

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>destructuring</title>
9  </head>
10
11 <body>
12
13     <script>
14         /*
15         ,         ***** destructuring (비구조화) *****
16         ,
17         ,         destructuring은 object나 array, 그 외 요소들 안의
18         ,         변수를 바깥으로 끄집어 내서 사용할 수 있도록 하는것.
19         ,
20         ,         예를들어 우리한테
21         ,         settings라는 객체가 있고,
22         ,         누군가 유저를 팔로우하면 notification을 보낸다고 가정
23         ,         alert같은 것들을 설정
24         ,
25         ,         const settings = {
26         ,             notifications : {
27         ,                 follow : true,
28         ,                 alerts : true,
29         ,                 unfollow : false
30         ,             },
31         ,             color : {
32         ,                 theme : "dark"
33         ,             }
34         ,         }
35         ,         이런것들은 우리가 js로 코딩하면서 만들게 될 수도있는 그런 종류의 변수들이다.
36         ,         이제 우리가 하고싶은것은 일종의 체크이다.
37         ,
38         ,         만약 follow 가 true일때 이메일을 보내려고 한다면
39         ,
40         ,         if (settings.notifications.follow) {
41         ,             // send email
42         ,         }
43         ,         이런식으로 작성을 할것이다.
44         ,         이런경우는 notifications가 없거나 follow가 없거나 많은 변수 상황들에 의해
45         ,         코드가 정상작동하지 않을 경우가있다.
46         ,         이런때에 활용 할것이
47         ,
48         ,         object destructuring이다.
49         ,         내가 하고싶은것은 settings 안의 notifications의 follow를 가져올것이다.
50         ,         if 로 체크하는것이 아닌 변수를 만든다.
51         ,
52         ,         const follow = settings.notifications.follow 이런방식이 아닌
53         ,         변수 자체에 destructuring을 적용
54         ,         const follow = settings;
55         ,         이런식으로 타겟 object를 뒤에 넣어주고 이제 구조상 follow가 나올때까지 settings안을
56         ,         찾아볼것이다.
57         ,         상단방식은 접근하기 어려우니
58         ,
59         ,         const {
60             notifications : { follow }, 3단계 레벨의 안으로 접근하고 싶다면 변수에 컬럼 :
61             하고 중괄호로 묶으면 된다.

```

```

60 |         *** 여기서 중요한점은 notifications를 콘솔에 찍어보면 에러가 난다.
61 |         그 이유는 notifications자체를 변수로 가져오는것이 아니라 그 안에 있는 follow라는
값을 변수로 만드는것이다.
62 |         color
63 |     } = settings;
64 |     라는 식으로 해주면, notifications안으로 접근하고 그것은 settings안에 있는데,
follow만을 가져올것이다.
65 |
66 |     만약 color의 theme이 아니라 전체를 가져오고싶다면
67 |     color를 추가
68 |
69 |     **** 이런 방식은 큰 오브젝트에서 특정 변수나 그 안에 속한 작은 오브젝트에 접근할 수 있게
해준다
70 |
71 |     const {
72 |         notifications 이 부분이 const를 생성하는것이다.
73 |     } = settings
74 |
75 |     const notifications = settings.notifications    두 구문이 같은것.
76 |
77 |     ***** 없는것을 찾아볼때
78 |
79 |     const {
80 |         bibi : { barnes = true }
81 |     } = settings;
82 |     console.log(barnes);
83 |
84 |     이런식으로 현재 object에 없는것을 찾아보면
85 |     에러가 나온다.
86 |
87 |     이런때에 default값을 넣어줄 수 있다.
88 |
89 |     barnes = true 가 무슨뜻이냐하면
90 |     settings안의 notifications의 안으로 가서
91 |     barnes가 있는지 찾아본 다음에
92 |     없다면 true라는 값을 넣는것이다. boolean 값 뿐만 아니라 모든 타입을 넣을 수 있다.
93 |
94 |     bibi라는 것이 통째로 없다면 이런식으로도 체크가 가능
95 |     const { bibi : { barnes = false } = {} } = settings;
96 |     이것은 bibi가 없다면 default값을 설정하는것이다.
97 |     무슨 의미냐 하면
98 |     settings안을 보고 bibi 안으로 가서 barnes를 찾는것이다.
99 |     만약 barnes가 없다면 false로 설정하고
100 |     bibi 자체가 없다면 빈 object로 만들것이다.
101 |     console.log(barnes)를 찍어보면 false 가 나올것이다.
102 |     왜냐하면 지금 bibi는 빈 객체이기 때문
103 |
104 |     참조를 해서 settings에 넣고싶다면
105 |     const { bibi, bibi : { barnes = false } = {} } = settings;
106 |
107 |     이것이 'one-line-statement'이다.
108 |     이런식으로 one-line-statement를 사용하는게 if 어찌고를 사용하는것보다 좋다
109 |
110 |     target object를 적어주고 우리가 가지고 올 값의 경로를 적어주는 것
111 |     */
112 |
113 |     const settings = {
114 |         notifications : {
115 |             follow : true,
116 |             alerts : true,
117 |             unfollow : false
118 |         },

```

```
119         color : {
120             theme : "dark"
121         }
122     }
123
124     const {
125         notifications : { follow },
126         color
127     } = settings;
128     console.log(color);
129
130     // 없는것을 찾아보려고할때
131     const { bibi : { barnes = false } = {} } = settings;
132     console.log(settings);
133
134
135
136
137     </script>
138 </body>
139
140 </html>
```