8. Distructuring

*var* array = ['javascript', 'css', 'js'];  
*var* [a, ...rest] = array;  
console.log(rest);  
  
*var* course = {  
 name: 'javascript',  
 price: 1000,  
 image: 'image-address',  
 children: {  
 name: 'reactjs'  
 }  
};  
*var* {name, ...rest} = course;  
console.log(rest);  
  
*var* {name: parentName, children: {name}} = course;  
console.log(parentName);  
console.log(name);

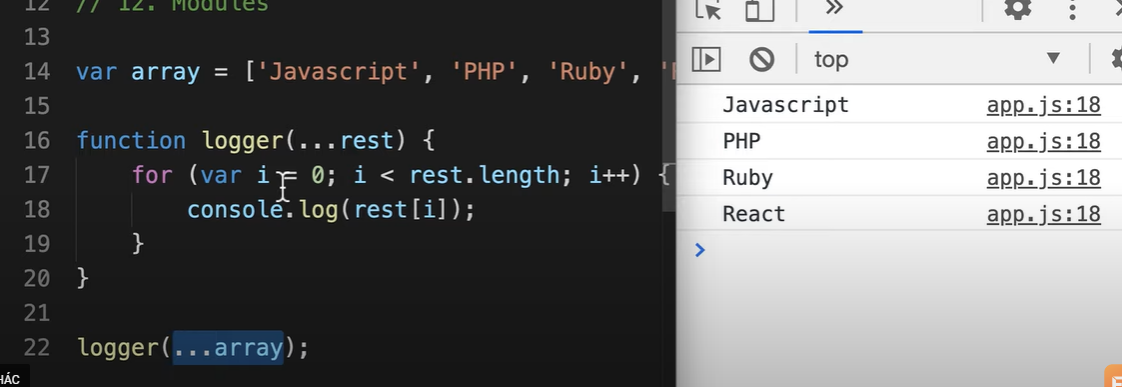
*const* {  
 a = 'default',  
 b,  
 ...rest  
} = {  
 b: 'val1',  
 c: 'val2',  
 d: 'val3'  
};  
  
console.log(a, b, rest); *‘default’,’val1’,{c:’val2’;d:’val3’}*

10. spreat – Nối mãng, object

*var* array1 = ['javascript', 'css', 'js', 'jsx'];  
*var* array2 = ['react', 'react-dom', 'react-dom-dom', 'react-dom-dom'];  
*var* array3 = [...array1, ...array2];  
console.log(array3)

*var* object1 = {  
 name: 'javascript',  
}  
*var* object2 = {  
 price: 1000  
}  
*var* object3 = {

name: ‘abc’,  
 ...object1,  
 ...object2  
}  
console.log(object3)

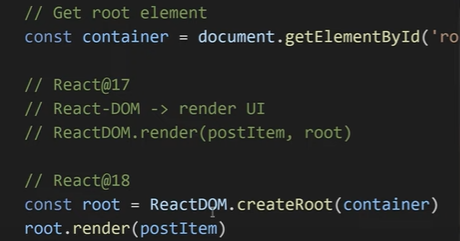


React create element

<script>  
 *const* h1DOM = document.createElement('h1');  
 h1DOM.tite = 'hello';  
 h1DOM.className = 'heading';  
 h1DOM.innerText = 'hello guys!';  
 document.body.appendChild(h1DOM);  
  
 *// react  
 // react.createElement(type,props,children,n)  
 const* h1React = React.createElement('h1',{  
 title: 'hello',  
 className: 'heading'  
 },'Hello guys!')  
  
 *const* ulReact = React.createElement(  
 'ul',  
 {  
 id: 'ul-id',  
 style: 'color: red;'  
 },  
 React.createElement('li',*null*,'Javascript'),  
 React.createElement('li',*null*,'ReactJS')  
 )  
 console.dir(h1DOM);  
 console.log(h1React);  
 console.log(ulReact);  
 document.body.appendChild(ulReact)  
</script>

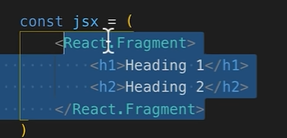
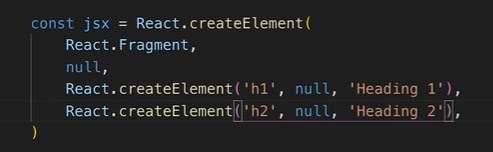
ReactDOM

<script>  
 *const* postItem = React.createElement(  
 'div',  
 {  
 className: 'post-item'  
 },  
 React.createElement(  
 'h2',  
 {  
 title: 'hoc react'  
 },  
 'hoc react'  
 ),  
 React.createElement('p',{},'hoc reactjs')  
 )  
 console.log(postItem)  
 ReactDOM.render(postItem, root)  
</script



JSX

<script *type*="text/babel">  
 *const* course = [  
 {  
 name: 'html,css'  
 },  
 {  
 name: 'web responsive'  
 },  
 {  
 name: 'reactjs'  
 }  
 ]  
  
 *const* jsx = (  
 <ul>  
 {course.map(course=>  
 <li>{course.name}</li>  
 )}  
 </ul>  
 );  
 console.log(jsx);  
 ReactDOM.render(jsx, document.getElementById('root'))  
</script>

= 

React Element Type

<script *type*="text/babel">  
 *function* Header(){  
 *return* (  
 <div *className*="header">New Header</div>  
 )  
 }  
  
 *class* Content *extends* React.Component {  
 render(){  
 *return* (  
 <div *className*="content">New Content</div>  
 )  
 }  
 }  
  
 *const* app = (  
 <div *className*="wrapper">  
 <Header></Header>  
 <Content></Content>  
 <div *className*="footer">Footer</div>  
 </div>  
 )  
  
 ReactDOM.render(app, document.getElementById('app'))  
</script>

<script *type*="text/babel">  
  
 *const* Form = {  
 Input(){  
 *return* <input />  
 },  
 Checkbox(){  
 *return* <input *type*="checkbox"/>  
 }  
 }  
  
 *function* Button({title,href,onClick}){  
 *let* Component = 'button';  
 *const* props = {}  
  
 *if*(href){  
 Component = 'a';  
 props.href = href;  
 }  
 *if*(onClick){  
 props.onClick = onClick;  
 }  
 *return* (  
 <Component {...props}>{title}</Component>  
 )  
 }  
  
 *function* App(){  
 *const* type = 'Input';  
 *const* Component = Form[type];  
 *return* (  
 <React.Fragment>  
 <div *id*="wrapper">  
 <Component/>  
 </div>  
 <Button  
 *title*="Click me!"  
 *// href = "google.com"  
 onClick*={()=>console.log(Math.random())}  
 />  
 </React.Fragment>  
 )  
 }  
  
 ReactDOM.render(<App/>, document.getElementById('root'))

JSX #3

