


<script>

//함수자체도 기능의 추상화 임

(즉 밀크커피만들기() 기능은 밀크커피가 어떻게 만들어지는지 세부사항은 노출되지 않음, 알수없음) :

그래서 함수를 블랙박스라는 표현을 사용하기도 함 입력->  → 출력

//고차함수

고차 함수는 함수 추상화를 한 단계 높인 것

- 함수를 통해 얻은 추상화를 한 단계 높인 것이 고차 함수이다.
- 고차 함수(higher order function)는 함수를 인자(argument)로 받을 수 있고, 함수의 형태로 리턴할 수 있는 함수이다.

```
function fun1(){
  let str="";
  for( let i=0; i<5; i++){
    str += i + "-";
  }
  return str;           // 0-1-2-3-4- 반환
}
```

```
function fun2(){
  let str="";
  for( let i=0;i<3; i++){
    str+= i+", ";
  }
  return str;           // 0,1,2, 반환
}
```

```
console.log( fun1());
console.log( fun2());
```

```
function higherF( n, f){    //반복횟수,실행할 코드를 함수를 받겠다
  let str="";
  for( let i=0;i< n;i++){
    str +=f(i);
  }
  return str;
}
```

```
let rtn1= higherF( 5, function( i){
  return i+"-";
} );
```

```
let rtn2= higherF( 3, function( i){
  return i+", ";
} );
```

```
console.log( rtn1 );
console.log( rtn2 );
```

fill()매서드

```
// arr.fill(value [, start] [, end])    대괄호 묶은것은 생략가능  
// start 인덱스에서 end인덱스 전까지 범위  
// 배열생성시 undefined로 채워짐
```

```
// 배열의 요소를 특정값으로 초기화할때 사용함
```

```
let arr = new Array(5).fill(0);  
alert( arr);
```

```
let arr2 = new Array(5).fill(5);  
alert( arr2);
```

```
let arr3 = new Array(5).fill(7, 2,4) ; // 3번째에서 4전까지  
alert( arr3);
```

// map매서드

// 배열의 요소를 callback을 사용하여 차례대로 처리한다
// callback에서 반환한 값으로 새로운 배열을 만들어서 리턴함

// map(callback)
// 인수로 넘긴 함수에는 인수 세개(value, index, array)가 전달되며 반드시 return 해야함
// 순서대로 매개변수를 작성하면됨
// 필요없는 정보는 받지 않아도 됨
// 매개변수의 순서를 지켜야 함

let arr=[2,4,6,8];

let newArr1 = arr.map(function(item){
 return item * item
});
//[4,16,36,64]

let newArr2 = arr.map(function(item, index){
 return item * index;
});
//[0, 4,12,24]

let newArr3 = arr.map((v) =>v*v) ;
// [4,16,36,64]

alert(arr) ;
alert(newArr1) ;
alert(newArr2) ;
alert(newArr3) ;

//map사용시 주의사항

//초기화되지 못한 배열은 map을 사용못함

// 아무것도 없는값이라도 채워야 map을 사용할 수 있음

let arr3 = new Array(10).fill().map((value, index)=>index+1);
alert(arr3);

splice() 매서드

<script>

```
// 이어붙이기|splice
// splice( index,0, item)    // 추가하기
// splice( index, 1)        // 삭제하기

let arr = ["one","two","three","four"];
let removed = arr.splice( 0,1);    // 제거된 배열을 배열로 반환함
                                     // 0번째 요소에서 1개 삭제

alert( removed[0]);                // 삭제된 요소 확인함
alert( arr);                       // 삭제된 후의 배열을 확인할수 있음 총 개수 3개
// ["two","three","four"];

arr.splice(0,0, "star");    //0번째 요소 왼쪽에 추가함을 의미 "star"를
alert( arr);
// ["star","two","three","four"];
```

</script>

push()매서드

```
<script>

    //배열의 끝에 요소를 추가함
    let arr= [ ];

    arr.push(5);
    alert( arr);

    arr.push(3);
    alert( arr);

</script>
```

//pop()매서드

```
<script>
    let arr = [9,5,3,5];
    alert(arr);

    arr.pop(); // 배열의 마지막 요소 꺼내기 , 배열이 줄어듦
    alert( arr ) ;

    arr.pop(); // 배열의 마지막 요소 꺼내기 , 배열이 줄어듦
    alert( arr ) ;

    arr.pop(); // 배열의 마지막 요소 꺼내기 , 배열이 줄어듦
    alert( arr ) ;

    arr.pop(); // 배열의 마지막 요소 꺼내기 , 배열이 줄어듦
    alert( arr ) ;

</script>
```

