|  |  |
| --- | --- |
| TOPIC/TITLE  React Rendering with the ternary operation and operator | |
| Keywords/Questions | Notes : |
|  |
| function renderConditionnaly() {  if (isLoggedIn === true) {  return <h1>Hello</h1>;  } else {  return <Login />;  }  }  function App() {  return <div className="container">{renderConditionnaly()}</div>;  } |
| The problem with this method it the fact that it’s a statement not an expression and JSX can’t read statement so we have to use ternary operation js method |
|  |
|  |
|  |
| Can’t work |
| function App() {  return <div className="container">{  if (isLoggedIn === true) {  return <h1>Hello</h1>;  } else {  return <Login />;  }  }  </div>;  } |
| work |
| function App() {  return (  <div className="container">{isLoggedIn ? <h1>Hello</h1> : <Login />}</div>  );  }  You can use null if you don’t want one the condition (true or false) |
| const currentTime = new Date().getHours();  console.log(currentTime);  function App() {  return (  <div className="container">{  *//isLoggedIn ? <h1>Hello</h1> : <Login />*  currentTime>20? <h1>why are you still working?????</h1>:null |
|  |
|  |
|  |
|  |
|  |
|  |
| You can write the expression like this |
|  |
| currentTime>20 && <h1>why are you still working?????</h1> |
| Because and operator require both side to be true it won’t make it true. If currently is good it will directly check for the second expression which is the H1 |
| Practice solution |
| function App() {  return (  <div className="container">  <Form isRegistered={userIsRegistered} />  </div>  );  }  export default App; |
| function Form(props) {  return (  <form className="form">  <input type="text" placeholder="Username" />  <input type="password" placeholder="Password" />  {!props.isRegistered && (  <input type="password" placeholder="Confirm Password" />  )}  <button type="submit">{props.isRegistered ? "Login" : "Register"}</button>  </form>  );  } |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  | |