

```

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>super part two</title>
8  </head>
9  <body>
10
11      <div>
12          <h1>Count One</h1>
13          <span id="count"></span>
14          <button id="add">Plus 1</button>
15          <button id="minus">Minus 1</button>
16      </div>
17
18      <div>
19          <h2>Count Two</h2>
20          <span id="count2"></span>
21          <button id="add2">Plus 1</button>
22          <button id="minus2">Minus 1</button>
23      </div>
24
25      <script>
26          /*
27          ,
28          ,          ***** this *****
29          ,
30          ,          전 파일에서 우리는 이상한 에러에 직면했다.
31          ,          this.repaintCount는 HTML button element의 function이 아니다. 라는 에러가 나온다.
32          ,
33          ,          일단,
34          ,          addEventListener()함수 바로 안에
35          ,          console.log(this)를 찍어보면 잘 출력이되고있다.
36          ,          하지만 increase 함수 this가 button으로 찍히고있다.
37          ,
38          ,          따라서 몇가지 이유때문에 this는 변한다. 더이상 class를 가르키고 있지 않고 button을 가르킨
39          ,          다. 이것이 자바스크립트의 기본적인 동작이다. button이나 어떤것에 eventlistener를 추가하고 함수
40          ,          를 호출하면 자바스크립트는 this를 변경시키고, this는 element를 가르킨다.
41          ,
42          ,          * 그냥 this를 하게되면 window를 가르키고,
43          ,          현재 이벤트 리스너 안에서는 counter를 가르키고,
44          ,          increase를 호출한 안에서 this를 하면 이벤트가 등록된 element를 가르킨다.
45          ,
46          ,          * 내가 event listener를 만들면 그 event 의 function이 this를 Copy할것이다.
47          ,          event를 호출한 element안으로.
48          ,
49          ,          * 만약 this를 항상 class를 가르키게 하고싶다면 (몇개의 이벤트리스너가 생기던 상관없이)
50          ,          함수를 arrow function으로 바꾸어주면된다.
51          ,
52          ,          this 키워드 사용법, 규칙이 많다 이것도 그중 하나이다.
53          ,          eventlistener를 target에 추가할때 이 event의 handler는 this를 event target에 가르
54          ,          키게 한다
55          ,          클래스 없이 만들어도 동작할것이다. 타겟을 만들고 이벤트함수들을 놓고,
56          ,          기존에 하던방식대로.
57          ,          이건 단지 코드를 조직하는 다른방식이다 다른 규칙과함께
58          ,

```

```

59 ,      다른점은 class를 파일에 넣고 import하면 원하는 만큼
60 ,      class를 만들어낼 수 있다.
61 ,      class와 함께하면 blueprint를 만들어야한다.
62 ,      같은 것을 재사용할 수 있으니 원하는 만큼 만들 수 있다.
63 ,
64 ,      새로운 Counter를 만들어서 argument에 전달해주는것을 보내주면 새로운 Class로 등록이된다.
65 ,      그리고 argument를 바꾸고싶다하면 예를들어,
66 ,      initialNumber의 시작이 0이아닌 666으로 시작하고싶다.
67 ,      그렇다면 Counter안에 선언해주면된다.
68 ,      this.counter.innerHTML = initialNumber 기본값을 바로 출력해놓고 시작해보자.
69 ,
70 ,      ** 정리
71 ,
72 ,      우리는 보다시피 재사용 할 수 있는것을 만들었다.
73 ,      매우 조직적이고 원한다면 파일에 넣어 프로젝트 어디서든 쓸 수 있다.
74 ,      이것이 우리가 class를 사용하는 이유이다.
75 ,      함수를 사용하면 똑같이 만들수는 있지만 이것처럼 재사용은 불가능할것이다.
76 ,      */
77
78
79 class Counter {
80
81     constructor({ initialNumber = 0, counterId, plusBtn, minusBtn }) {
82         this.count = initialNumber;
83         this.counter = document.getElementById(counterId);
84         this.counter.innerHTML = initialNumber
85         this.plusBtn = document.getElementById(plusBtn);
86         this.minusBtn = document.getElementById(minusBtn);
87         this.addEventListeners();
88     }
89     addEventListeners = () => {
90         this.plusBtn.addEventListener("click", this.increase);
91         this.minusBtn.addEventListener("click", this.decrease);
92     }
93     increase = () => {
94         this.count = this.count + 1;
95         this.repaintCount()
96     }
97
98     decrease = () => {
99         this.count = this.count - 1;
100        this.repaintCount()
101    }
102    repaintCount = () => {
103        this.counter.innerText = this.count;
104    }
105 }
106
107 new Counter({
108     counterId : "count",
109     plusBtn : "add",
110     minusBtn : "minus"
111 });
112
113 new Counter({
114     counterId : "count2",
115     plusBtn : "add2",
116     minusBtn : "minus2",
117     initialNumber : 666
118 })
119

```

```
120 | </script>
121 | </body>
122 | </html>
```