

배영

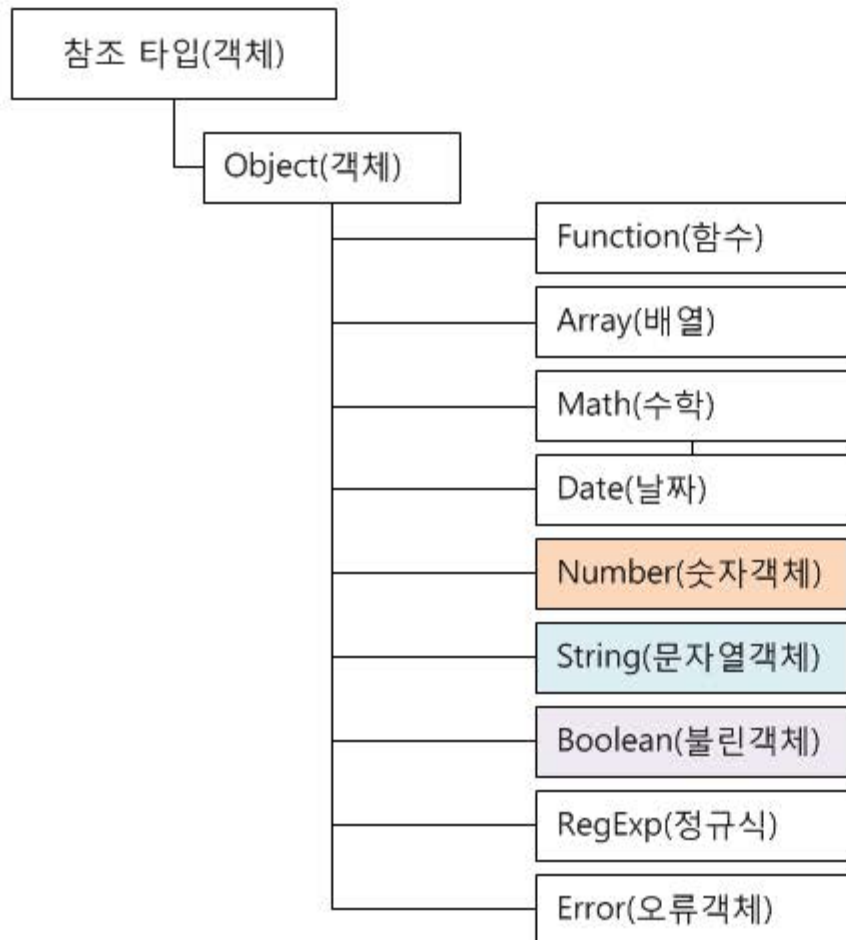




- ✓ 배열이란?
 - 변수들의 모임
- ✓ 배열 선언, 초기화, 대입
 - 배열 리터럴(`[]`)를 이용하는 방법 `var array = [] ;`
 - 배열의 인덱스는 0부터 시작
- ✓ 배열 요소 CRUD(create/read/update/delete)
 - 추가 / 읽기 / 수정 / 삭제
- ✓ 배열 요소 열거
 - for 를 사용하는 경우
 - for in 을 사용하는 경우
- ✓ Array 객체의 프로퍼티와 메서드
 - .length : 배열 길이
 - .splice() : 배열의 일정 부분을 추출(삭제)
 - .join() : 특정 문자를 사용해 모든 요소 내용을 문자열로 반환
 - .sort() , reverse() : 배열의 요소의 값을 정렬



참조 타입



- 배열

```
var array1 = [ 2, 3, 4.5, 6.78 ];  
var array2 = [ 'a', 'abc', 'ef' ];  
var array3 = [ 2, 'abc', true ];
```

- 함수

```
var log = function () {  
    console.log('It is a function');  
};
```

- 객체

```
var obj = { name : 'you', age:30 };
```





왜 배열을 사용해야 하는가?

