```
<!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>symbol</title>
</head>
<body>


  <script>
    /*
,
,         ***** Symbol *****
,         자바스크립트는 Data type들을 가지고있는데 Symbols는 새로운 Data type이다.
,         고유한 Data type.
,         한 Symbol은 다른 Symbol들에 대해서 고유한 성질을 영원히 가진다.
,
,         const hello = Symbol();
,         const bye = Symbol();
,         console.log(hello == bye) // false
,
,         고유한 성질때문에 둘은 같지 않다고 나오게된다.
,
,         ** Usecase
,
,         1. Uniqueness
,         Symbols는 생성자에 한가지를 가지는데 Description을 가진다.
,         argument에 Description을 넣을수있는데, Description은 value가 아니다.
,
,         그래서 여기서 할 수 있는 일을 예로들어보면
,         예를들어 Object가 있고 매우 큰 경우에 이 Object는 사용자정보가 저장된다고해보자.
,         const superBig = {
,           bibi : {
,             handsome : true
,           }
,         };
,         문제는 만약에 어딘가에 반복되는 bibi가 있다면,
,         우리는 bibi의 값을 바꿀때에 생길것이다.
,
,         superBig["bibi"].handomse = false; 이런식으로 하거나, 아니면
,         새로운 bibi가 있다면 누군가가 한가지를 추가하면 어떨까.
,         const superBig = {
,           bibi : {
,             age : 24
,           }
,         };
,         현재 객체에는 bibi가 age를 가지고있고, handsome을 가지고있지않는다.
,         * bibi라는 key는 unique하지 않기 때문에 우리가 할 수 있는일은 symbol을 넣는것이다.
,         그래서 이제 다른사람들이 무엇을 하든 저 Symbol bibi를 절대 변겨ㅓㅇ할 수 없다.
,
,         const superBig = {
,           [Symbol("bibi")] : {
,             handsome : true
,           },
,           [Symbol("bibi")] : {
,             age : 24
,           }
,         };
,         이런식으로 하더라도 이 둘은 서로 다르다.
```

```
,         콘솔으로 찍어보며녀 두 Symbold이 나온다.
,         이 점은 내가 나만의 Object에 매우 고유한 부분을 만들 수 있게 한다.
,         누군가 바꾸거나 반복하기를 원하지않을때에.
,
,         2. Privacy
,
,         const superPrivacy = {
,           [Symbol("bucky")] : {
,             bboy : true
,           },
,           hiphop : "no"
,         }
,         console.log(Object.keys(superPrivacy))
,
,         이런 객체가 있다고 가정하고 Object.keys를 찍어보면
,         hiphop이라는 key만 있다고나오게된다. Symbol은 고려되지 않는것같다.
,         나만의 Privacy properties를 원한다면 Symbol이 나을수도있다.
,         하지만, 지금의 자바스크립트는 Symbols가 처음에 나왔을때의
,         privacy한 느낌이 많이 없어졌다.
,         Object.getOwnPropertySymbols(superPrivacy)라고
,         찍어보면 Symbol이 출력됨.
,
,         여기서 value를 얻고싶다면
,         const s = Object.getOwnPropertySymbols(superBig);
,         s.forEach(symbol => console.log(superBig[symbol]))
,         이렇게 forEach를 돌려서 가져올 수 있다.
,
,     */

    // symbol
    const hello = Symbol("hello");
    const bye = Symbol("hello");
    console.log(hello == bye) // false

    // uniqueness
    const superBig = {
      [Symbol("bibi")] : {
        handsome : true
      },
      [Symbol("bibi")] : {
        age : 24
      }
    };
    console.log(superBig)

    // privacy
    const superPrivacy = {
      [Symbol("bucky")] : {
        bboy : true
      },
      hiphop : "no"
    }
    console.log(Object.keys(superPrivacy))

    // get value
    const s = Object.getOwnPropertySymbols(superBig);
    s.forEach(symbol => console.log(superBig[symbol]))

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