훈련교사 : 전 은 석
email : euns_jun@naver.com

## ❖ 함수의 반환값

◆ 함수에 반환값이 있는 경우 **return 반환값;** 형식으로 값을 반환 해줄 수 있다.

```
function 함수명(매개변수1, 매개변수2, 매개변수3) {    // 함수선언
        실행문장;
        return  반환값;
}
result = 함수명(인자1, 인자2, 인자3);     // 함수 호출
```

```
<script>
    var result;
    function add(name, n) {
        document.write(name + " 학생이 1부터 " + n + "까지 덧셈 수행<br>");
        var sum=0;
        for(var i=1; i<=n; i++) {
            sum+=i;
        }
        return sum;
    }
    result=add('홍길동', 10);
    document.write("결과 : " + result + "<p/>");
    result=add('이영희', 100);
    document.write("결과 : " + result + "<p/>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

## ❖ 함수를 변수에 담기

◆ 함수를 변수에 담을 수도 있다.

```
<script type="text/javascript">
        function abcd() {// 함수 선언
                var a=10;
                var b=5;
                alert(a-b);
        }

        var fun1 = abcd; // 함수 대입
        fun1();                         // 변수 실행
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

- 여러 데이터 값을 저장하는 공간
- 원소: 배열에 저장된 하나 하나의 데이터
- 인덱스: 원소를 구분하는 번호, 0부터 매김


## ◆ 배열 리터럴로 생성하기

```
var 배열명=[원소1, 원소2, 원소3, ... ];
```

```
<script type="text/javascript">
      var city=[];    // 배열 변수 선언
      city[0]="Seoul";
      city[1]="Busan";
      city[2]="Incheon";
      city[3]="Mokpo";
      city[4]="Sejeong";
      function printArr(){
    var i;
    for(i=0; i<city.length; i++) {
        document.write("배열 데이터 ["+ i + "] = " + city[i] + "<br>");
    }
}
printArr();
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

## ◆ 배열 객체로 생성하기

```
var 배열명=new Array(원소1, 원소2, 원소3, ... );
```

```
<script type="text/javascript">
     var city=new Array("Seoul","Busan","Incheon");
 function printArr() {
    var i;
    for(i=0; i<city.length; i++) {
       document.write("배열 데이터 ["+ i + "] = " + city[i] + "<br>");
    }
 }
printArr();
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

## ◆ 배열 객체 생성 확인 방법

| 방법 | 사용 예 | 결과 |
|---|---|---|
| 타입 확인 연산자인 typeof 사용 | typeof city | object |
| 배열 객체의 메소드인 isArray( ) 사용 | Array.isArray(city) | true |
| Array 생성자의 연산자인 instanceof 사용 | city instanceof Array | true |

```
<script type=“text/javascript”>
var city=new Array("Seoul","Busan","Incheon");
function printArr() {
    if(city instanceof Array) {
        document.write("배열 객체가 생성되었습니다.<p/>");
        var i;
        for(i=0; i<city.length; i++) {
            document.write("배열 데이터 ["+ i + "] = " + city[i] + "<br>");
        }
    }
    else {
        document.write("배열 객체가 아닙니다.<br>");
        document.write("데이터 : " + city + "<br>");
    }
}
printArr();
document.write("<p/> city 변수 타입 : " + typeof city + "<br>");
document.write("배열 객체 확인 결과 : " + Array.isArray(city) + "<br>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열 데이터 접근 및 조작

## ◆ 홀수 번째 저장된 데이터만 0으로 초기화하기

```
<script>
    var arrdata=[];
    function insertArr() {
        var i=0;
        for(i=0; i<=99; i++) {
            arrdata[i]=i+1;
            document.write(arrdata[i] + " ");
        }
    }
    function delArr() {
        var i;
        for(i=0; i<arrdata.length; i++) {
            if(i%2==0) {
                arrdata[i]=0;
            }
            continue;
        }
        selectArr();
    }
    function selectArr() {
        var i;
        for(i=0; i<arrdata.length; i++) {
            document.write(arrdata[i] + " ");
        }
        document.write("<p>홀수 번째 데이터 초기화 완료!" + "</p>");
        document.write("<a href='22_arr.html'>돌아가기</a>");
    }
    insertArr();
</script>
```

## ❖ 배열 데이터 접근 및 조작

### ◆ 배열에 저장된 데이터 삭제하기

