

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>rest parameters</title>
9  </head>
10
11 <body>
12
13     <script>
14         /*
15         ,           ***** rest parameters *****
16         ,
17         ,           parameter(매개변수)라는것은 우리가 함수에게
18         ,           전달하는 인자들을 이야기하는 것이다.
19         ,
20         ,           그럼 rest parameter은 무엇일까
21         ,           이것을 제일 잘 설명하려면 끝도없는 parameter를 전달받는 함수를 만들어보면 된다,
22         ,
23         ,           알다시피 spread에서 ... 이 점 세개는 변수를 전개하는 것이다.
24         ,           rest에선 축소(contract) 시킨다.
25         ,
26         ,           const infiniteArgs = (...kimchi) => console.log(kimchi);
27         ,
28         ,           infiniteArgs(
29         ,               "bibi",
30         ,               1,
31         ,               true,
32         ,               false,
33         ,               null,
34         ,               [1,2,34,4],
35         ,               "blabla"
36         ,           )
37         ,
38         ,           간단하게 이런식으로 짜면 infiniteArgs함수에 엄청나게 많은 인자들을 보내준다
39         ,           전같으면 const infiniteArgs = ({ first, second ...}) => ...; 비구조화 하는것
처럼
40         ,           이런식으로 썼겠지만
41         ,           하나의 변수를 만들어 ... 을 찍어주면 그 안에 몇개의 인자를 받아주던
42         ,           한 배열로 만들어서 반환해준다.
43         ,
44         ,           ** rest는 모든 값을 하나의 변수로 축소(contract)시킨다.
45         ,           우리가 해준것은 똑같이 ... 를 찍은것 뿐이지만
46         ,           저 의미는
47         ,           무슨 값이 있던간에 전부다 kimchi라고 불리는 변수안에 넣자. 라는 뜻.
48         ,
49         ,           *** rest 기능의 핵심
50         ,
51         ,           절친 리스트가 있다고 가정해보자. 거기서 제일 친한 첫번째 친구만을 빼고싶을때
52         ,           const bestFriendsMaker = (firstFriend, ...rest) => {
53         ,               console.log(`my best friend is ${firstFriend}`);
54         ,               console.log(rest);
55         ,           };
56         ,           bestFriendsMaker("bibi", "kkamzzu", "chocho", "ed")
57         ,
58         ,           rest(나머지)라는 변수를 많이 쓰기도 하지만 원하는 변수를 넣어도된다.
59         ,           저런식으로 순번에 맞게 가져오고 rest(나머지)는 다른곳에다가 저장할 수 있다.
60         ,

```

```
61 | ,          ** rest 는 array를 만든다
62 | ,          ** ... sparead와도 같은 문법이지만 * parameter에 들어가게되면 rest 구문으로 실행
63 | ,
64 | ,          */
65 |
66 | // rest example
67 | const infiniteArgs = (...kimchi) => console.log(kimchi);
68 |
69 | infiniteArgs(
70 |     "bibi",
71 |     1,
72 |     true,
73 |     false,
74 |     null,
75 |     [1,2,34,4],
76 |     "blabla"
77 | )
78 |
79 | // rest friends
80 | const bestFriendsMaker = (firstFriend, ...rest) => {
81 |     console.log(`my best friend is ${firstFriend}`);
82 |     console.log(rest);
83 | };
84 | bestFriendsMaker("bibi", "kkamzzu", "chocho", "ed")
85 | </script>
86 | </body>
87 |
88 | </html>
```