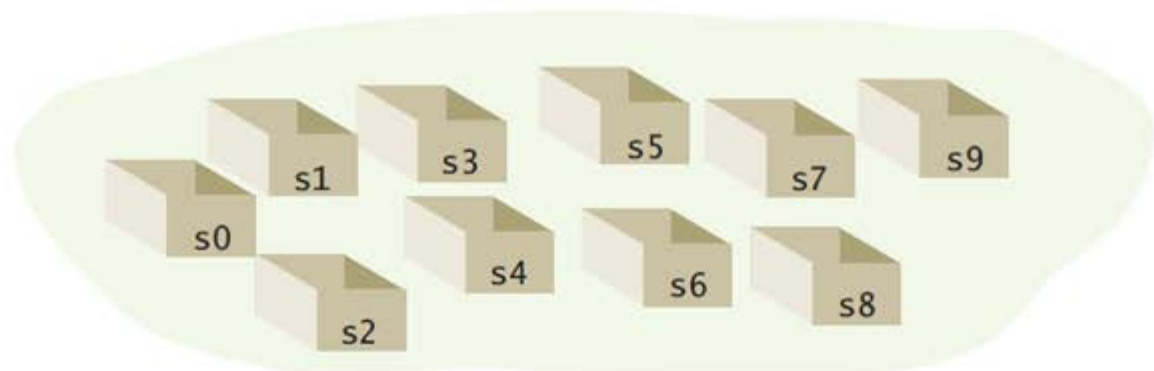
- 6 개의 정수를 입력 받고 합계를 구해서 결과를 출력하는 프로그램을 작성하시오.
- 12 개의 정수를 입력 받고 합계를 구해서 결과를 출력하는 프로그램을 작성하시오.
- 24 개의 정수를 입력 받고 합계를 구해서 결과를 출력하는 프로그램을 작성하시오.
- 100 개의 정수를 입력 받고 합계를 구해서 결과를 출력하는 프로그램을 작성하시오.





배열이란?

- 배열(array): 변수 여러 개를 한꺼번에 다룰 수 있게 자료형
 - 변수들의 모임
 - 객체 리터럴 ([])을 사용해 생성
 - 배열의 인덱스는 0부터 시작. **1이 아니고**



배열은 변수들을 모아놓은 것
배열은 하나의 이름을 공유한다.

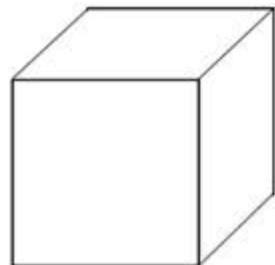


배열은 변수들의 모임이다.

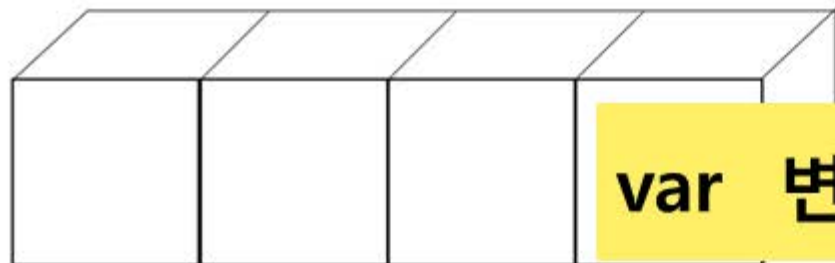




배열의 선언



var 변수명 = 초기값;

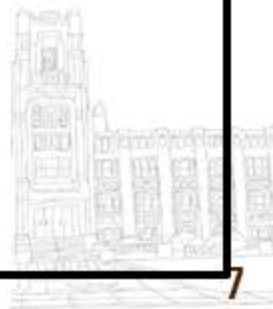


var 변수명 = [초기값, ...];

- 배열의 선언:
 - 배열 리터럴을 이용하는 방법
`var arr = [];`
 - Array 생성자를 이용하는 방법
`var arr = new Array();`



```
<script>  
  // 객체 리터럴을 이용한 배열 선언  
  var arr = [];  
  console.log( arr );  
  console.log( arr.length );  
  console.log( arr[0] ); // (출력값)  
  console.log( arr[1] ); // (출력값)  
  console.log( arr[2] ); // (출력값)  
  
  var arr1 = [1, 2, 3, 4, 5];  
  console.log( arr1 );  
  console.log( arr1.length );  
  console.log( arr1[0] );  
  console.log( arr1[1] );  
  
</script>
```





배열의 초기화

- 배열의 선언: 배열 리터럴
 - `var arr1 = [2, 3, 4.5, 6.78];`
 - `var arr2 = [2, 'abc', true];`
 - `var arr3 = ['JavaScript', 'jQuery', 'prototype', 'ASP.NET'];`
- 배열의 요소로 어떤 자료형도 넣을 수 있다.
 - 숫자, 문자, 불린, undefined, null, 배열, 객체, 함수 등이 가능.
 - `[273, 'abc', true, [57, 39.23], { name : 'you' }, function () {}];`

