Spring AU ’21 – React JS – Morning Session

Name: Sheik Abudhahir K

Date: 13/01/2021

<u>Types of Cloning the objects:</u>

Normally in javascript objects are reference typed, so we can’t assign directly suppose we can assign the objects directly let’s see what will happen

```
assignment.js ×
Day 6 > js > assignment.js > cloneObject > cloneData
const data = {
    "name":"Sheik Abudhahir",
    "age":"23",
    "email":"sheik.k@accolitedigital.com",
    "phone":"9994648367"
  };

  const cloneObject = () => {

    const cloneData = data
    //Both objects before modifying
        console.log(data);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    console.log(cloneData);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    cloneData.name = "Sheik"
    //Both objects after modifying the clone data
        console.log(data);  { name: 'Sheik', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    console.log(cloneData);  { name: 'Sheik', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    //Here modified cloned data is not affected the original data
  };

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                                   Quokka
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at cloneData Day 6/js/assignment.js:13:4

{ name: 'Sheik',
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at data Day 6/js/assignment.js:16:9

{ name: 'Sheik',
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at cloneData Day 6/js/assignment.js:17:4
```

Here I used the ‘=’ operator to clone the object, this case if we change the cloning object it also will affect the base object.

To avoid this problem we move to some other methods to clone object.

1. Spread
2. Objects.assign()
3. Deep copy using JSON

Spread

Here I modify the cloned object but it does not affect the base object data

Usually we use '…' for spread eg: obj2 = {…obj1} -> obj1 is already defined object

assignment.js

Day 6 > js > assignment.js > cloneObject > cloneData

```js
const data = {
    "name":"Sheik Abudhahir",
    "age":"23",
    "email":"sheik.k@accolitedigital.com",
    "phone":"9994648367"
  };

  const cloneObject = () => {

    const cloneData = {...data}
    //Both objects before modifying
        console.log(data);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    console.log(cloneData);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    cloneData.name = "Sheik"
    //Both objects after modifying the clone data
        console.log(data);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    console.log(cloneData);  { name: 'Sheik', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    //Here modified cloned data is not affected the original data
  };
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Quokka

```
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at cloneData Day 6/js/assignment.js:13:4

{ name: 'Sheik Abudhahir',
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at data Day 6/js/assignment.js:16:9

{ name: 'Sheik',
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at cloneData Day 6/js/assignment.js:17:4
```

This one is shallow copy method, still the sub-objects are connected with the old base object

eg: In this example I changed the sub-object name.first, it was affected the base object too. Then I changed the email it wasn't affect the base object.

```js
const cloneObject = () => {

  const cloneData = {...data}
  //Both objects before modifying
      console.log(data);  { name: { first: 'Abu', second: 'Sharp' }, age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
  console.log(cloneData);  { name: { first: 'Abu', second: 'Sharp' }, age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
  cloneData.name.first = "Sheik"
  cloneData.email = "k.s.abudhahir@gmail.com"
  //Both objects after modifying the clone data
      console.log(data);  { name: { first: 'Sheik', second: 'Sharp' }, age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
  console.log(cloneData);  { name: { first: 'Sheik', second: 'Sharp' }, age: '23', email: 'k.s.abudhahir@gmail.com', phone: '9994648367' }
  //Here modified cloned data is not affected the original data
};
```

Object.assign()

Here also I modified the age in cloned object, but this modification does not affect the base object.

That is, I mentioned in the example below.

Eg for method -> obj2 = Object.assign({}, obj1)

```
JS assignment.js ●
Day 6 > js > JS assignment.js > [e] cloneObject
1   const data = {
2       "name":"Sheik Abudhahir",
3       "age":"23",
4       "email":"sheik.k@accolitedigital.com",
5       "phone":"9994648367"
6     };
7
8     const cloneObject = () => {
9
10      const cloneData = Object.assign({},data)
11      //Both objects before modifying
12          console.log(data);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
13      console.log(cloneData);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
14      cloneData.age = "20"
15      //Both objects after modifying the clone data
16          console.log(data);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
17      console.log(cloneData);  { name: 'Sheik Abudhahir', age: '20', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
18      //Here modified cloned data is not affected the original data
19    };
20
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                    Quokka
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at cloneData Day 6/js/assignment.js:13:4

