
    this.incrementCount();
    this.incrementCount();
    this.incrementCount();
}
```

```
constructor() {
    super();
    this.state = {
        count: 0;
    }
}

incrementCount() {
    this.setState((state) => {
        return {count: state.count + 1}
    });
}

handleSomething() {
    this.incrementCount();
    this.incrementCount();
    this.incrementCount();
}
```

setState()

TRIGGER RENDER

STATE
ANTI-PATTENR

```
class MyCompo          exten          mponent {
    constructo          ps
        super(p
        this.sta
            som                    rops.someValue,
        };
    }
}
```

„When should I use a function and when a class?"

~undefined

REACT HOOKS

```jsx
import { useState } from 'react';

const Example = () => {
    // Declare a new state variable, which we'll call "count"
    const [count, setCount] = useState(0);

    return (
        <div>
            <p>You clicked {count} times</p>
            <button onClick={() => setCount(count + 1)}>
                Click me
            </button>
        </div>
    );
}
```

CLASS COMPONENT DIFFERENCES

LIFECYCLE HOOKS

LIFECYCLE HOOKS

MOUNTING

UPDATING

UNMOUNTING

“Faza Render”

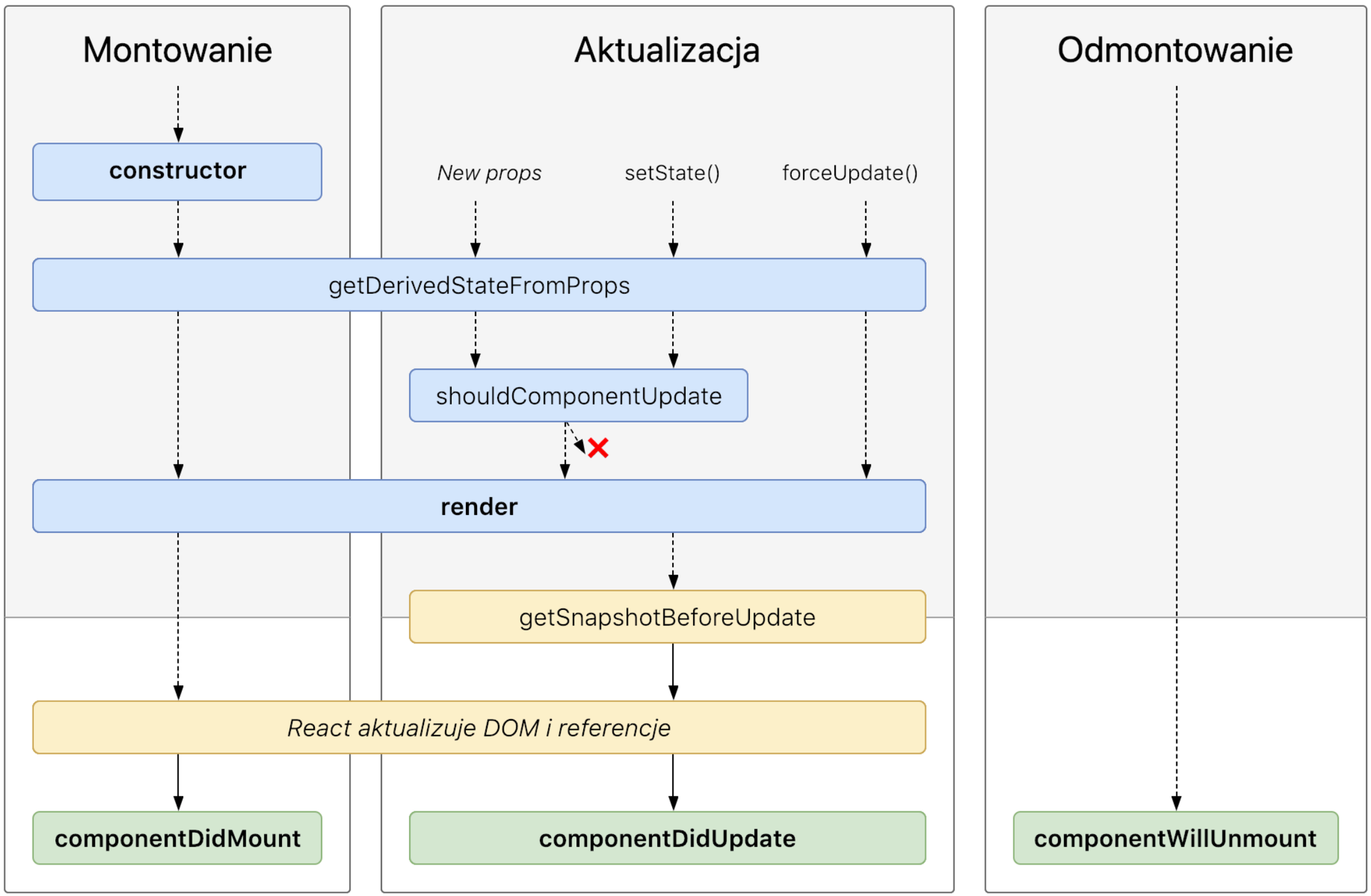
Czysta i bez efektów ubocznych. Może zostać wstrzymana, anulowana lub zre-startowana przez React.

“Faza Pre-commit”

Może czytać DOM.

“Faza Commit”

Może operować na DOM, tworzyć efekty uboczne i kolejkować aktualizacje.



constructor()

initialize state (this.state)

bind methods

static getDerivedStateFromProps()

return an object to update
the state

fired on every render

render()

required

componentDidMount()

load data from a remote endpoint

side-effect

DOM

shouldComponentUpdate()

performance optimization

getSnapshotBeforeUpdate()

componentDidUpdate()

invoked immediately after updating

good to do network requests

componentWillUnmount()

invoked immediately before a component is unmounted and destroyed

cleaning up

BIND

```
class Foo extends Component {
  handleClick() {
    console.log("Click happened", this.state);
  }
  render() {
    return <button onClick={this.handleClick}>Click Me</button>;
  }
}
```

BIND

```
class Foo extends Component {
    constructor(props) {
        super(props);
        this.handleClick = this.handleClick.bind(this);
    }
    handleClick() {
        console.log('Click happened', this.state);
    }
    render() {
        return <button onClick={this.handleClick}>Click Me</button>;
    }
}
```

BIND

```
class Foo extends Component {
  handleClick = () => {
    console.log("Click happened", this.state);
  }
  render() {
    return <button onClick={this.handleClick}>Click Me</button>;
  }
}
```

BIND

```
class Foo extends Component {
  handleClick() {
    console.log("Click happened", this.state);
  };
  render() {
    return (
      <button onClick={this.handleClick.bind(this)}>
         Click Me
      </button>
    );
  }
}
```

BIND

```
class Foo extends Component {
  handleClick() {
    console.log("Click happened", this.state);
  }
  render() {
    return (
      <button onClick={() => this.handleClick()}>
        Click Me
      </button>
    );
  }
}
```

THANKS