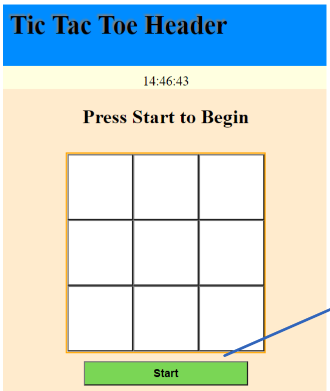
Conditional Component Rendering

Thursday, March 24, 2022   2:46 PM

- Sometimes we need to render a component conditionally
- Here we want to display the 'Start' button only when **this.state.move===null**
-

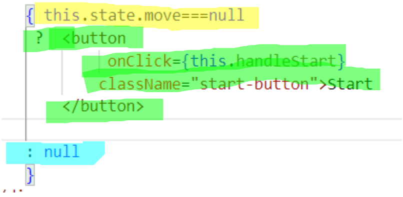


Display this button when game is not running

- Not started
- Finished.

Challenge

- JSX doesn't support statement like if
- We can use conditional operator lie
    - Ternary operator
    - Short circuit && or ||



```
{ this.state.move===null
 ? <button
     onClick={this.handleStart}
    className="start-button">Start
   </button>

 : null
}
```

- Check the condition
- Render one UI if condition is true

- Render other UI if condition is false
    - null here means, render no UI in false case

Alternative — use short circuit operator

```
{   this.state.move  ||   <button>Start </button> }
```

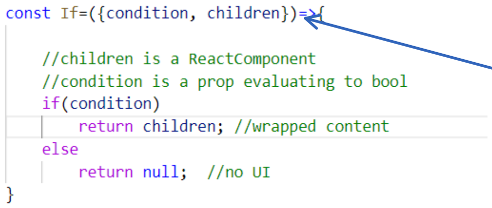
- If first part is a falsy (false, 0, null, undefined)
- Second part is evaluated only when first part is false

```
{  this. state. book   &&   <BookInfo book={book} /> }
```

- If book exists (truthy) then only display the BookInfo
- Second part is evaluated only if first part is true

Create a custom If component

- Display component only when some condition is true.



```
const If=({condition, children})=>{

    //children is a ReactComponent
    //condition is a prop evaluating to bool
    if(condition)
        return children; //wrapped content
    else
        return null;  //no UI
}
```

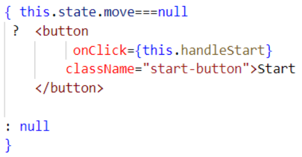
```
{

     condition: true,
     children:<h1>Condition was True</h1>

}
```

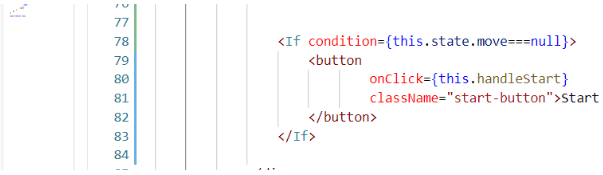
Usage

```
<If condition={a>b} >
     <h1>Condition was true</h1>
</If>
```



```
{ this.state.move===null
 ?  <button
       onClick={this.handleStart}
      className="start-button">Start
    </button>

: null
}
```



```
<If condition={this.state.move===null}>
    <button
            onClick={this.handleStart}
            className="start-button">Start
    </button>
</If>
```