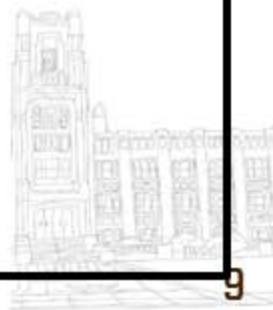



```
<script>
```

```
var arr2 = [2, 'abc', true];  
console.log( arr2 );  
console.log( arr2.length );  
console.log( arr2[0] );  
console.log( arr2[1] );  
console.log( arr2[2] );
```

```
var arr3 = ['JavaScript', ['jQuery', 'prototype'], 'ASP'];  
console.log( arr3 );  
console.log( arr3.length );  
console.log( arr3[0] );  
console.log( arr3[1] );  
console.log( arr3[2] );
```

```
</script>
```





배열 요소 추가(C)

- 배열 리터럴을 이용한 배열의 선언

```
var numbers = [];
```

```
numbers[0] = 0;  
numbers[1] = 1;  
numbers[2] = 2;  
numbers[3] = 3;  
numbers[4] = 4;  
numbers[5] = 5;
```



```
// for 문으로.
```

```
for (var j = 0; j <= numbers.length-1; j=i+1) {  
    console.log('numbers[' + j + '] = ' + numbers[j] );  
}
```



```
<script type="text/javascript">
```

```
var numbers = [];
```

```
numbers[0] = 0;
```

```
numbers[1] = '11111';
```

```
numbers[2] = true;
```

```
numbers[3] = undefined;
```

```
numbers[4] = null;
```

```
numbers[5] = [11, 12, 13];
```

```
console.log('numbers[0] = ' + numbers[0]);
```

```
console.log('numbers[1] = ' + numbers[1]);
```

```
console.log('numbers[2] = ' + numbers[2]);
```

```
console.log('numbers[3] = ' + numbers[3]);
```

```
console.log('numbers[4] = ' + numbers[4]);
```

```
console.log('numbers[5] = ' + numbers[5]);
```

```
console.log('numbers[6] = ' + numbers[6]);
```

```
console.log('numbers.length = ' + numbers.length);
```

```
// for 문으로 배열 요소 출력하기 : 직접 추가하시오.
```

```
</script>
```

```
numbers[0] = 0
```

```
numbers[1] = 11111
```

```
numbers[2] = true
```

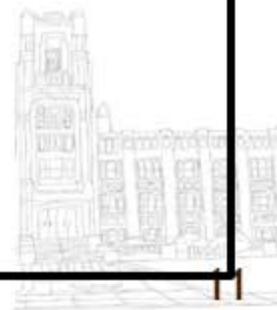
```
numbers[3] = undefined
```

```
numbers[4] = null
```

```
numbers[5] = 11,12,13
```

```
numbers[6] = undefined
```

```
numbers.length = 6
```





배열 요소 추가(C)

