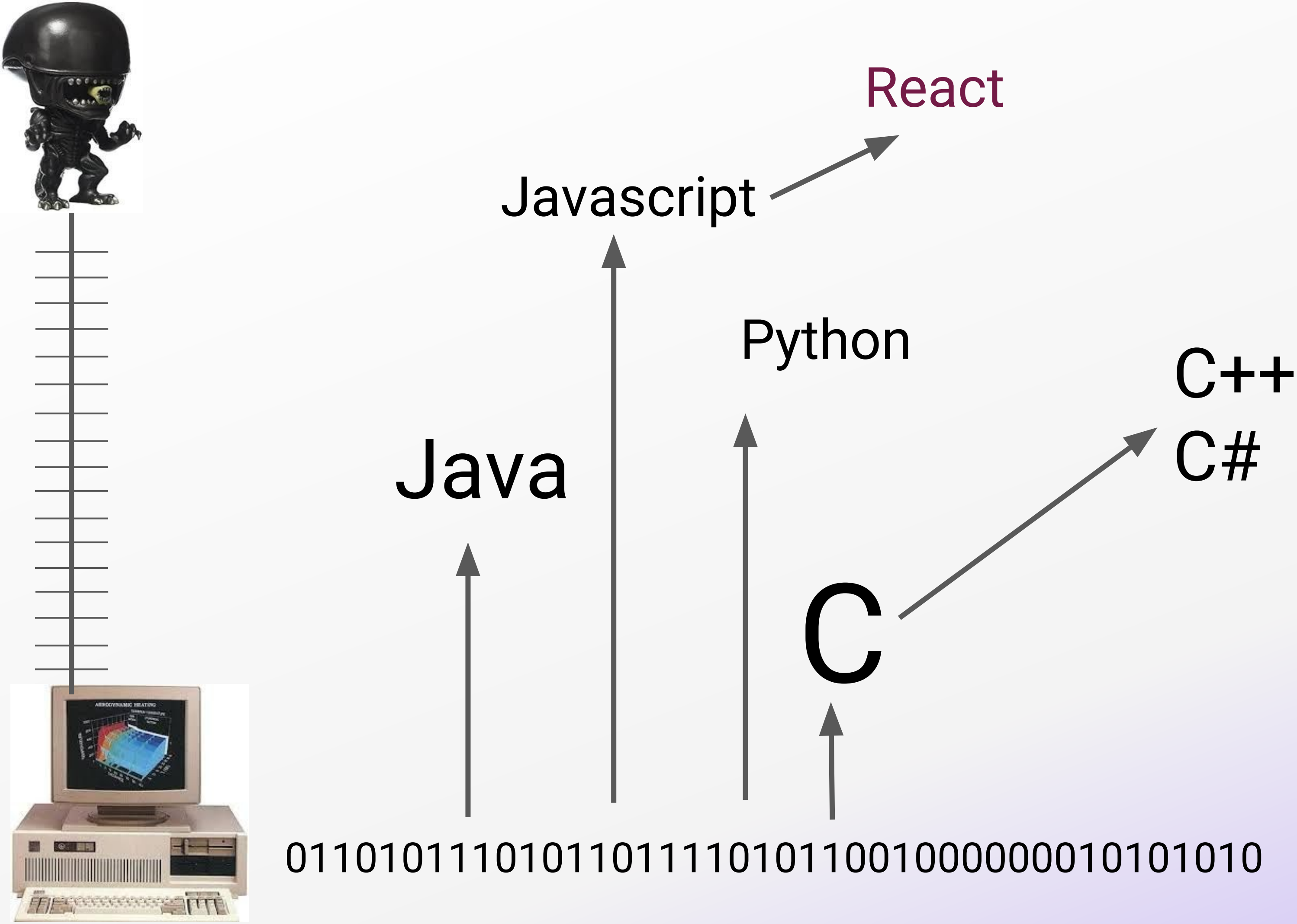
t4tech

t4tech

https://t4tech-nyc.github.io

https://github.com/t4tech-nyc



React

Javascript

Python

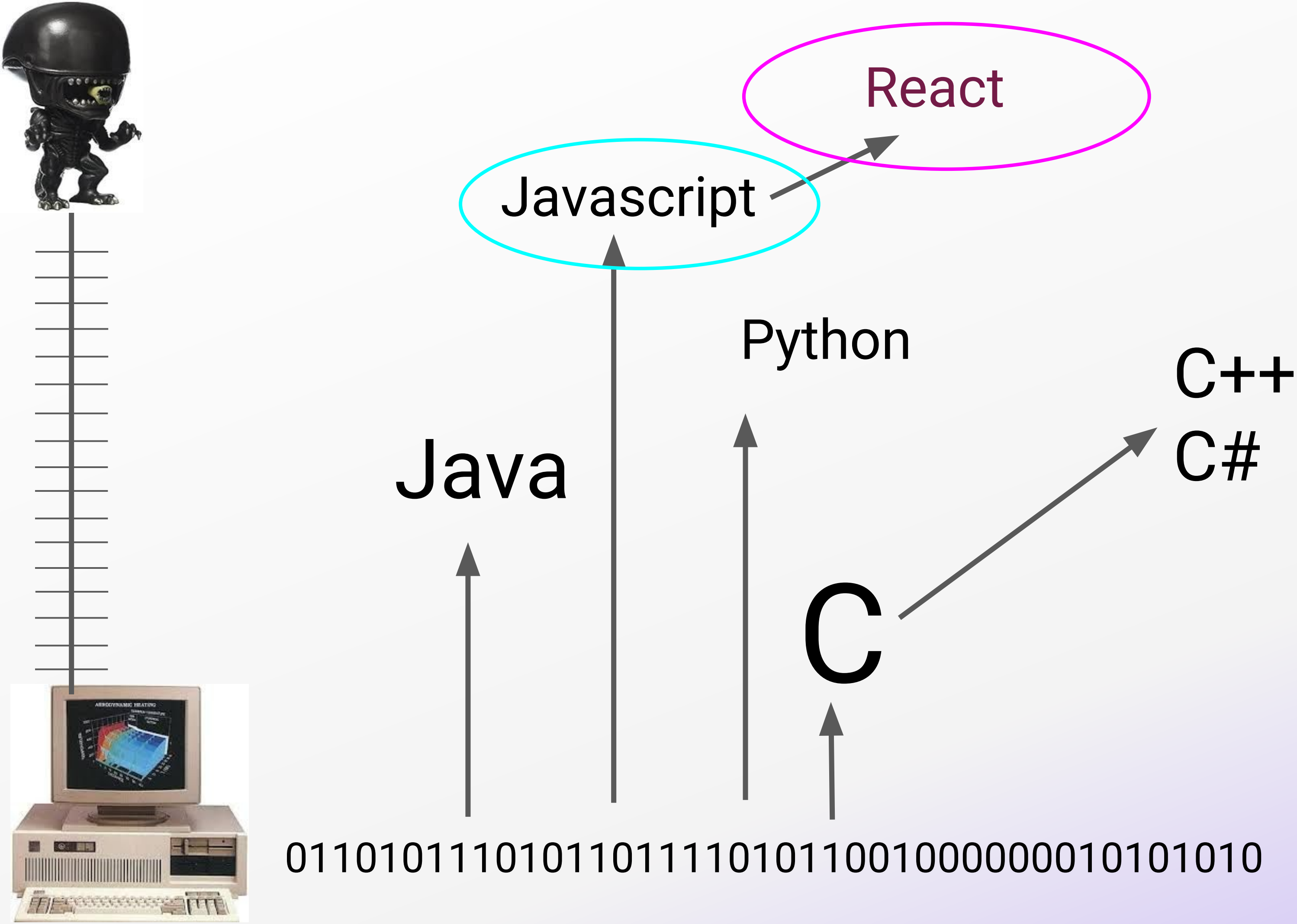
C++

C#

Java

C

011010111010110111101011001000000010101010



React

Javascript

Python

C++

C#

Java

C

0110101110101101111010110010000000010101010

React.js

React.js

Javascript library for front end development

React.js

Javascript library for front end development

Useful for state management

React.js

Javascript library for front end development

Useful for state management

Promotes modularity

React.js

Javascript library for front end development

Useful for state management

Promotes modularity

