

디지털컨버전 스 기반 UXUI Front 전문 개발자 양성과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 - JungHyun
LEE(이정현)

akdl911215@naver.com

1) NewSetTest

```
1 const NewSetTest = () => {
2
3   let setData = new Set()
4   console.log(setData)
5   // 출력 : Set(0) { }
6   //       아무런 데이터 값이 없기 때문에 안뜬다.
7
8   let setData2 = new Set()
9   setData2.add(1)
10  setData2.add("betazon")
11  console.log(setData2)
12  // 출력 : Set(2) {1, "betazon"}
13
14
15  return (
16    <div className="NewSetTest">
17      <p>NewSetTest</p>
18    </div>
19  )
20 }
21
22 export default NewSetTest
```

2) NewSetInitTest

```
1 const NewSetInitTest = () => {
2   let setData = new Set(["Banana", "Watermelon"])
3
4   console.log("NewSetInitTest: " + setData)
5   // 출력 : [object Set]
6   console.log("NewSetInitTest: " + setData.size)
7   // 출력 : 2
8
9   return (
10     <div className="NewSetInitTest">
11       <p>NewSetInitTest</p>
12     </div>
13   )
14 }
15
16 export default NewSetInitTest
```

3) AddSetTest

```
1  const AddSetTest = () => {
2    let setData = new Set()
3
4    setData.add("Cherry")
5    setData.add("strawberry")
6    setData.add("apple")
7
8    // 정 : addSet은 해당배열에 지정한 값이 존재하냐?
9    // true false로 반환한다.
10
11   console.log(setData)
12   // 출력 : {"Cherry", "strawberry", "apple"}
13   console.log("addSetTest: " + setData.has("apple"))
14   // 출력 : true
15   console.log("addSetTest: " + setData.has("grape"))
16   // 출력 : false
17
18   return (
19     <div className="AddSetTest">
20       <p>
21         AddSetTest
22       </p>
23     </div>
24   )
25 }
26 export default AddSetTest
```

4) ForEachWithSet

```
1  const ForEachWithSet = () => {
2      let setData = new Set()
3
4      setData.add("Cherry")
5      setData.add("strawberry")
6      setData.add("apple")
7
8      console.log(setData)
9
10     // (val1, val2) or (val1, val3) or (val2, val3)
11     // 동일하게 출력
12     // 매개변수명은 달라도 영향은 없다.
13     // 다만, forEach 동작하면 setData에 데이터값들을 2개씩
14     // 출력해준다.
15
16     setData.forEach(function(val1, val2) {
17         console.log(val1 + " : " + val2)
18     })
19     // 출력 : Cherry : Cherry
20     //       strawberry : strawberry
21     //       apple : apple
22
23     setData.forEach(function(val1, val3) {
24         console.log(val1 + " : " + val3)
25     })
26     // 동일 출력
27
28     setData.forEach(function(val2, val3) {
29         console.log(val2 + " : " + val3)
30     })
31     // 동일 출력
32
33     // arrow function은 ES6에 새롭게 추가된 기능으로 문법 코드가 간결하고
34     // 직관적이어서 이미 많은 개발자들이 이용하고 있으며, 높은 비율로 function을
35     // 대체하고 있다.
36
37     // setData.forEach(function(val2, val3) { 을
38     // setData.forEach((val1, val2) => 대체하는 것임.
39     setData.forEach((val1, val2) =>
40         console.log("arrow - " + val1 + " : " + val2)
41     )
42     // 출력은 "arrow - " 제외하고 동일
43
44
45     return (
46         <div className="ForEachWithSet">
47             <p>ForEachWithSet</p>
48         </div>
49     )
50 }
51
52 export default ForEachWithSet
```

