

# TypeScript in React: *A Love Story* ❤️



Kylie Stewart, Engineer at Formidable



TypeScript

Developer



```
yarn create react-app your-app --  
  typescript
```

What is  
TypeScript,  
exactly?

Statically  
Typed

*Java, TypeScript,  
Swift*

Dynamically  
Typed

*PHP, JavaScript,  
Python*

# Compared to JavaScript

```
class Button extends Component {
  render() {
    return (
      <button {...this.props}>{children}</button>
    );
  }
}

Button.defaultProps = {
  href: '',
  disabled: false
};

export default Button;
```

```
interface IButtonProps {
  href: string;
  disabled?: boolean;
  size?: string;
}

class Button = React.Component<IButtonProps> => {
  public static defaultProps: Partial<IButtonProps> = {
    href: '',
    disabled: false
  };

  public render() {
    return (
      <button {...this.props}>{children}</button>
    );
  }
};

export default Button;
```

**What else?**

# Interfaces

```
interface IButtonProps {  
  href: string;  
  disabled?: boolean;  
  size?: string;  
}  
  
class Button = React.Component<IButtonProps> => {  
  public static defaultProps: Partial<IButtonProps> = {  
    href: '',  
    disabled: false  
  };  
  
  public render() {  
    return (  
      <button {...this.props}>{children}</button>  
    );  
  }  
};  
  
export default Button;
```



# Unions

```
interface IButtonProps {  
  href: string;  
  disabled?: boolean;  
  size?: string;  
  message?: string | string[];  
}
```

# Enums

```
enum ButtonSize = {  
    Small,    // 0  
    Medium,   // 1  
    Large     // 2  
}  
  
interface IButtonProps {  
    href: string;  
    disabled?: boolean;  
    size?: ButtonSize;  
}
```

any

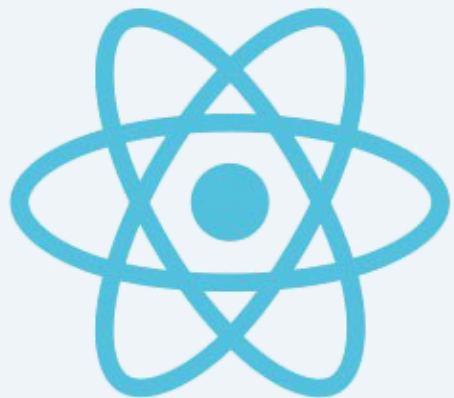


# Object Literal Checks

```
function logMessage(anything: { message: string }) {  
  console.log(anything.message);  
}  
  
const first = { message: "Hi React Denver", date: "2019-04-16T22:27:57.510Z" };  
const second = { message: "I like cheese", priority: "high" };  
const third = { note: "No message here" };  
  
logMessage(first);      // 👍  
logMessage(second);    // 👍  
logMessage(third);     // 🛑 Error: property 'message' is missing
```

# Allowing Extraneous Props

```
let someObj: {  
  mustBeString: string;  
  [index: string]?: any;  
};  
  
myObject = { mustBeString: 'hello', whatever: true }; // 👍  
myObject = { mustBeString: 1, extra: 'a string' }; // 🚫 Error: property 'mustBeString' must be type string
```



