REACT ROUTER 201

Declarative routing for React

TRAJECTORY

- **Route props**
  - match
  - history
  - location

- **How to pass our own props to the components given to <Route/>**

```
<Route path='/puppies' component={Puppies} />
```

WHAT PROPS DOES THAT COMPONENT RECEIVE?

THE ROUTE PROPS

- **match**
- **history**
- **location**

match

- Contains info about how a Route’s *path* matched the current url
- Contains an especially useful *params* key that parses url parameters

```
<Route path=‘                    ‘ component={     } />
```

```
const Puppy = (props) => {
  return <div>{                              }</div>
}
```

```
<Route path='/puppies/:puppyId' component={    } />
```

```
const Puppy = (props) => {
  return <div>{                    }</div>
}
```

```
<Route path='/puppies/:puppyId' component={Puppy} />
```

```
const Puppy = (props) => {
  return <div>{                         }</div>
}
```

```
<Route path='/puppies/:puppyId' component={Puppy} />
```

```
const Puppy = (props) => {
  return <div>{props.match.params.puppyId}</div>
}
```

```
<Route path='/puppies/:puppyId' component={Puppy} />
```

```
const Puppy = (props) => {
  return <div>{props.match.params.puppyId}</div>
}
```

```
http://localhost:3000/#/puppies/
1
```

props.match.params.puppyId

1

```
<Route path='/puppies/:puppyId' component={Puppy} />
```

```
const Puppy = (props) => {
  return <div>{props.match.params.puppyId}</div>
}
```

http://localhost:3000/#/puppies/2

props.match.params.puppyId

2

history

- Used to manipulate browser's history programmatically
  - You can programmatically navigate using `history.push`
- Other methods for dealing with history (goBack, goForward)

```
const Puppy = () => {
  return (
    <button onClick={               }>
      Go Elsewhere
    </button>
  )
}
```

```
const Puppy = () => {
  handleClick = () => {
  }
  return (
    <button onClick={handleClick}>
      Go Elsewhere
    </button>
  )
}
```

```
const Puppy = (props) => {
  handleClick = () => {
    console.log('You clicked!')
    props.history.push('/elsewhere')
  }
  return (
    <button onClick={handleClick}>
      Go Elsewhere
    </button>
  )
}
```

location

- Contains information about where the url is now

- Useful in combination with some lifecycle hooks for telling whether or not navigation happened or not (we'll see examples of this later)

```
<Route path='/puppies' component={Puppies} />
```

HOW DO WE PASS PROPS TO THAT COMPONENT?

```
<Route path='/puppies' component={Puppies} />
```

```
<Route path='/puppies' render={() => <Puppies />} />
```

```
const MainPage = () => {
 return (
  <Router>
   <div>
    <Route path='/puppies'/>
   </div>
  </Router>
 )
}
```

```
const MainPage = () => {
  const [user, setUser] = useState({});
  return (
    <Router>
      <div>
        <Route path='/puppies'/>
      </div>
    </Router>
  )
}
```

```
const MainPage = () => {
  const [user, setUser] = useState({});
  return (
    <Router>
      <div>
        <Route
            path='/puppies'
            render={() => <Puppies />}
        />
      </div>
    </Router>
  )
}
```

```
const MainPage = () => {
  const [user, setUser] = useState({});
  return (
    <Router>
      <div>
        <Route
            path='/puppies'
            render={() => <Puppies user={user} />}
        />
      </div>
    </Router>
  )
}
```

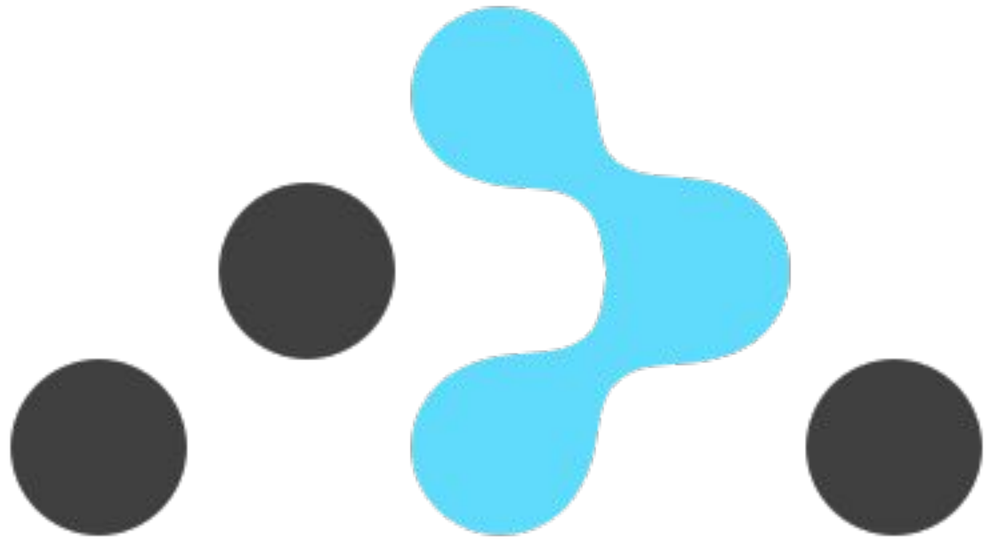
```
<Route path='/puppies' render={() => <Puppies />} />
```

```
<Route
  path='/puppies'
  render={routeProps => <Puppies {...routeProps} />}
/>
```

SUMMARY

- **React Router passes three props to your components**
  - match
  - history
  - location

- **You can pass your own props using render instead of component in a <Route/>**



REACT ROUTER