5) SetEntryIterationTest

```
1  const SetEntryIterationTest = () => {
2      console.log("SetEntryIterationTest Start")
3
4      let setData = new Set()
5
6      setData.add("Cherry")
7      setData.add("strawberry")
8      setData.add("apple")
9      setData.add("바보")
10     setData.add("천!재!")
11     setData.add("!@#$$%^")
12     setData.add(1)
13     setData.add(999)
14
15     console.log(setData)
16     // 출력 : {"Cherry", "strawberry", .....}
17     //       모든게 출력이 된다.
18
19     var setIter = setData.entries()
20
21     for (var entry of setIter) {
22         console.log(entry)
23         // 출력 : {"Cherry", "Cherry"}
24         //       ...
25         //       {999, 999}
26     }
27
28     console.log("SetEntryIterationTest Fin")
29
30     return (
31         <div className="SetEntryIterationTest">
32             <p>SetEntryIterationTest</p>
33         </div>
34     )
35 }
36
37 export default SetEntryIterationTest
```

6) SetKeyIterationTest

```
1  const SetKeyIterationTest = () => {
2      console.log("SetKeyIterationTest Start")
3
4      let setData = new Set()
5
6      setData.add("cherry")
7      setData.add("strawberry")
8      setData.add("apple")
9      setData.add("바보")
10     setData.add("천재!")
11     setData.add("!@#$$%^")
12     setData.add(1)
13     setData.add(999)
14
15     console.log(setData)
16     // 출력 : {"Cherry", "strawberry", .....}
17     //       모든게 출력이 된다.
18
19     var setIter = setData.keys()
20
21     for (var key of setIter) {
22         console.log(key)
23         // 출력 : cherry
24         //       ...
25         //       999
26     }
27
28     console.log("SetKeyIterationTest Fin")
29
30     return (
31         <div className="SetIterationTest">
32             <p>SetKeyIterationTest</p>
33         </div>
34     )
35 }
36
37 export default SetKeyIterationTest
38
```


1) NewMapTest

```
1 const NewMapTest = () => {
2   let mapData = new Map()
3
4   console.log(mapData)
5   // 출력 : Set(0) { }
6   //       아무런 데이터 값이 없기 때문에 안뜬다.
7
8   return (
9     <div className="NewMapTest">
10      <p>NewMapTest</p>
11    </div>
12  )
13 }
14
15 export default NewMapTest
```

2) NewMapInitTest

```
1 const NewMapInitTest = () => {
2   let mapData = new Map(
3     [
4       ["apple", "red"], ["grape", "purple"]
5     ]
6   )
7
8   console.log(mapData)
9   // 출력 : {"apple" => "red", "grape" => "purple"}
10  console.log("Size: " + mapData.size)
11  // 출력 : 2
12
13  return (
14    <div className="NewMapInitTest">
15      <p>NewMapInitTest</p>
16    </div>
17  )
18 }
19
20 export default NewMapInitTest
```

3) MapSetTest

```
1  const MapSetTest = () => {
2    let mapData = new Map(
3      [
4        ["apple", "red"], ["grape", "purple"]
5      ]
6    )
7
8    console.log(mapData)
9    // 출력 : {"apple" => "red", "grape" => "purple"}
10
11    mapData.set("test1", "blue")
12    mapData.set("test2", "green")
13    // set 으로 mapData => {"test1", "blue"} ("test2", "green") 추가.
14
15    console.log(mapData)
16    // 출력 : {"apple" => "red", "grape" => "purple", "test1" => "blue", "test2" => "green"}
17    console.log(mapData.get("test2"))
18    // 출력 : green // test2 의 속성값을 출력
19    console.log(mapData.get("grape"))
20    // 출력 : purple // grape 의 속성값을 출력
21    console.log(mapData.has("test1"))
22    // 출력 : true // set과 map 객체에 주어진 요소가 존재하는지 여부를 판별해 반환
23    console.log(mapData.has("apple"))
24    // 출력 : true
25    console.log(mapData.has("해행"))
26    // 출력 : false
27
28    return (
29      <div className="MapSetTest">
30        <p>MapSetTest</p>
31      </div>
32    )
33  }
34
35  export default MapSetTest
```

