

```

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>arrow function</title>
8  </head>
9  <body>
10     <script>
11         /*
12         ,         ***** arrow function *****
13         ,
14         ,         arrow function은 자바스크립트에서 함수의 모습을 개선한것이다.
15         ,         var처럼 let과 const가 대체되는것은 아니고 보기좋게 만든것.
16         ,
17         ,         기존 function을 만들려면
18         ,         function nameOf(arg){
19         ,
20         ,         }
21         ,         이런식으로 만들거나, 이름을 빼고 익명함수 혹은
22         ,         const hello = function(arg){...}
23         ,         처럼 상수 안에 넣을 수 있었다.
24         ,
25         ,         arrow function은 기본적으로 => 의 형태이다.
26         ,         좀 더 코드를 보기 쉽게 만든것.
27         ,
28         ,         ***** map() *****
29         ,         map은 function을 줄것이다 그리고 모든 item들을 가져와서
30         ,         function을 실행시켜줄것이다.
31         ,
32         ,         map이 하는 일은 각각의 아이템마다 함수를 호출하는 일을 한다.
33         ,
34         ,         const names = ['bibi', 'kkamzzu', 'barnes'];
35         ,         const heartz = names.map(function(item){
36         ,             return item + "♥";
37         ,         });
38         ,
39         ,         map은 함수를 호출하고 각각의 item들을 인자로 넣어줌.
40         ,         잊지말아야하는것이 여기서 map은 어떤것을 return해야한다.
41         ,         return값을 새로운 array로 만들.
42         ,         0 로 하든 어쨌든 새로운 array로 만들.
43         ,
44         ,         const heartz = names.map((item, index) => {
45         ,             })
46         ,
47         ,         map function의 두번째 argument는 index 이다.
48         ,         argument가 하나라면 (item)이 아닌 item => 로 적어주어도 됨.
49         ,
50         ,         ***** implicit return *****
51         ,
52         ,         arrow function은 기능이 향상되었다.
53         ,         주로 function을 다루다보면 어떤것에 적용을 시키는데
54         ,         그것은 주로 복잡한 내용을 가지고 있지는 않다.
55         ,         표현식 할줄만 있을때라고 가정하면 return을 쓰거나
56         ,         {}를 쓰기에느낌 과한것같다.
57         ,
58         ,         그래서 이경우 arrow function은 implicit return을 한다.
59         ,
60         ,         const heartz = names.map(item => item + "♥");
61         ,         이런식으로 된다. 이것은 item이라는 argument를 가진것을

```

```
62 | , 리턴하는 함수이다.
63 | , 괄호를 지우고 세미콜론, return function을 지우고
64 | , 가장 중요한 부분만남김
65 | ,
66 | , implicit return 의 뜻은 같은 줄에 뭘 적든지 간에
67 | , return 된다는 의미.
68 | ,
69 | , *** 괄호를 추가하는 순간 implicit return은사라짐
70 | , 내용이 있고 함수가 좀 복잡해지면 function의 body를 만들게 되고
71 | , 괄호가 추가되니 implicit return 기능이 사라지게된다.
72 | , */
73 | /*
74 | , const names = ['bibi', 'kkamzzu', 'barnes'];
75 | , function addHeart(item) {
76 | ,     return item + "♥";
77 | , }
78 | , const heartz = names.map(addHeart);
79 | ,
80 | , 요즘은 다른 함수를 만들어 넣지않는다.
81 | , 혼란을 줄 때가 있기 때문에 바로 적기를 원한다.
82 | , 거기서 arrow function이 개선되서 나오게 된것이다.
83 | , */
84 |
85 |     const names = ['bibi', 'kkamzzu', 'barnes'];
86 |     const heartz = names.map(item => item + "♥");
87 |     console.log(heartz);
88 | </script>
89 | </body>
90 | </html>
91 |
```