```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Sets</title>
</head>
<body>
  <script>
    /*
,
,       ***** Sets *****
,
,       알다시피 자바스크립트는 object를 가지고 있다.
,       그리고 object를 사용해서 object의 property를 삭제하거나 얻거나 추가, 있는것을 확인할 수 있다.
,       하지만 멋진느낌은 아니다.
,
,       현재의 object는
,       const user = {
,         age : 12,
,         name : "bibi"
,       }
,
,       user.oneMore = true; // add
,       user.name = null; // delette
,       이런식으로 추가하거나 지울수있다. 이것이 set을 만든 이유같다.
,       왜냐하면 set은 object와 비슷하다.
,
,       * Mozilla가 set에 대해 정의한 내용.
,       set을 사용하면 어떤 타입의 고유한 value든 저장할 수 있게 해준다.
,
,       이들은 unique해서 동일한 두 value를 저장할 수 없다.
,       * key value가 아님.
,
,       예를들어 user라는 objectt에서 age의 value가 12인데,
,       12라는 value를 다른 key에도 있다고 가정하고, 2개의 12로 set을 만들수없다.
,       unique하지 않기 때문
,
,       const sexySet = new Set([1,2,3,4,5,6,7,7,7,7,7,8])
,       new Set을 대입하고 values, value는 배열로 주어질 수 있고 한 value로 줄수도있다.
,       로 찍어보면 중복된 7은 무시되고 하나만 나오게된다.
,       그래서 이것은 unique한 value를 저장하는 것이다.
,
,       만약 내가 중복된 값을 저장하지 못하게하는 매커니즘이 필요한다면 아주 잘 쓰일것이다.
,       현재 자바스크립트에서 object의 value를 체크해야한다면, 이것이다. 고유한 value를 위한것
,
,       ** Set을 위한 API
,       콘솔창에 sexySet을 찍어보면 많은것들이 있지만 그들 중 하나가 'has'다.
,       sexySet.has(9) 이런식으로 찍어보면 false로 나오게된다.
,       우리는 이것들을 object에 가지고 있지않다.
,       우리가 가지고 있지않은것(api)를 가지고 있는 object를 가지는 상상을해보자.
,       이것이 set을 좋아하는 이유다.
,       delete로 지울수도있음. 어디에 위치하는지 지정할 필요도없고
,       그냥 value로 지울 수 있음.
,       또한 sexySet.claer() 이런식으로하면 안에 들어있는 value값을 초기화 시킬 수 있다.
,
,       이제 sexySet을 파괴했으니 추가할수 있다.
,       sexySet.add("Hi") 배열로도 추가할수있음.
,
,       * size가 length랑 같은 맥락임.
```

```
,       sexySet.keys()를 하면 모든 value를 주는데 iterator 내에 있어서 이론적으로
,       무언가 붙여서 할 수 있을것이다.
,
,       ** 정리
,       당장 내가 원한다면 몇가지 object를 저장할 수 있을것이다.
,       object를 delete했다 add 했다 할 수 있고, 실제로 object를 저장할 수 있다.
,       쉽게 말하면 작은 데이터베이스로 취급할 수 있다.
,       옵션은 끝이 없고 방금 얘기한것들은 기능중 몇개일 뿐임.
,
,     */

    // before object
    const user = {
      age : 12,
      name : "bibi"
    }

    user.oneMore = true; // add
    user.name = null; // delette
    console.log(user);

    // set
    const sexySet = new Set([1,2,3,4,5,6,7,7,7,7,7,8])
    console.log(sexySet)

    // api
    console.log(sexySet.has(9))
    sexySet.delete(5)
    console.log(sexySet)
    sexySet.clear()
    console.log(sexySet)
    sexySet.add("Hi")
    console.log(sexySet);

  </script>
</body>
</html>
```