{ name: 'Sheik Abudhahir',
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at data Day 6/js/assignment.js:16:9

{ name: 'Sheik Abudhahir',
  age: '20',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at cloneData Day 6/js/assignment.js:17:4
```

This one is shallow copy method, still the sub-objects are connected with the old base object

eg: In this example also I changed the sub-object name.first, it was affected the base object too. Then I changed the age it wasn't affect the base object.

```
11    const cloneObject = () => {
12
13      const cloneData = Object.assign({},data)
14      //Both objects before modifying
15          console.log(data);  { name: { first: 'Abu', second: 'Sharp' }, age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
16      console.log(cloneData);  { name: { first: 'Abu', second: 'Sharp' }, age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
17      cloneData.name.first = "Sheik"
18      cloneData.age = "20"
19      //Both objects after modifying the clone data
20          console.log(data);  { name: { first: 'Sheik', second: 'Sharp' }, age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
21      console.log(cloneData);  { name: { first: 'Sheik', second: 'Sharp' }, age: '20', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
22      //Here modified cloned data is not affected the original data
23    };
24
```

To avoid this problem we go to deep copy method (JSON)

JSON

Here I changed the mail id of the cloned object, and before this whatever we saw was shallow copy methods, but this one is deep copy method, this means clone the entire object and remove the link between new object and old object.

```
assignment.js ×
Day 6 > js > assignment.js > cloneObject
const data = {
    "name":"Sheik Abudhahir",
    "age":"23",
    "email":"sheik.k@accolitedigital.com",
    "phone":"9994648367"
  };

  const cloneObject = () => {

    const cloneData = JSON.parse(JSON.stringify(data))
    //Both objects before modifying
        console.log(data);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    console.log(cloneData);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    cloneData.email = "k.s.abudhahir@gmail.com"
    //Both objects after modifying the clone data
        console.log(data);  { name: 'Sheik Abudhahir', age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    console.log(cloneData);  { name: 'Sheik Abudhahir', age: '23', email: 'k.s.abudhahir@gmail.com', phone: '9994648367' }
    //Here modified cloned data is not affected the original data
  };

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                                   Quokka
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at cloneData Day 6/js/assignment.js:13:4

{ name: 'Sheik Abudhahir',
  age: '23',
  email: 'sheik.k@accolitedigital.com',
  phone: '9994648367' }
  at data Day 6/js/assignment.js:16:9

{ name: 'Sheik Abudhahir',
  age: '23',
  email: 'k.s.abudhahir@gmail.com',
  phone: '9994648367' }
  at cloneData Day 6/js/assignment.js:17:4
```

Another one example for sub-object

Here modification of sub-object wasn't affected the base object 'data'

```
    const cloneData = JSON.parse(JSON.stringify(data))
    //Both objects before modifying
        console.log(data);  { name: { first: 'Abu', second: 'Sharp' }, age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    console.log(cloneData);  { name: { first: 'Abu', second: 'Sharp' }, age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    cloneData.name.first = "Sheik"
    cloneData.email = "k.s.abudhahir@gmail.com"
    //Both objects after modifying the clone data
        console.log(data);  { name: { first: 'Abu', second: 'Sharp' }, age: '23', email: 'sheik.k@accolitedigital.com', phone: '9994648367' }
    console.log(cloneData);  { name: { first: 'Sheik', second: 'Sharp' }, age: '23', email: 'k.s.abudhahir@gmail.com', phone: '9994648367' }
    //Here modified cloned data is not affected the original data
  };
```