 Create more portable code

 Write better documentation

 Easy refactoring

 Increase dev productivity

# TypeScript Errors

`noImplicitAny`

`strictNullChecks`



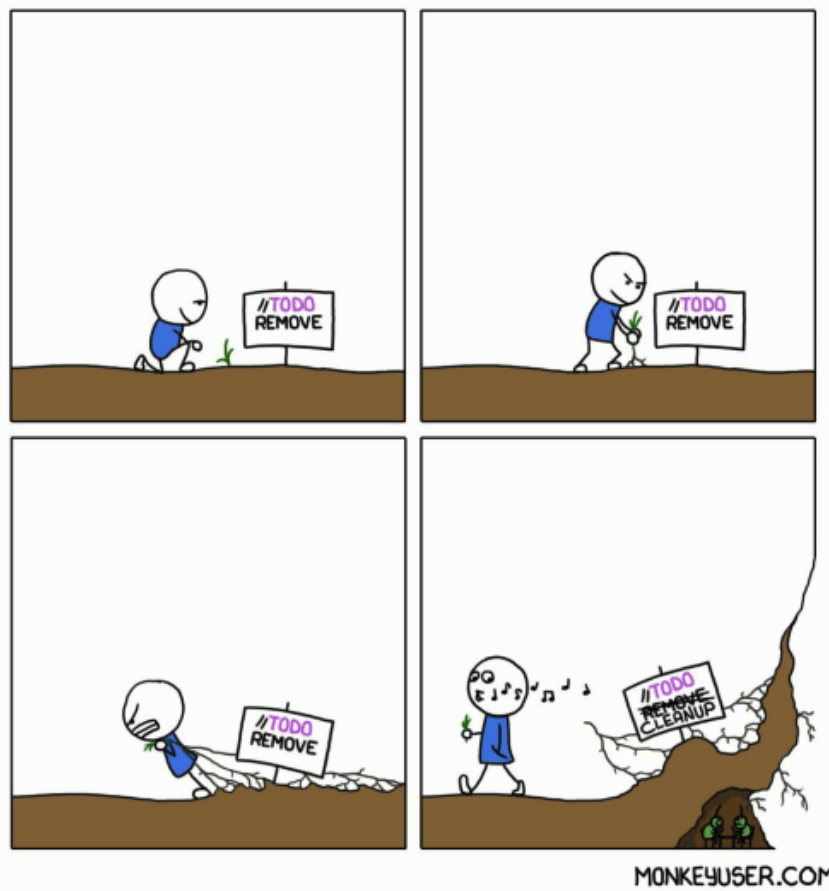
# Write Better Component APIs

# PropTypes vs. Interfaces

```
class Button extends Component {  
  render() {  
    return (  
      <button {...this.props}>{children}</button>  
    );  
  }  
}  
  
Button.defaultProps = {  
  href: '',  
  disabled: false  
};  
  
Button.propTypes = {  
  children: PropTypes.node,  
  href: PropTypes.string.isRequired,  
  disabled: PropTypes.boolean,  
  size: PropTypes.string  
};  
  
export default Button;
```

```
interface IButtonProps {  
  href: string;  
  disabled?: boolean;  
  size?: string;  
}  
  
class Button = React.Component<IButtonProps> => {  
  public static defaultProps: Partial<IButtonProps> = {  
    href: '',  
    disabled: false  
  };  
  
  public render() {  
    return (  
      <button {...this.props}>{children}</button>  
    );  
  }  
};  
  
export default Button;
```

# Composing Better Props and State



# Extending Interfaces

```
interface IWithUserDataProps {  
  id: number;  
  age: number;  
  firstName: string;  
  lastName: string;  
  nickname?: string;  
}
```

```
interface IGreetingProps extends IWithUserDataProps {  
  message: string;  
}
```

# Interfaces x TDD

✓ Create more portable code

✓ Write better documentation

✓ Easy refactoring

? Increase dev speed

```
6 |  
7  var server = express();  
8  server.use(bodyParser.json());  
9  |
```

```
10 server|  
11 |  
12 |  
13 |  
14 |  
15 |  
16 |  
17 |  
18 |  
19 |  
20 |
```





Reads 3rd party  
declaration files

Gives you cleaner  
errors

Helps you refactor



**Sebastian McKenzie** ✓

@sebmck



TypeScript is the Clippy of JavaScript



Property 'toLowerCase' does not exist on type 'string'. Did you mean 'toLowerCase'?

any

```
let s
```

```
const x = s.toLowerCase();
```

*Try it for yourself!*

<https://typescriptcourses.com/typescript-fundamentals>

<https://www.typescriptlang.org/play>

<https://basarat.gitbooks.io/typescript>

# THANK YOU!

@KYLIESTEW ON TWITTER

[github.com/kale-stew/ts-in-react](https://github.com/kale-stew/ts-in-react)