COMPOSITION VS. INHERITANCE

Composition over Inheritance

- Composition and inheritance are the approaches to use multiple components together in React.js .
- This helps in code reuse.
- React recommend using composition instead of inheritance as much as possible and inheritance should be used in very specific cases only.
- Composition works with functions as well as classes both.

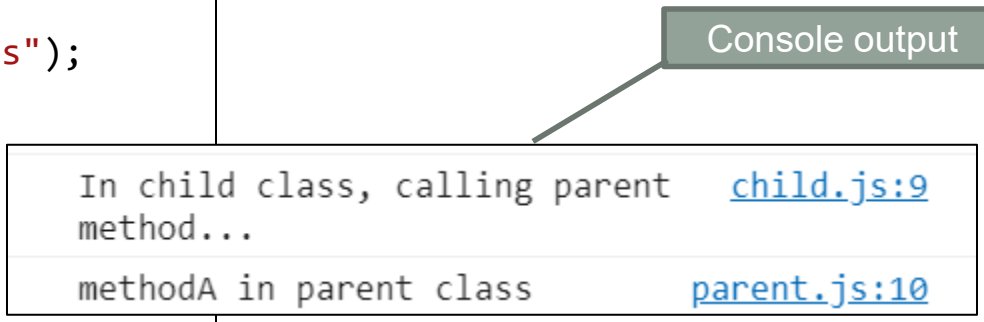
Inheritance in JS

```
class Automobile {
    constructor() {
        this.vehicleName = automobile;
        this.numWheels = null;
    }
    printNumWheels() {
        console.log(`This ${this.vehicleName} has ${this.numWheels} wheels`);
    }
}
class Car extends Automobile {
    constructor() {
        super(this);
        this.vehicleName = 'car';
        this.numWheels = 4;
    }
}
class Bicycle extends Automobile {
    constructor() {
        super(this);
        this.vehicleName = 'bike';
        this.numWheels = 2;
    }
}
const car = new Car();
const bike = new Bicycle();
car.printNumWheels() // This car has 4 wheels
bike.printNumWheels() // This bike has 2 wheels
```

```
class Parent extends React.Component {
  constructor(props) {
    super(props);
    this.methodA = this.methodA.bind(this);
  }

  methodA() {
    console.log("methodA in parent class");
  }

  render() {
    return false;
  }
}
```




```
import Parent from "./parent";

 class Child extends Parent {
    constructor() {
      super();
    }
    render() {
      console.log("In child class, calling parent method...");
      this.methodA();
      return false;
    }
  }
```

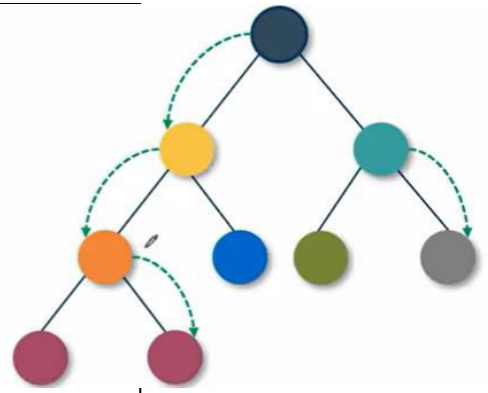
Composition

- Composition is a code reuse technique where a larger object is created by combining multiple smaller objects.

```
class App extends React.Component {
    render() {
      return <Toolbar theme="dark" />;
    }
  }

  function Toolbar(props) {
    return (
      <div>
        <ThemedButton theme={props.theme} />
      </div>
    );
  }

  class ThemedButton extends React.Component {
    render() {
      return <Button theme={this.props.theme} />;
    }
  }
```



```
class App extends Component {
state = {
   date:new Date()
...
}
...
return (
  <div className="container">
    <NewsHeader className="news" subject="Sports"
          date={this.state.date.toString()} />
```

```
const newsHeader = (props) => {
 return(
 <div>
    <h1>News for {props.date}</h1>
    <h2>News Heading : {props.subject}</h2>
    <NewsContent title="Content Title-1" content="Lots of Content-1" />
    <NewsContent title="Content Title-2" content="Lots of Content-2" />
    <NewsContent title="Content Title-3" content="Lots of Content-3" />
```

```
const newsContent = (props) => {
 return(
  <div>
    {/*complex code that filters out news based on subject*/}
    <h4><b><i>News Title {props.title}</i></b></h4>
    <h4>News Content : {props.content}</h4>
    <Author title={props.title} name="Shrilata" />
```

```
const author = (props) => {
 return(
  <h6>Author for {props.title} - {props.name}</h6>
```

News for Sun Jun 1

News Heading : Sports

News Title Content Title-1

News Content : Lots of Content-1

Author for Content Title-1 - Shrilata

News Title Content Title-2

News Content : Lots of Content-2

Author for Content Title-2 - Shrilata

News Title Content Title-3

News Content : Lots of Content-3

Author for Content Title-3 - Shrilata