Colored by Color S

4) MapIterationTest

```
1  const MapIterationTest = () => {
2      let mapData = new Map(
3          [
4              ["apple", "red"], ["grape", "purple"], ["바보", "천재"], [1, 20]
5              // 앞 = key 값 // 뒤 = value 값.
6          ]
7      )
8
9      console.log(mapData)
10     // 출력 : {"apple" => "red", "grape" => "purple", "바보" => "천재", 1 => 20}
11
12     var keyIter = mapData.keys()
13
14     for (var key of keyIter) {
15         console.log(key)
16         // 출력 : apple
17         //         grape
18         //         바보
19         //         1
20     }
21
22     var valueIter = mapData.values()
23
24     for (var value of valueIter) {
25         console.log(value)
26         // 출력 : red
27         //         purple
28         //         천재
29         //         20
30     }
31
32     var entryIter = mapData.entries()
33
34     for (var entry of entryIter) {
35         console.log(entry)
36         // 출력 : {"apple", "red"}
37         //         {"grap", "purple"}
38         //         {"바보", "천재"}
39         //         {1, 20}
40     }
41
42     mapData.forEach(
43         (value, key) =>
44             console.log(key + " : " + value)
45         // 출력 : apple : red
46         //         grap : purple
47         //         바보 : 천재
48         //         1 : 20
49     )
50
51     return (
52         <div className="MapIterationTest">
53             <p>MapIterationTest</p>
54         </div>
55     )
56 }
57
58 export default MapIterationTest
```

5) MapDeleteTest

```
1  const MapDeleteTest = () => {
2      let mapData = new Map(
3          [
4              ["apple", "red"], ["grape", "purple"]
5          ]
6      )
7
8      console.log(mapData)
9      // 출력 : {"apple" => "red", "grape" => "purple"}
10
11     mapData.delete("apple")
12
13     console.log(mapData)
14     // 출력 : {"grape" => "purple"}
15
16     mapData.clear()
17     // clear 는 모든 값을 지운다.
18
19     console.log(mapData)
20     // 출력 : { }
21
22     let mapData2 = new Map(
23         [
24             ["apple", "red"], ["grape", "purple"], ["코딩", "천재"]
25         ]
26     )
27
28     console.log(mapData2)
29     // 출력 : {"apple" => "red", "grape" => "purple", "코딩" => "천재"}
30
31     mapData2.clear()
32     console.log(mapData2)
33     // 출력 : { }
34
35     let mapData3 = new Map(
36         [
37             ["apple", "red"], [1, 999], ["코딩", "천재"]
38         ]
39     )
40
41     console.log(mapData3)
42     // 출력 : {"apple" => "red", 1 => 999, "코딩" => "천재"}
43
44     mapData3.delete(1)
45
46     console.log(mapData3)
47     // 출력 : {"apple" => "red", "코딩" => "천재"}
48
49     return (
50         <div className="MapDeleteTest">
51             <p>MapDeleteTest</p>
52         </div>
53     )
54 }
55
56 export default MapDeleteTest
```

```

1  class IteratorManager {
2      constructor() {
3          this.itrData = []
4      }
5      add(data) {
6          this.itrData.push(data)
7      }
8      //minus(data) {
9      //    this.itrData.shift(data)
10     // }
11     // Symbol.iterator 심볼은 객체에 대응하는 기본
12     // 이레이터를 지정합니다.
13     [Symbol.iterator]() {
14         return this.itrData.values()
15     }
16 }
17
18 const IteratorTest = () => {
19     let iterData = ["one", "two", "three"]
20
21     for (let entry of iterData) {
22         console.log(entry)
23         // 출력 : one
24         //         two
25         //         three
26     }
27
28     let varData = ["four", "five", "six"]
29
30     let itr = varData[Symbol.iterator]()
31
32     console.log(itr.next())
33     // 출력 : {value: "four", done: false}
34     console.log(itr.next())
35     // 출력 : {value: "five", done: false}
36     console.log(itr.next())
37     // 출력 : {value: "six", done: false}
38     console.log(itr.next())
39     // 출력 : {value: undefined, done: true}
40
41     let varItr = varData.values()
42
43     console.log(varItr.next())
44     // 출력 : {value: "four", done: false}
45     console.log(varItr.next())
46     // 출력 : {value: "five", done: false}
47     console.log(varItr.next())
48     // 출력 : {value: "six", done: false}
49     console.log(varItr.next())
50     // 출력 : {value: undefined, done: true}
51
52     return (
53         <div className="IteratorTest">
54             <p>IteratorTest</p>
55         </div>
56     )
57 }
58
59 export default IteratorTest
60

```