- 배열의 요소 추가는 순차적으로 넣을 필요 없다
- 임의의 위치에 동적으로 추가할 수 있다

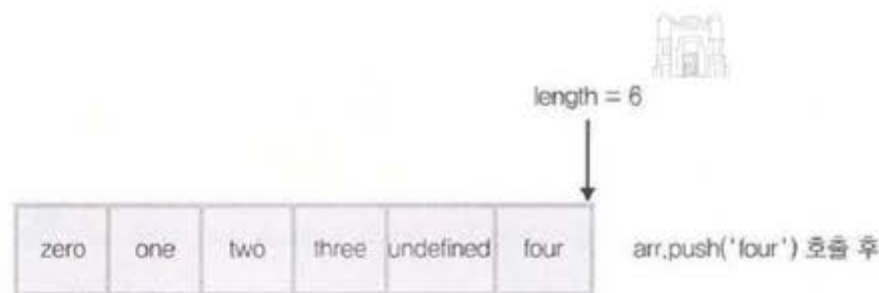
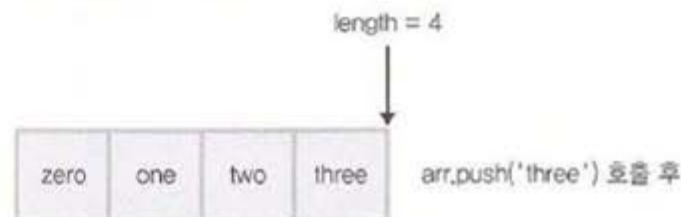
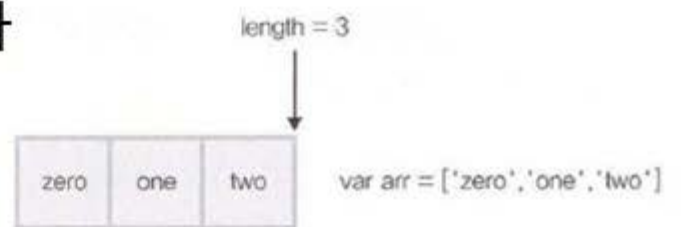
```
// 배열 리터럴을 이용한 배열 생성  
var arr = ['zero', 'one', 'two'];
```

```
// 배열 요소 추가  
arr[3] = 'three';
```

```
// 배열 요소 추가  
arr.length = 5;
```

```
// 배열 요소 추가  
arr[100] = '백';
```

```
// 배열의 마지막에 요소 추가.  
arr[arr.length] = 'four';
```



```
< length = 3  
[true, 100, "eight"]  
length = 72  
[true, 100, "eight", 60: null, 70: undefined, 71: true]  
length = 101  
[true, 100, "eight", 60: null, 70: undefined, 71: true, 100: "ten"]  
>  
-----  
arr[arr.length] = {fname: 'mike', lname: 'jone'};  
console.log('length = ' + arr.length); // (출력값)  
console.log(arr );  
  
// length 값 변경 후. push() 메서드 호출  
arr.length = 100;  
arr[arr.length] = 'ten';  
console.log('length = ' + arr.length); // (출력값)  
console.log(arr );  
  
// for 문으로 배열 요소 출력하기 : 직접 추가하시오.  
</script>
```





배열 요소 수정(U)

- 자바스크립트 배열의 경우는 값을 순차적으로 넣을 필요 없이 아무 인덱스 위치에나 값을 동적으로 추가할 수 있다

- 배열 요소 수정 코드

```
// 배열 리터럴을 이용한 배열 생성  
var arr = [true, 100, 'eight'];
```

```
// 배열 요소 추가  
arr[6] = null;  
arr[10] = undefined;
```

```
// for 문으로 배열 요소 출력
```

// 배열 요소 수정

0번 방을 'update'로 바꾸시오

마지막 방을 'update'로 바꾸시오

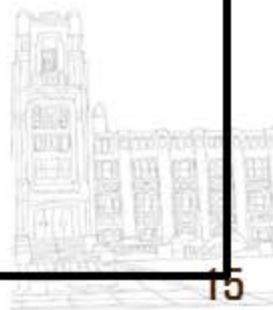


```
<script type="text/javascript">
length = 3 [true, 100, "eight"]
length = 13 [true, 100, "eight", 6: null, 10: undefined, 11: true, 12: "update"]
length = 13 ["update", 100, "eight", 6: null, 10: undefined, 11: true, 12: "update"]

// 배열 요소 추가
arr[6] = null;
arr[10] = undefined;
arr.push(true);
arr[arr.length] = 'update';
console.log('length = ' + arr.length, arr);

// 배열 요소 수정
arr[0] = 'update';
console.log('length = ' + arr.length, arr);

// length 값 변경 후. push() 메서드 호출
arr.length = 100;
arr.push('ten');
console.log('length = ' + arr.length, arr);
</script>
```





배열 요소 삭제(D)

- 배열 요소 삭제
 - `splice()`를 이용한 삭제

- 배열 요소 삭제 예제 코드

```
var arr = ['zero', 'one', 'two', 'three'];  
console.log(arr);  
console.log(arr.length);
```

```
arr.splice(2, 1);  
console.log(arr);  
console.log(arr.length);
```



```
<script type="text/javascript">
  var arr = ['zero', 'one', 'two', 'three'];
  console.log(arr);
  console.log(arr.length);

  arr.splice(2, 1);

  console.log(arr);
  console.log(arr.length);
</script>
```

