The React Props Cheatsheet

React Sandbox: https://codesandbox.io/s/practical-pond-xotyg?file=/src/App.js

1. Props can be passed conditionally

```
export default function App() {
  return (
    <MyComponent />
  );
}

function MyComponent(props) {
  return (
    // shows Hello, doesn’t throw error
   // despite title prop not being passed
    <h1>Hello {props.title}!</h1>
  )
}
```

2. Props passed with just their name have a value of true

```
export default function App() {
  return (
    // showTitle passes an implicit value of true
    // just as if we set it to showTitle={true}
    <MyComponent showTitle />
  );
}

function MyComponent(props) {
  if (props.showTitle) {
    // displays "Hello React"
    return <h1>Hello React</h1>;
  }

  return null;
}
```

3. Props can be accessed as an object or destructured

```
export default function App() {
  return (
    <MyComponent
      showTitle
      title="Hello React"
    />
  );
}

function MyComponent({ showTitle, title }) {
  // can use object destructuring, like we have here
  // or you can access each property from an object
  // usually called "props"
  if (showTitle) {
    return <h1>{title}</h1>;
  }

  return null;
}
```

4. Components can be passed as props (including children)

```
export default function App() {
  return (
    <MyComponent showTitle>
      <h2>Hello React</h2>
    </MyComponent>
  );
}

function MyComponent({ showTitle, children }) {
  if (showTitle) {
    return children;
  }

  return null;
}
```

5. Anything can be passed as a prop (especially functions)

```
import React from 'react'

export default function App() {
  const [title, setTitle] = React.useState('Hello React')

  return (
    <MyComponent
      title={title}
      setTitle={setTitle}
    />
  );
}

function MyComponent({ title, setTitle }) {
   // we are receiving title as a prop value
  // but are able to update it dynamically using state
  function updateTitle() {
    setTitle("Hello world!")
  }

  return <h2 onClick={updateTitle}>{title}</h2>
}
```

6. Update a prop's value with state

```
import React from 'react'

export default function App() {
  return (
    <MyComponent
      title="My Title To Change"
    />
  );
}

function MyComponent({ title }) {
  // in this example, we are receiving a static value from props
  // but changing it to a dynamic one using state
  const [newTitle, setNewTitle] = React.useState(title)

  function updateNewTitle() {
    setNewTitle("Something else")
  }

  return <h2 onClick={updateNewTitle}>{newTitle}</h2>
}
```

7. Props can be spread in individually

```
import React from 'react'

export default function App() {
  const data = {
    title: "my title",
    description: "my description",
    button: "see more"
  }

  return (
    <MyComponent {...data} />
  );
}

function MyComponent({ title, description, button }) {
  return (
    <div>
      <h1>{title}</h1>
      <p>{description}</p>
      <button>{button}</button>
    </div>
  )
}
```

8. Props can be given a default value if none is provided

```
import React from 'react'

export default function App() {
  const data = {
    description: "my description",
    button: "see more"
  }

  return (
    <MyComponent {...data} />
  );
}

function MyComponent({ title = "My Default Title", description, button }) {
  return (
    <div>
      <h1>{title}</h1>
      <p>{description}</p>
      <button>{button}</button>
    </div>
  )
}
```

9. Props can be renamed to avoid errors

```
import React from 'react'

export default function App() {
  const data = {
    title: "my title",
    description: "my description",
    button: "see more"
  }

  return (
    <MyComponent {...data} />
  );
}

function MyComponent({ title, description, button: buttonText }) {
  const button = <button>like this post</button>

  return (
    <div>
      <h1>{title}</h1>
      <p>{description}</p>
      <button>{buttonText}</button>
      {button}
    </div>
  )
}
```

10. Don't attempt to destructure props multiple times

```
import React from 'react'

export default function App() {
  const data = {
    title: "my title",
    description: "my description",
    // button: {
```

```
    //   text: "see more"
    // }
  }

  return (
    <MyComponent {...data} />
  );
}

function MyComponent({ title, description, button: { text } }) {
  const button = <button>like this post</button>

  // our app breaks if we try to destructure the property text
  // from an object that doesn't exist "button" on data
  return (
    <div>
      <h1>{title}</h1>
      <p>{description}</p>
      <button>{text}</button>
      {button}
    </div>
  )
}
```