```
<!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>generator</title>
</head>
<body>
  <script>
    /*
,
,       ***** generator *****
,
,       기본적으로 pause할 수 있는 함수.
,       그래서 generator를 사용하기 위해선 어떤 규칙이 있다. async await처럼
,
,       우리가 할것은 function에 *을 붙이는것이다.
,       *를 넣으면 한 단어를 해제하게된다.(unlock 쓸수있게)
,
,       yield라고 하는것인데, return과 같다.
,
,       function* listPeople(){
,         yield "Dal";
,         yield "bibi";
,         yield "nico";
,         yield "kkamzzu"
,       }
,       const listG = listPeople();
,
,       listG를 찍어보면 listPeople을 얻는다, 뒤에 suspended라고 나오는데 object가 suspended라는
뜻.
,       단지 suspended라고 나오고 return해주지 않는다.
,
,       listG.nex()를 찍어보면 첫번째 value인 Dal을 return해준다.
,       나에게 정보를 주는데 이건 단지 Dal만 return하지 않는다.
,       value를 return하는데 word value를 가진 object.
,       done이 false라는 것은 아직 끝나지 않았다는것.
,       계속해서 호출한다면 그 다음 value들을 얻고 마지막것을 하면
,       return값이 undefined가 나오고 done은 true가 나온다.
,       더이상 호출해도 값이 찍히지 않음.
,
,       * 보통 generator를 위한 usecase는 많지않다.
,
,       generator는 많은 개발자들이 사용하지 않는다. 자바스크립트의 뒷배경에서 사용될것이다
,       async await는 generator이다.
,       generator의 꼭대기에 만들어졌고, 기본적으로 generator이다.
,       우리는 뭔가를 기다리고 value를 yield하고 반복한다.
,
,       * friends라는 array가 하나있다고 가정.
,       나를 위한 friend를 작성하고, 우리가 할 것은 함수를 작성한다.
,
,       const friends = ["bibi", "hello", "kkamzzu", "dozi", "thisisneverthat"]
,       function* friendTeller(){
,         for (const friend of friends) {
,           yield friend;
,         }
,       }
,
,       const friendLooper = friendTeller()
,
```

```
,       ** generator는 보통 많이 사용하지않고,
,       라이브러리를 사용한다.
,       복잡하진 않다. 기본적으로 나는 실행을 중지할 수 있고
,       내가 next를 호출할 때 나는 뭔가 얻게된다.
,       이것들은 value로 오지않고 object로 온다.
,
,       이게 끝났는지를 체크할 수 있는 어떤 value든 선언할 수 있다.
,
,
,
,     */

    //
     function* listPeople(){
         yield "Dal";
         yield "bibi";
         yield "nico";
         yield "kkamzzu"
     }
     const listG = listPeople();

     // array
     const friends = ["bibi", "hello", "kkamzzu", "dozi", "thisisneverthat"]
     function* friendTeller(){
       for (const friend of friends) {
         yield friend;
       }
     }

     const friendLooper = friendTeller()

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