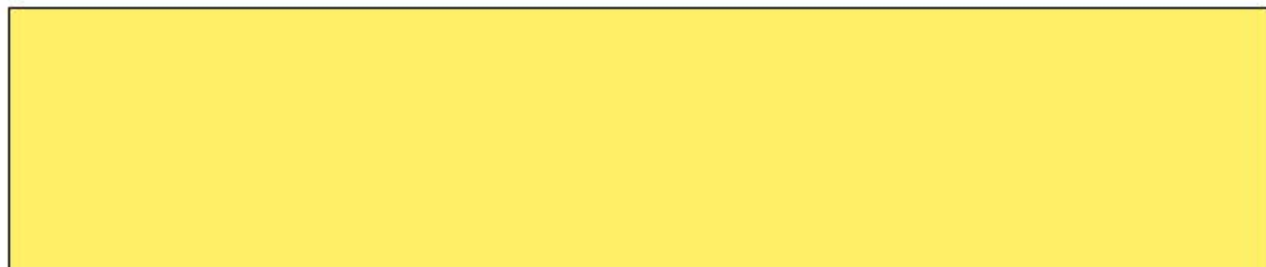
```
arr.length: 4
▼ arr: Array[4]
  0: "zero"
  1: "one"
  2: "two"
  3: "three"
  length: 4
  ► __proto__: Array[0]
```

```
arr.length: 3
▼ arr: Array[3]
  0: "zero"
  1: "one"
  2: "three"
  length: 3
  ► __proto__: Array[0]
```



배열 요소 열거

- for 를 이용한 열거



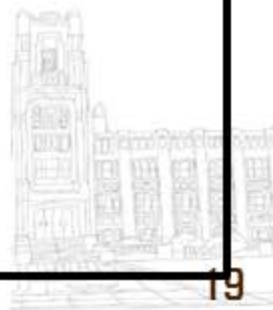
- for in을 이용한 열거




```
<script type="text/javascript">
  var arr = ['zero', 'one', 'two', 'three'];

  console.log('for 문 출력');
  for (var i=0; i<arr.length; i++) {
    console.log (i, arr[i] );
  }

  console.log('for in 문 출력');
  for (var prop in arr) {
    console.log(prop, arr[prop] );
  }
</script>
```





배열 프로퍼티와 메서드

- 배열 프로퍼티
 - length
- 배열 메서드
 - splice() : 배열의 일정 부분을 삭제
 - sort() : 배열의 값을 오름차순으로 정렬
 - reverse() : 배열의 값을 내림차순으로 정렬
 - join() : 특정 문자를 사용해 모든 요소 내용을 문자열로 반환
 - push() : 배열에 추가
 - concat() : 배열 합치기





배열 조작 메서드

- 배열형의 값을 취급하기 위한 객체
- 배열 조작을 위해 프로퍼티와 메서드가 있다.

분류	멤버	개요
가공	concat(ary)	지정 배열을 현재의 배열에 연결
	join(del)	배열 내의 요소를 구분문자 del로 연결
	slice(start [,end])	start~end-1번째의 요소를 빼냄
	splice(start, cnt [,rep [,...]])	배열 내의 start+1~start+cnt+1번째의 요소를 rep, ...로 치환
추가/삭제	pop()	배열 끝의 요소를 취득하여 삭제
	push(data)	배열 끝에 요소를 추가
	shift()	배열 선두의 요소를 취득하여 삭제
	unshift(data1 [,data2,...])	배열 선두에 지정 요소를 추가
정렬	reverse()	역순으로 정렬(반전)
	sort([fnc])	요소를 오름차순으로 정렬
기타	length	배열의 사이즈
	toString()	「요소, 요소, ...」의 형식으로 문자열로 치환

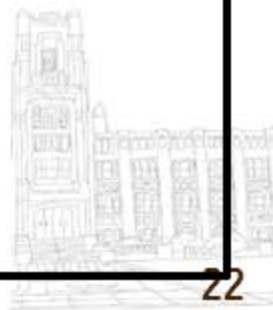


```
<script>
  var arr1 = ['S', 'T', 'O', 'H', 'B'];
  var arr2 = ['Y', 'A', 'M', 'D'];

  // 배열 합치기: concat()
  var arr3 = arr1.concat(arr2);
  console.log( arr3 );

  // 배열의 요소들을 문자열로 만들기: join()
  var str1 = arr1.join('/');
  console.log( str1 );

  var str2 = arr1 + ',' + arr2;
  console.log( str2 );
</script>
```



```
<script>
  // 배열 삭제(추출) : splice()
  var arr4 = arr1.splice(3, 2);
  console.log( arr1 ); // 'S', 'T', 'O'
  console.log( arr4 ); // 'H', 'B'

  // 배열 정렬: sort() . 문자 순으로 정렬
  var arr5 = [30, 3, 2, 156, 34];
  arr5.sort( );
  console.log( arr5 ); // 156, 2, 3, 30, 34
```

