Javascript / React General Notes

Tags	Tools
Created time	@April 20, 2023 10:19 PM
Last edited time	@April 24, 2023 10:34 AM

```
import React, { useState } from "react";

function Use() {
  const [cnt, setCnt] = useState(0);

  function handleClick(newCnt) {
    setCnt(newCnt);
  }

  return (
    <div>
      <p>A cnt {cnt} created by useState </p>
      <button
        value="click to decrement cnt"
        onClick={() => {
          handleClick(cnt - 1);
        }}
        disabled={cnt <= 0}
      >
        Click to decrement cnt
      </button>
    </div>
  );
}

export default Use;
```

```
function Use () {
  const [cnt, setcnt] = useState(0);

  function handleClick(newcnt){
    setcnt(newcnt);
  }

  return (
  <div>
  <p>A cnt {cnt} created by useState </p>
  <button value="click to increment cnt" onClick={this.handleClick.bind(this, cnt - 1)} />
  <button value="click to increment cnt" onClick={this.handleClick.bind(this, cnt + 1)} />
  </div>
  );
}
```

```
import React from 'react';

class Set extends React.Component {
  constructor(props){
    super(props);
    this.state = {
      cnt: 0,
    }
    this.handleClick = this.handleClick.bind(this);
  }
  handleClick() {
    this.setState({cnt: this.state.cnt + 1});
  }

  render() {
    return (
      <div>
        <p>Another cnt {this.state.cnt} created by setState</p>
        <button onClick={this.handleClick}>Click</button>
      </div>
    )
  }
}
```

```
export Hello;

//then in another file it will be

import { Hello } from './hello';

///////////////////////////////////////

export default Hello;

//then

import Hello form './hello';
```

By using the named export syntax with curly braces, you are explicitly importing the `Hello` component from the `hello` module.

This means that you can also have other named exports in the same module, which can be useful for organizing related components or functions in a single file.

However, if you try to import the `Hello` component using `import Hello from './hello'` (without curly braces), you would get an error because there is no default export in the `hello` module.

So, if you want to use the `export Hello` syntax, make sure to import the component using curly braces and the exact name of the exported component.

```
function addNumbers(a, b) {
  return a + b;
}

const addFive = addNumbers.bind(null, 5);

console.log(addFive(10)); // Output: 15
```

```
function sayHi() {
  console.log(this);
}

sayHi(); // refers to the global object
// Window {0: global, window: Window, self: Window, document: document, name: '', location: Location, …}

const person = {
  name: 'John',
  sayHi: function() {
    console.log(this);
  }
};

person.sayHi(); // refers to the person object
// {name: 'John', sayHi: f}
```

```
{
  "name": "John",
  "age": 30,
  "city": "New York"
}
```

```
[  {    "name": "John",    "age": 30,    "city": "New York"  },  {    "name": "Jane",    "age": 25,    "city": "Los Angeles"  }]
```

JSON is a data interchange format that is designed to be independent of any particular programming language. It has a specific syntax that allows data to be serialized and deserialized between different programming languages. JSON requires that all keys be enclosed in double quotes, and only a limited set of data types are supported, including strings, numbers, booleans, arrays, and other JSON objects.

On the other hand, JavaScript objects can have keys that are not enclosed in quotes, and they can also contain functions as values, which are not supported in JSON.

```
function MyComponent(props) {
  const handleClick = (myProp, event) => {
    const buttonId = event.target.id;
    event.target.innerText = 'Button ' + buttonId + ' was clicked!';
    // handle click event with myProp and buttonId
  };

  return (
    <div>
      <button id="buttonA" onClick={handleClick.bind(null, 'propA')}>Button A</button>
      <button id="buttonB" onClick={handleClick.bind(null, 'propB')}>Button B</button>
    </div>
  );
}
```

```
<button style={{ paddingRight: '20px' }}>+</button>
```

The reason we need two sets of curly braces (`{{ }}`) in the `style` attribute is because the first set of curly braces indicates that we're embedding a JavaScript expression in JSX, while the second set of curly braces creates a JavaScript object literal.

The outer curly braces are used to embed a JavaScript expression in JSX. In this case, we're using an object literal to define the styles we want to apply to the button. The inner curly braces are used to create a JavaScript object literal containing the actual CSS styles.

So, the first set of curly braces tells React that we're embedding a JavaScript expression in JSX, and the second set of curly braces creates a JavaScript object literal that defines the styles we want to apply.