One way data flow

React.js

Javascript library for front end development

Useful for state management

Promotes modularity

One way data flow

React.js

State =

Modularity =

One Way Data Flow =

React.js

State = The given truth of an application *at that moment in time*

Modularity =

One Way Data Flow =

React.js

State = The given truth of an application *at that moment in time*

Modularity = Reusable, flexible, bite-size building blocks

One Way Data Flow =

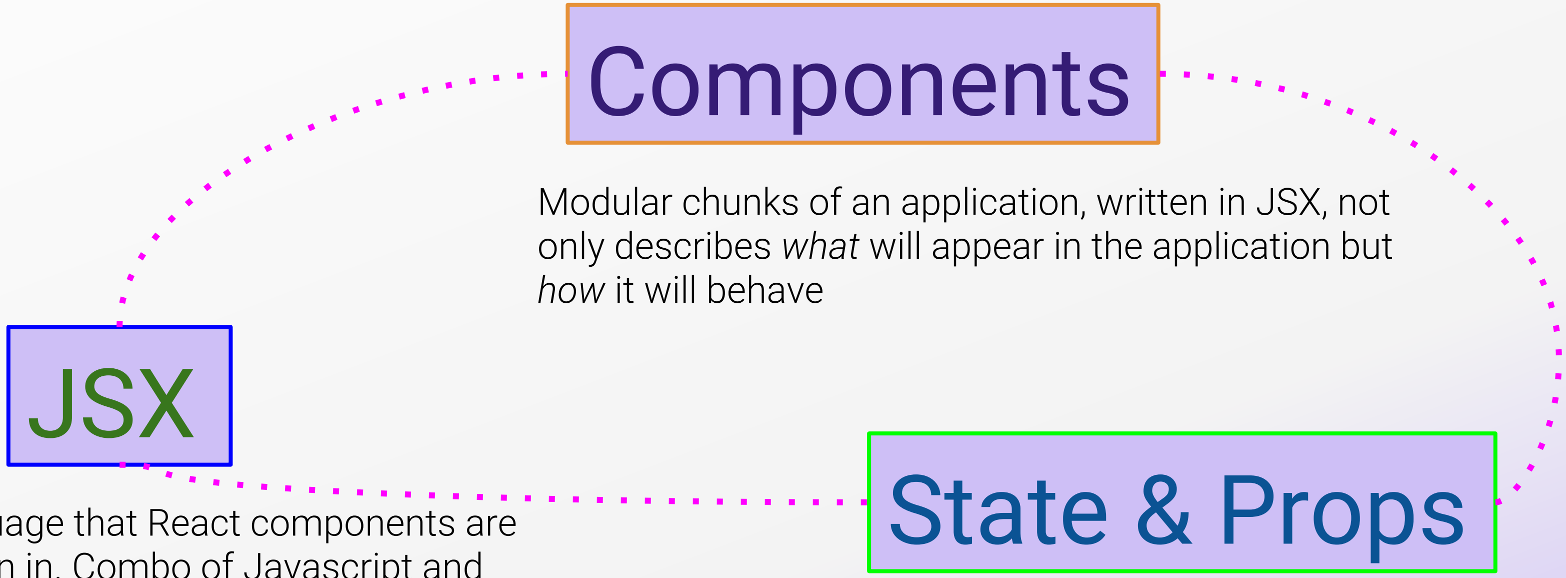
React.js

State = The given truth of an application *at that moment in time*

Modularity = Reusable, flexible, bite-size building blocks

One Way Data Flow = Data flows from the top down, things are inherited hierarchically

React.js



Components

Modular chunks of an application, written in JSX, not only describes *what* will appear in the application but *how* it will behave

JSX

Language that React components are written in. Combo of Javascript and HTML (XML)

State & Props

Built in vaults within components used to store, generate, and pass along data

JSX

HTML + Javascript

JSX

```
<html>
    <body>
        <h1 id=”titleHeader”>
            my cool ass webpage
        </h1>

        <p id=”favePokemonText”>
            My favorite pokemon is
                <span id=”fave”>
                    Pikachu
                </span>
        </p>
    </body>
</html>
```