```
<script>
    var arrdata=[];
    function insertArr() {
        var i=0;
        for(i=0; i<=99 ; i++) {
            arrdata[i]=i+1;                    // 1~100 저장
            document.write(arrdata[i] + " ");    // 데이터 출력
        }
        document.write("<p>배열 크기 : " + arrdata.length + "</p>");
    }
    function delDataArr() {
        var i;
        for(i=0; i<arrdata.length; i++) {
            arrdata[i]=0;                    // 배열 데이터를 0으로 초기화
        }
        selectArr();
    }
    function allDelArr() {
        arrdata.length=0;    // 배열 초기화
        selectArr();
    }
    function selectArr() {
        var i;
        for(i=0; i <arrdata.length; i++) {
            document.write(arrdata[i] + " ");    // 데이터 조회
        }
        document.write("<p> 배열 크기 : " + arrdata.length + "</p>");
        document.write("<a href='23_arr.html'>돌아가기</a>");
    }
    insertArr();    // 배열 데이터 생성 함수 호출
</script>
<p/>
<button type="button" onclick="delDataArr()">배열 데이터 초기화</button>
<button type="button" onclick="allDelArr()">배열 데이터 삭제</button>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

## ◆ join

- 배열에 저장된 모든 원소를 문자열로 변환한 후 연결하여 출력

```
<script type="text/javascript">
   var city=["서울", "부산", "대전"];
   var joindata1=city.join();
   var joindata2=city.join('-');
   var joindata3=city.join(' 그리고 ');
   document.write("조인 결과1 : " + joindata1 + "<p/>");
   document.write("조인 결과2 : " + joindata2 + "<p/>");
   document.write("조인 결과3 : " + joindata3 + "<p/>");
</script>
```

조인 결과1 : 서울,부산,대전
조인 결과2 : 서울-부산-대전
조인 결과3 : 서울 그리고 부산 그리고 대전

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

## ◆ concat

- 지정된 배열에 두 개 이상의 데이터를 결합하거나 다른 배열 객체를 결합

```
<script type="text/javascript">
        var city01=["서울", "부산", "대전"];
        var city02=["대구", "광주", "인천"];
        var city03=["전주", "부여", "세종"];
        var data1=city01.concat("수원", "오산");
        var data2=city01.concat(city02);
        var data3=city01.concat(city03, city02);
        document.write("결과1 : " + data1 + "<p/>");
        document.write("결과2 : " + data2 + "<p/>");
        document.write("결과3 : " + data3 + "<p/>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

## ◆ reverse

- 배열 원소의 순서를 반대로 정렬

```
<script>
    var data=[9, 8, 7, 6, 5, 4, 3, 2, 1];
    document.write("배열 : " + data.join() + "<p/>");
    var rdata=data.reverse();    // 배열 원소를 반대로 정렬
    document.write("결과 : " + rdata + "<p/>");
</script>
```

## ◆ sort

- 배열 원소를 정렬

```
<script>
    var ndata1=[19, 38, 67, 26, 55, 24, 53, 12, 31];
    var ndata2=[132, 2, 41, 123, 45, 1234, 6, 29, 4567];
    var edata=['Apple', 'Html', 'Game', 'Computer', 'Java'];
    var kdata=['서울', '부산', '구포', '대구', '인천'];
    document.write("수치 정렬1 : " + ndata1.sort() + "<p/>");
    document.write("수치 정렬2 : " + ndata2.sort() + "<p/>");
    document.write("수치 정렬3 : " + ndata2.sort(function(a, b) {return a - b;}) + "<p/
>");
    document.write("영문 정렬 : " + edata.sort() + "<p/>");
    document.write("한글 정렬 : " + kdata.sort() + "<p/>");
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

## ◆ slice

- 배열의 특정 범위에 속하는 원소만 선택하여 배열 생성

```
<script>
    var kdata=['서울', '부산', '구포', '대구', '인천', '대전', '세종'];
    var str1=kdata.slice(0, 4);
    var str2=kdata.slice(2, -1);
    var str3=kdata.slice(-4, -2);
    document.write("부분 배열1 : " + str1 + "<p/>");
    document.write("부분 배열2 : " + str2 + "<p/>");
    document.write("부분 배열3 : " + str3 + "<p/>");
</script>
```

