**Begin   
Jak włączyć wsparcie?**$npx // używane do binarnych skryptów do inicjalizacji i do operacji bez projektu

$npx create-react-app first –template typescript

$npm start – creates developer server

\*\*public\*\*

index.html - <div id=”root”> <- miejsce instalacji aplikacji w reacie

\*\*src\*\*

index.tsx

root.render(

<React.ScriptMode> <App /> </>);

) /// uruchamia elementy

className = {value} /// atrybut jak class w html’u

file name & component name (e.x. App) should start with upperCaseLetters

‘

\*\*hooks\*\*

\*\*Example component\*\*

function Message() {  
 *// JSX: Will be compiled to js form a'like html* const *temp* = 'Abxc';  
 return <**h1**>**Hellow** {*temp*}</**h1**>;  
}  
  
export default Message;

import Message from "./Message";  
  
function App() {  
 return <**div**><**Message**/></**div**>;  
}  
  
export default App;

\*\*Wypisywanie tablicy jako listy

function ListGroup(){  
 const *items* = ["Hi", "There", "One"];  
 *items*.map(item => <**li**>{item}</**li**>)  
 return (  
 <><**h1**>**List**</**h1**>  
 <**ul** className="list-group">  
 {*items*.map(item => <**li** key={item} className="list-group-item">{item}</**li**>)}  
 </**ul**></>  
 );  
}

\*\*Hooks\*\*

używane do dostawania się do metod reacta

\*updating number\*

const arr[{nazwa zmiennej przechowywanej}, set{nazwa zmiennej}] = useState({value}) // [0] – varieble; [1] – updater function

const [*selectedIndex*, setSelectedIndex] = useState(-1);

onClick={() => {setSelectedIndex(index)}}

\*\*Additional \*\*

<></> - pomiędzy tym można zwracać parę obiektów na raz z componentu, dlatego bo bazowo tam jest używany <Fragment>

\*children\*

Do pozyskania dzieci z App.tsx jako argument

interface Prop{

children:text;

}

children: ReactNode; // Przesyła html content