

# HIGHER ORDER COMPONENTS

---

*So hiii-i-i-igh...*

**HIGHER ORDER COMPONENT: A FUNCTION  
THAT TAKES A COMPONENT AS AN  
ARGUMENT, AND RETURNS A NEW  
COMPONENT**

# HIGHER ORDER COMPONENTS

- **React.Children:** makes it easy to share a containing view (“picture frame”) between components
- **Higher order components:** makes it easy to share *behavior* between components
- **Very similar to a “render prop”**

# HIGHER ORDER FUNCTIONS

```
const logOnce = () => {  
  
}
```

# HIGHER ORDER FUNCTIONS

```
const logOnce = () => {  
  console.log('once')  
}
```

# HIGHER ORDER FUNCTIONS

```
const logOnce = () => {  
  console.log('once')  
}
```

```
const addMoreLogging = (otherFunc) => {  
  
  
  
  
  
  
}
```

# HIGHER ORDER FUNCTIONS

```
const logOnce = () => {  
  console.log('once')  
}
```

```
const addMoreLogging = (otherFunc) => {  
  return () => {  
  
  }  
}
```

# HIGHER ORDER FUNCTIONS

```
const logOnce = () => {  
  console.log('once')  
}
```

```
const addMoreLogging = (otherFunc) => {  
  return () => {  
    console.log('twice')  
    otherFunc()  
  }  
}
```



# HIGHER ORDER FUNCTIONS

```
const logOnce = () => {  
  console.log('once')  
}
```

```
const addMoreLogging = (otherFunc) => {  
  return () => {  
    console.log('twice')  
    otherFunc()  
  }  
}
```

```
const logTwice = addMoreLogging(logOnce)
```

# HIGHER ORDER FUNCTIONS

```
const logOnce = () => {  
  console.log('once')  
}
```

```
const addMoreLogging = (otherFunc) => {  
  return () => {  
    console.log('twice')  
    otherFunc()  
  }  
}
```

```
const logTwice = addMoreLogging(logOnce)  
  
logTwice()
```

# HIGHER ORDER COMPONENTS

```
const SimpleComponent = (props) => <p>{props.foo}</p>
```

```
const SimpleComponent = (props) => <p>{props.foo}</p>
```

```
const addLogging = (SomeComponent) => {
```

```
}
```

```
const SimpleComponent = (props) => <p>{props.foo}</p>
```

```
const addLogging = (SomeComponent) => {  
  return class Logger extends React.Component {
```

} }

```
const SimpleComponent = (props) => <p>{props.foo}</p>
```

```
const addLogging = (SomeComponent) => {  
  return class Logger extends React.Component {
```

```
    render () {  
      console.log('rendering with props: ', this.props)
```

```
    }
```

```
  }
```

```
}
```

```
const SimpleComponent = (props) => <p>{props.foo}</p>
```

```
const addLogging = (SomeComponent) => {  
  return class Logger extends React.Component {
```

```
    render () {  
      console.log('rendering with props: ', this.props)  
      return <SomeComponent {...this.props} />  
    }  
  }  
}
```



```
const SimpleComponent = (props) => <p>{props.foo}</p>

const addLogging = (SomeComponent) => {
  return class Logger extends React.Component {
    componentDidMount () {
      console.log('component mounted!')
    }

    render () {
      console.log('rendering with props: ', this.props)
      return <SomeComponent {...this.props} />
    }
  }
}
```

```
const SimpleComponent = (props) => <p>{props.foo}</p>

const addLogging = (SomeComponent) => {
  return class Logger extends React.Component {
    componentDidMount () {
      console.log('component mounted!')
    }

    render () {
      console.log('rendering with props: ', this.props)
      return <SomeComponent {...this.props} />
    }
  }
}

const EnhancedComponent = addLogging(SimpleComponent)
```

```
const SimpleComponent = (props) => <p>{props.foo}</p>

const addLogging = (SomeComponent) => {
  return class Logger extends React.Component {
    componentDidMount () {
      console.log('component mounted!')
    }

    render () {
      console.log('rendering with props: ', this.props)
      return <SomeComponent {...this.props} />
    }
  }
}
```

```
const EnhancedComponent = addLogging(SimpleComponent)
export default EnhancedComponent
```

