

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>cloning styled components</title>
8  </head>
9  <body>
10     <script>
11         /*
12         ,          ***** styled components *****
13         ,
14         ,          styled components는 리액트를 위한 라이브러리, 패킷 그런것이다
15         ,          이것은 JS에서 css를 쓸 수 있게 해주고, html을 얻을 수 있다.
16         ,          template literals의 도움을 받아 이것을 카피해 만들어볼것이다.
17         ,
18         ,          두가지를 해볼것이다.
19         ,          첫번째는 function 이 있어야한다.
20         ,          function은 우리가 만들고싶은 element를 받고
21         ,          적용하고 싶은 css도 받은 다음에 css를 적용한 element를 return할것이다.
22         ,
23         ,          const styled = css => console.log(css);
24         ,
25         ,          styled`border-radius : 10px;color : blue;`
26         ,
27         ,          보통 styled()안에다가 넣지만 빼고 콘솔을 찍어보면
28         ,          내 css argument가 바뀌었다. 여전히 function은 실행되고
29         ,          styled()한것과 같은 결과가 나온상태이다.
30         ,          콘솔을 찍어보면 엄청 긴 argument가 나올것이다.
31         ,          string인 하나의 argument. array안에 하나만 있는
32         ,
33         ,          이것이 styled component가 동작하는 방식이다
34         ,
35         ,          **** styled`border-radius : 10px;color : blue;`
36         ,
37         ,          이것은 function을 text로 호출하는 또 다른 방법이다.
38         ,          이것이 template literal의 장점 중 하나다.
39         ,          이렇게 함수를 호출하면 string들은 argument로 들어가게 된다.
40         ,
41         ,          const styled = aElement => {
42         ,              const el = document.createElement(aElement);
43         ,              return el;
44         ,          }
45         ,
46         ,          const title = styled("h1")`
47         ,              color : blue;
48         ,              font-size : 1rem;
49         ,          `
50         ,          console.log(title);
51         ,
52         ,          이런식으로 하고 찍어보면 error가 난다
53         ,          styled(...) is not a function
54         ,          이것은 function을 두번 호출했다는 뜻이다.
55         ,          const title = styled("h1")() 이런식
56         ,          왜냐면 우리가 element를 return하기 때문이다.
57         ,
58         ,          styled("h1") 이것은 하나의 표현식이다
59         ,          <h1></h1>같은것인데 이 h1를 한번 더 부르는 셈이 되는것.
60         ,
61         ,          우리가 해야 할 것은

```

```

62 | ,           function안에 function을 return하는것이다.
63 | ,
64 | ,           element를 return하는 대신에 function을 return
65 | ,           이 function을 argument를 가져갈것이다
66 | ,           `
67 | ,           color : blue;
68 | ,           font-size : 1rem;
69 | ,           `
70 | ,           이건 string이다. (args) => {
71 | ,           console.log(args[0]); arg의 첫번째 엘리먼트로 들어오니 그것만 필요
72 | ,           const styles = args[0] 변수로 넣어주고
73 | ,           }
74 | ,           el.style = styles; element의 스타일을 지정
75 | ,           return el; element를 return
76 | ,
77 | ,           이 이후에 확인을해보면 h1을 가지고있고 style이 안에 들어가있다.
78 | ,           styled("h1")("color : blue")같이 두번째의 ()를 넣을 필요가 없어진것이다.
79 | ,           ()를 넣어도 잘 동작한다
80 | ,
81 | ,           이것이 react에서 쓰는 방식
82 | ,
83 | ,           지금까지 styled componentdml clone을 만들었다.
84 | ,           여기서 react를 다루지는 않지만 잘된다.
85 | ,
86 | ,           이것이 template literal의 힘이다.
87 | ,           */
88 |
89 | const styled = aElement => {
90 |   const el = document.createElement(aElement);
91 |   return args => {
92 |     const styles = args[0];
93 |     el.style = styles;
94 |     return el;
95 |   };
96 | }
97 |
98 | const title = styled("h1")`
99 |   color : blue;
100 |   font-size : 1rem;
101 | `;
102 |
103 | const subTitle = styled("span")`
104 |   color : green;
105 | `;
106 | title.innerText = "we just cloned";
107 | subTitle.innerText = "styled components";
108 |
109 | document.body.append(title, subTitle);
110 |
111 | </script>
112 | </body>
113 | </html>
114 |

```