JSX

```
<html>
    <body>
        <h1 id=”titleHeader”>
            my cool ass webpage
        </h1>

        <p id=”favePokemonText”>
            My favorite pokemon is
                <span id=”fave”>
                    Pikachu
                </span>
        </p>
    </body>
</html>
```

```
let myFavePokemon = ‘Ninetails’

document.getElementById(‘fave’)
    .innerHTML = myFavePokemon
```

JSX

```
<html>
    <body>
        <h1 id=”titleHeader”>
            my cool ass webpage
        </h1>

        <p id=”favePokemonText”>
            My favorite pokemon is
                <span id=”fave”>
                    Pikachu
                </span>
        </p>
    </body>
</html>
```

```
let myFavePokemon = ‘Ninetails’

document.getElementById(‘fave’)
    .onClick(changePokemon)

let changePokemon = function() {
    document.getElementById(‘fave’)
        .innerHTML = myFavePokemon
    }
```

JSX

```
<h1 id=”titleHeader”>
    my cool ass webpage
</h1>

<p id=”favePokemonText”>
    My favorite pokemon is {myFavePokemon}
</p>
```

Components

Components

Written in JSX

Components

Written in JSX

Expects a “render” method

Components

Written in JSX

Expects a “render” method

May contain other components (inheritance)

Components

Written in JSX

Expects a “render” method

May contain other components (inheritance)

Possesses “lifecycle methods”

Components

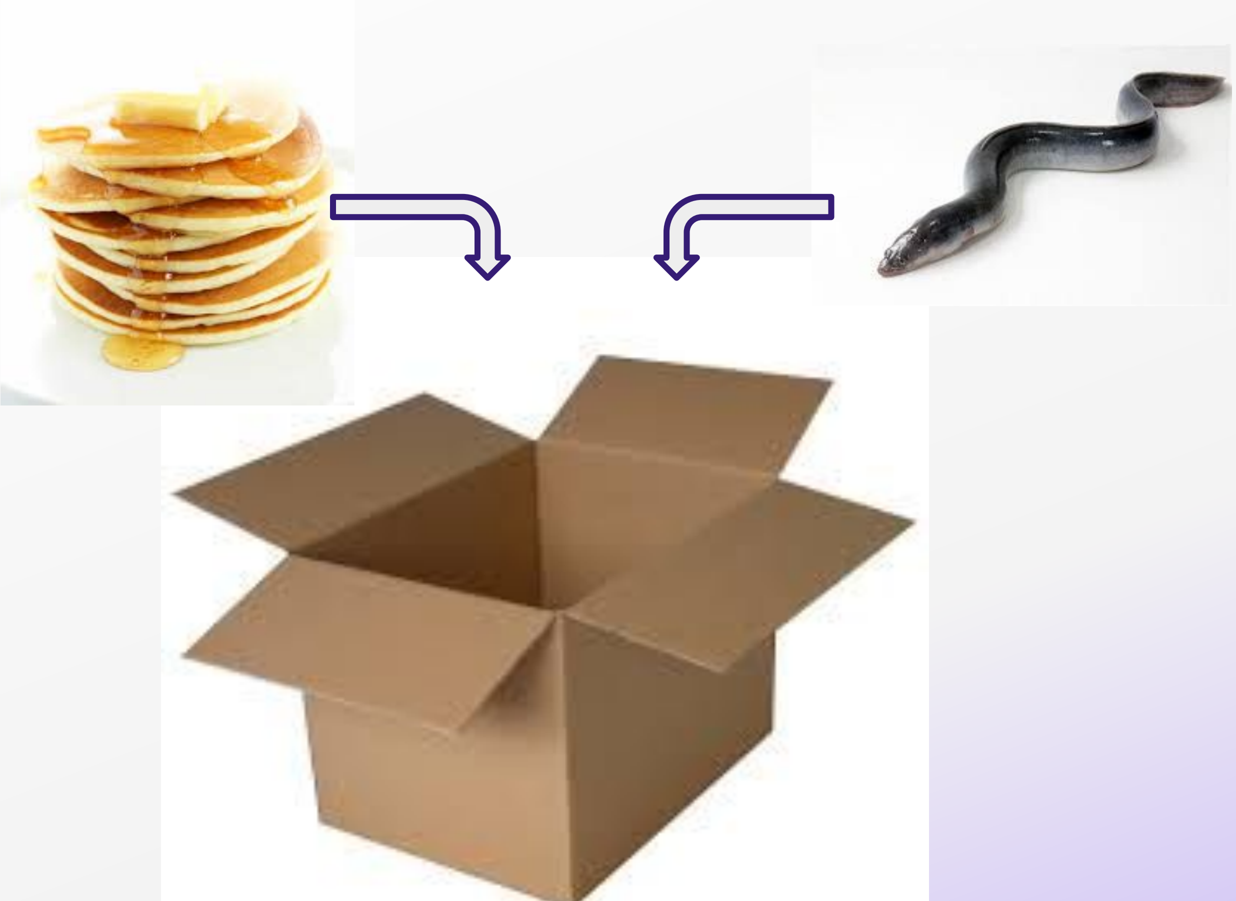
```
export default class MyComponent extends React.Component() {
        constructor(props) {
                super(props)

                This.state = { ...some stuff... }
        }

        render() {
                return(
                        <div> { ...other stuff ... } </div>
                )
        }
}
```

