

Install node js from [nodejs.org](https://nodejs.org)

After installing node js hit bellow command in terminal

```
npx create-react-app my-app
```

This will create a react app named my-app and install all the dependences its need.

Then go to the my-app folder using 'cd' command and in the my-app folder where package.json file lives enter

```
npm start
```

This will execute the project.

### >> How TO Add **Dynamic Content** in React (JSX) content

```
<p>i am a person and i am { Math.floor(Math.random() *30) }</p>
```

Where "Math.floor(Math.random() \*30)" is a js function that will return random value

Dynamic values are added within curly braces { Dynamic value }

### >> Working With **Props**

In app.js

```
import Person from './Person/Person'
class App extends Component {
  render() {
    return (
      <div className="App">
        <h1>hello i am react</h1>
        <Person name = "neeraj" age="55"/>
        <Person name = "joy" age="43"/>
      </div>
    );
  }
  //render
}
```

In Person.js

```
import React from 'react';
const person = (props) => {
  return(
    <div>
      <p>i am a {props.name} and i am {props.age}</p>
    </div>
  );
}
export default person;
```

>> **Children properties**

In app.js

```
import Person from './Person/Person'
class App extends Component {
  render() {
    return (
      <div className="App">
        <Person name = "joy" age="43"> This is a children text </Person>
      </div>
    );
  }
  //render
}
```

In Persion.js

```
import React from 'react';
const person = (props) => {
  return(
    <div>
      <p>i am a {props.name} and i am {props.age}</p>
      <p>{props.children}</p>
    </div>
  );
}
export default person;
```

>> **STATE** in React

In app.js

```
import Person from './Person/Person'
class App extends Component {

  state = {
    person:[
      { name:'neeraj', age: 100 },
      { name:'joy', age: 211 }
    ]
  }
  //state

  render() {
    return (
      <div className="App">
        <Person name = {this.state.person[0].name} age={this.state.person[0].age} />
        <Person name = {this.state.person[0].name} age={this.state.person[0].age} />
      </div>
    );
  }
  //render
}
```

In Persion.js

```
import React from 'react';
const person = (props) => {
  return(
    <div>
      <p>i am a {props.name} and i am {props.age}</p>
    </div>
  );
}
export default person;
```

>> **OnClick listener**

In app.js

```
import Person from './Person/Persion'
class App extends Component {

  state = {
    person:[
      { name:'neeraj', age: 100 },
      { name:'joy', age: 211 }
    ]
  }//state

  switchNameHandler = () => {
    console.log("switch name is called");
  }

  render() {
    return (
      <div className="App">
        <button onClick={this.switchNameHandler}>Switch Name</button>
        <Person name = {this.state.person[0].name} age={this.state.person[0].age} />
        <Person name = {this.state.person[0].name} age={this.state.person[0].age} />
      </div>
    );
  }//render
}
```

>> **setState**

In app.js

```
import Person from './Person/Persion'
class App extends Component {

  state = {
    person:[
      { name:'neeraj', age: 100 },
      { name:'joy', age: 211 }
    ]
  }//state
```

```

switchNameHandler = () => {
  this.setState({
    person:[
      { name:'neeraj_sharma', age: 100 },
      { name:'joy', age: 211 }
    ]
  })//setState
}//switchNameHandler

render() {
  return (
    <div className="App">
      <button onClick={this.switchNameHandler}>Switch Name</button>
      <Person name = {this.state.person[0].name} age={this.state.person[0].age} />
      <Person name = {this.state.person[0].name} age={this.state.person[0].age} />
    </div>
  );
}//render
}

```

### >> On Click in persion.js function

In app.js

```

import Person from './Person/Persion'
class App extends Component {

```

```

  state = {
    person:[
      { name:'neeraj', age: 100 },
      { name:'joy', age: 211 }
    ]
  }//state

```

```

switchNameHandler = () => {
  this.setState({
    person:[
      { name:'neeraj_sharma', age: 100 },
      { name:'joy', age: 211 }
    ]
  })//setState
}//switchNameHandler

```

```

render() {
  return (
    <div className="App">
      <button onClick={this.switchNameHandler}>Switch Name</button>
      <Person
        name = {this.state.person[0].name}
        age = {this.state.person[0].age}
        click = {this.switchNameHandler}
      />
    </div>
  );
}//render
}

```

In Persion.js

```
import React from 'react';
const person = (props) => {
  return(
    <div>
      <p onClick={props.click}>i am a {props.name} and i am {props.age}</p>
    </div>
  );
}
export default person;
```

**>> Pass argument to hander >> bind**In app.js

```
import Person from './Person/Persion'
class App extends Component {

  state = {
    person:[
      { name:'neeraj', age: 100 },
      { name:'joy', age: 211 }
    ]
  } //state

  switchNameHandler = (newName) => {
    this.setState({
      person:[
        { name:newName, age: 100 },
        { name:'joy', age: 211 }
      ]
    }) //setState
  } //switchNameHandler

  render() {
    return (
      <div className="App">

        <button
          onClick={this.switchNameHandler.bind(this,"neeraj_sharma")} >
          Switch Name
        </button>

        <Person
          name = {this.state.person[0].name}
          age = {this.state.person[0].age}
          click = {this.switchNameHandler.bind(this,"neeraj_kumar")}
        />
      </div>
    );
  } //render
}
```

**>> Another way Pass argument to handler without bind using arrow method =>**

There is a another way to pass argument but it is not the most efficient way

```
<button
onClick={ () => this.switchNameHandler("neeraj_sharma")}>
Switch Name
</button>
```

**>> Input tag**

In Persion.js

```
import React from 'react';
const person = (props) => {
  return(
    <div>
      <p>i am a {props.name} and i am {props.age}</p>
      <input type="text" onChange={props.changed} value={props.name}/>
    </div>
  );
}
export default person;
```

In app.js

```
import Person from './Person/Persion'
class App extends Component {

  state = {
    person:[
      { name:'neeraj', age: 100 },
      { name:'joy', age: 211 }
    ]
  }//state

  nameChangeHandler = (event) => {
    this.setState({
      person:[
        { name:event.target.value, age: 100 },
        { name:'joy', age: 211 }
      ]
    })//setState
  }//nameChangeHandler

  render() {
    return (
      <div className="App">

        <Person
          name = {this.state.person[0].name}
          age = {this.state.person[0].age}
          changed= {this.nameChangeHandler}/>

        </div>
      );
    }//render
  }
}
```