arr5를 숫자 크기 순으로 정렬하시오
--> 2,3,30,34,156

```
</script>
```




concat(array2) 배열을 서로 연결

[0]	[1]	[2]	[3]	[4]	[5]
apple	strawberry	orange	banana	melon	peach

toString 문자열로 치환

melon,peach

slice(1) 두 번째 이후의 요소를 추출

[1]	[2]	[3]
strawberry	orange	banana

join('&') &로 연결

melon&peach

array1

[0]	[1]	[2]	[3]
apple	strawberry	orange	banana

length 길이

array2

[0]	[1]
melon	peach

splice(1, 2, 'grape', 'lemon') 2~3번째의 요소를 치환

[0]	[1]	[2]	[3]
apple	grape	lemon	banana

shift 선두의 요소를 삭제

[0]	[1]	[2]
grape	lemon	banana

reverse 역순

[0]	[1]	[2]
banana	lemon	grape

pop 맨 끝의 요소를 삭제

[0]	[1]	[2]
apple	grape	lemon

unshift('melon', 'apple') 선두에 요소를 추가

[0]	[1]	[2]	[3]	[4]
melon	apple	grape	lemon	banana

push('melon') 맨 끝에 요소 추가

[0]	[1]	[2]	[2]
apple	grape	lemon	melon

sort 정렬

[0]	[1]	[2]	[3]	[4]
apple	banana	grape	lemon	melon



Array 객체

- (1) 많은 메소드가 원래의 배열에 영향을 끼친다
 - Array 객체의 메소드는 concat이나 slice, join, toString이란 메소드를 빼고는 조작 대상이 되는 원래의 배열에 영향을 끼친다.
 - 특히 reverse/sort 등의 메소드는 반환값으로도 정렬후의 결과를 되돌리지만, 원래의 배열도 정렬해 버리므로 주의가 필요하다.
- (2) 배열의 내용은 toString 메소드로 확인한다
 - 배열의 내용을 확인하는 데에는 toString 메소드를 사용하면 편리하다. toString 메소드는 배열 내의 요소를 콤마로 연결한 문자열을 반환한다.
 - 참고로, document.writeln 메소드 등 표시 관련 메소드에서는 배열 객체를 직접 건넬으로써 내부적으로 toString 메소드가 호출되기 때문에, 명시적으로 toString 메소드를 호출할 필요가 없다.
- (3) sort 메소드에는 사용자정의 함수도 지정 가능
 - sort 메소드에는 인수로서 사용자정의 함수를 지정하여 디폴트로는 처리할 수 없는 정렬처리를 독자적으로 구현할 수 있다. 단, 사용자정의 함수는 최소한 아래의 두 가지 규칙을 따라야 할 필요가 있다.



	배열
생성	<code>var arr = [] ;</code>
추가. C reate.	<code>arr[arr.length] = 값;</code>
읽기. R ead	<code>arr[방번호]</code>
수정. U pdate	<code>arr[방번호] = 값;</code>
삭제. D elete	<code>arr.splice(방번호)</code>
정렬. S orting	<code>arr.sort()</code>
검색. S earching	X
열거.	<pre>for(var i in arr) { ... }</pre>
프러퍼티	<code>arr.length</code>
답을 수 있는 것은?	다

다섯 명 심사위원의 점수를 입력하여 유효 점수와 평균을 출력하는 프로그램을 작성합니다. 유효점수는 최고점과 최저점을 제외한 점수이며 합계와 평균은 유효점수로 계산합니다. 단, 평균은 소수점 두 자리에서 반올림하시오.

■ 실행결과예시

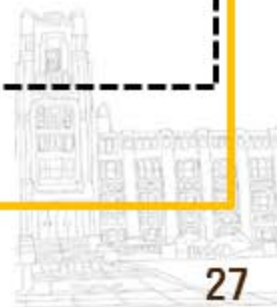
심사위원의 수를 입력하시오 : 5

5명 심사위원의 점수 입력 : 7 9 4 8 5

유효점수 : 5 7 8

합계 : 20

평균 : 6.67





Array 객체

- 배열을 취급하기 위한 객체
- Array 객체를 이용한 배열의 선언 예제

```
var ary = new Array(); // 공백의 배열을 생성
```

```
var ary = new Array('사토', '다카에', '나가타'); // 지정 요소로 배열을 생성
```

```
var ary = new Array(10); // 지정 사이즈로 배열을 생성
```

```
var ary = new Array(-10); // ??? Error
```

- 배열을 생성할 때에는 가능한 한 배열 리터럴을 이용하도록 하자.
 - var ary = [];