Props & State

Vaults baked into components made to store and pass along data

Props & State

Props & State

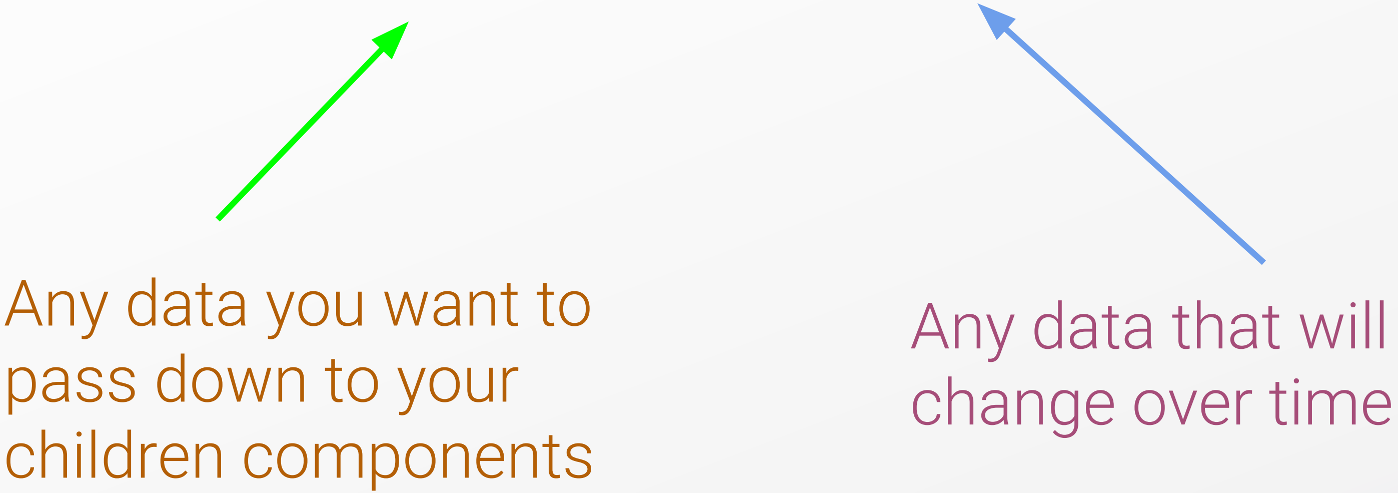


Props & State

Any data you want to pass down to your children components

Any data that will change over time

Props & State



Any data you want to pass down to your children components

Any data that will change over time

Props & State

```
export default class MyComponent extends React.Component() {
    constructor(props) {
        super(props)

        this.state = {
            myFavePokemon: ‘Ninetails’
        }
    }

    render() {
        return(
            <div>
                <h1> {this.state.myFavePokemon} </h1>
                <AnotherComponent favePoke={this.state.myFavePokemon} />
            </div>
        )
    }
}
```

That is it…

That is it...

Oh wait one last thing

Virtual Dom

hehe….

Virtual Dom

It’s complicated