REACT COMPONENT

LIFECYCLE METHODS

WHAT IS LIFECYCLE METHODS

- Method that you can override to run code at particular times in the process.

EXAMPLE USAGE

```
class App extends React.Component {
  exampleLifecycleMethod() {
    ...
  }
}
```

MOUNTING LIFECYCLE

- * constructor()
- componentWillMount()
- * componentDidMount()
- componentWillUnmount()

* Frequently used

CONSTRUCTOR

- Called before component is mounted
- Should call super()

COMMON USE CASE

- Initialize state

EXAMPLE USAGE

```
class App extends React.Component {
    constructor() {
        super()
        ...
    }
}
```

COMPONENT WILL MOUNT

- Called before the render method is executed
- Only hook that called on server side
- Invoke once
- Call this.setState() without re-rendering

COMMON USE CASE

- Initialize state (Prefer using constructor() instead)
- Do something both on server side and client side

EXAMPLE USAGE

```
class App extends React.Component {

    componentWillMount() {

        ?????????????

    }

}
```

COMPONENT DID MOUNT

- Called after the render method is executed
- Only called on client side
- Invoke once

COMMON USE CASE

- Initialization requires DOM should go here
- Fetch data from remote endpoint

EXAMPLE USAGE

```
class App extends React.Component {
  componentDidmount() {
    fetch(‘http://localhost:3000’)
      .then(res => this.setState({ res }))
  }
}
```

COMPONENT WILL UNMOUNT

- Invoked immediately before a component is unmounted

UPDATING LIFECYCLE

- * componentWillReceiveProps(nextProps)
- shouldComponentUpdate(nextProps, nextState)
- componentWillUpdate(nextProps, nextState)
- componentDidUpdate(prevProps, prevState)

* Frequently used

COMPONENT WILL RECEIVE PROPS

- Invoked before a mounted component receive new props
- Invoke even if the props have not changed
- Not invoke during mounting

COMMON USE CASE

- Do something when props change before render

EXAMPLE USAGE

```
class App extends React.Component {
  componentWillReceiveProps(nextProps) {
    if (nextProps.id !== this.props.id) {
      fetch(‘http://localhost:3000’)
        .then(res => this.setState({ res }))
    }
  }
}
```

SHOULD COMPONENT UPDATE

- Invoked before rendering when props or state are being received
- Return boolean, default is true
- If return false lifecycle after this will not invoke

COMMON USE CASE

- Manually update component

COMPONENT WILL UPDATE

- Invoked immediately before rendering when props or state are being received
- Not invoke for initial render

COMPONENT DID UPDATE

- Invoked immediately after rendering when props or state are being received