

```

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>weak sets</title>
8  </head>
9  <body>
10     <script>
11         /*
12         ,
13         ,     ***** sets *****
14         ,     weak sets는 sets과 같은듯 다르다.
15         ,     차이점은
16         ,
17         ,     1. 예를들어 numbers, text, array 같은 것들을 저장하지 못하고
18         ,     오로지 object만 저장할 수 있다.
19         ,
20         ,     2. 다음과 같은것들이 없다.
21         ,     size, entries, properties ... 우리가 보았던것들.
22         ,     add,delete,has 등은 가지고있다.
23         ,     normal set에서는 weak set보다는 많은 것들을 가지고있다.
24         ,     이유는 weak set은 오로지 object에서만 동작하기 때문.
25         ,
26         ,     * garbage collection
27         ,     기본적으로 모든 나의 변수 모든것이 메모리에 저장되어있다.
28         ,     컴퓨터내의메모리. 가끔 우리가 생성한 변수나 함수를 사용하는데
29         ,     사용하지않은 상태로 오랫동안 머무를 수 있고 아무도 그것을 호출하지 않을때가 있는데
30         ,     그때에서 공간을 차지하게된다. 이런 경우에 자바스크립트같은 언어가 garbage collector를 제공한다.
31         ,     이것은 들여다보는것인데 어떤 object 그리고 어떤 변수가 더이상 사용되지 않는지
32         ,     참조되지않거나 닿지않거나 쓸모가없어지면 컴퓨터 메모리에서 삭제제한다.
33         ,
34         ,     내가 weak set에 집어넣은 모든것들은 약하게 붙들려 있다.
35         ,     weakly held > 이 의미는 만약에 object가 없다면, 아무것도 없다면.
36         ,     내가 weak set에 넣은 objects를 가르키는 것이 없다면 이것은 지워질것이다.
37         ,
38         ,     만약 내가 object를 가지고 있다면. 이것을 weakSet에 추가해보자.
39         ,     const sexy = {
40         ,         im : true
41         ,     };
42         ,     weakSet.add(sexy);
43         ,
44         ,     object의 참조형태로 존재하고있다. 이것이 사라지지 않는 이유는
45         ,     garbage collector가 지우지않기 때문.
46         ,     그래서 sexy는 어디가지않는다.
47         ,     무언가가 set 내부의 Object를 가르키고 있기 때문.
48         ,
49         ,     weakSet.add({hello : true})
50         ,     이것은 set 내부에서 생성된 object이다.
51         ,     그래서 이 object는 외부코드에 있지 않고 set 내부에 있다.
52         ,     set 밖에 어떤것도 이 object를 가리키지 않고 hello : true에 대한 어떠한 참조도 존재하지 않는
53         ,     다.
54         ,
55         ,     sexy같은 경우는 sexy가 Object를 가리키고있다.
56         ,     하지만 지금 경우는 아무도 이것을 가리키지 않는다.
57         ,     일단 추가를 하고 보면 2개의 object가 있다.
58         ,
59         ,     garbage collector가 오면 어떻게 될까. hello : true를 가져갈것이다.
60         ,     아무도 가리키고 있지 않은 object를 가져감.

```

```
61 ,      memory의 collect garbage를 클릭해보고
62 ,      콘솔에 다시 찍어보면 hello true는 없어진다.
63 ,      현재는 내 코드를 읽어서 안바뀌는것같지만
64 ,      콘솔에서 add를 시작으로 만들어보면 없어지는것을 확인할 수 있다.
65 ,
66 ,      ** 정리
67 ,      weak set의 use case는 매우 적다. 무언가 이유가 있을것같지만. 이용을 안한다.
68 ,      만약 나의 object를 가리키는 무언가가 없다면 object는 garbage collector가 언젠가 가져갈것이다
69 ,      그래서 weak set을 신뢰하지 않는다.
70 ,
71 ,      */
72     const weakSet = new WeakSet();
73     // weakSet.add(1) 오류가 나옴.
74     // weakSet.add({hi:true}) 가능.
75     const sexy = {
76       im : true
77     };
78     weakSet.add(sexy);
79     weakSet.add({hello : true})
80
81     </script>
82 </body>
83 </html>
```