## ◆ splice

- 배열의 원소 추가 또는 제거

```
<script>
    var kdata=['서울', '부산', '구포', '대구', '대전'];
    var str1=kdata.splice(1, 2);
    document.write("삭제 데이터 : " + str1 + "<br>");
    document.write("남은 배열 : " + kdata + "<p/>");
    var str2=kdata.splice(1, 1, '강릉', '세종');
    document.write("삭제 데이터 : " + str2 + "<br>");
    document.write("남은 배열 : " + kdata + "<p/>");
    var str3=kdata.splice(2, Number.MAX_VALUE);
    document.write("삭제 데이터 : " + str3 + "<br>");
    document.write("남은 배열 : " + kdata + "<p/>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

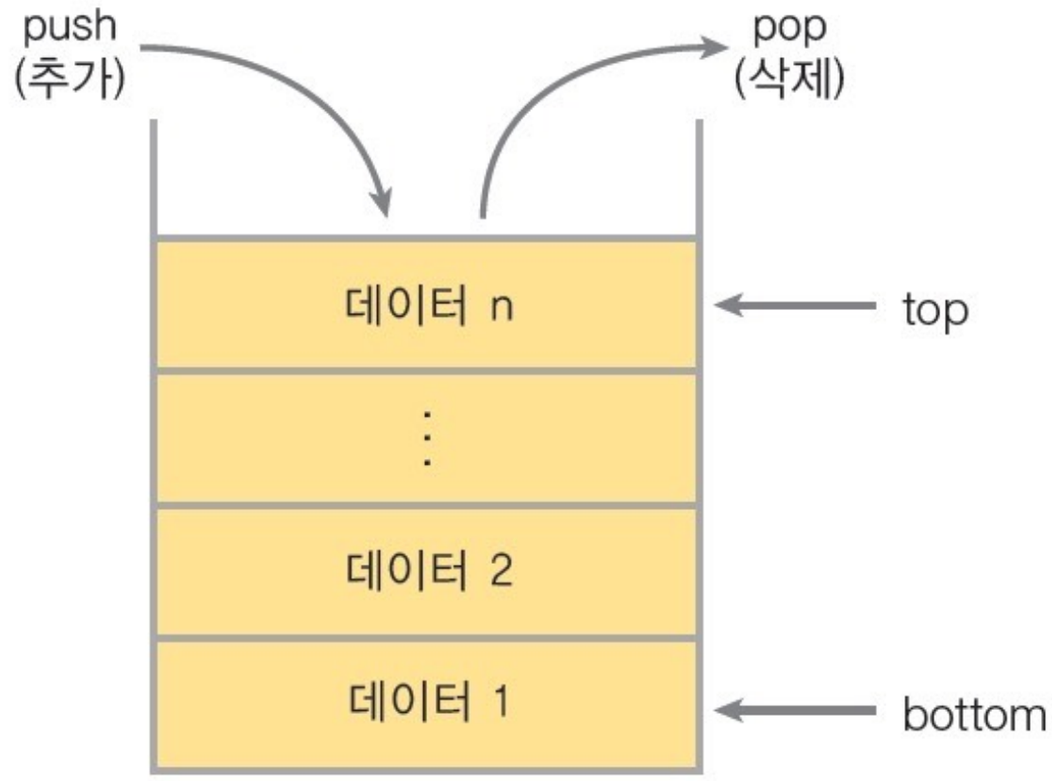
## ◆ pop & push

- push 메소드: 배열의 마지막 위치에 데이터를 추가하고 배열의 길이를 반환
- pop 메소드: 배열의 마지막 위치에 있는 데이터를 삭제하고 삭제한 데이터를 반환

### ∗ 스택

- 모든 데이터의 삽입과 삭제가 배열의 한쪽 끝에서만 수행되는 자료 구조

```
<script>
    var kdata=['서울', '부산', '구포', '대구', '대전'];
    var p1=kdata.push('청주', '세종');
    document.write("데이터 : " + p1 + "<br>");
    document.write("배열 데이터 : " + kdata + "<p/>");
    var p2=kdata.pop();
    document.write("데이터 : " + p2 + "<br>");
    document.write("배열 데이터 : " + kdata + "<p/>");
</script>
```




스택의 구조

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

## ◆ shift & unshift

- shift 메소드: 배열의 맨 처음 위치에 데이터를 삭제하고 배열의 삭제된 데이터 반환
- unshift 메소드: 배열의 맨 처음 위치에 데이터를 삽입하고 배열의 길이 반환

```
<script>
    var kdata=['서울', '부산'];
    var p1=kdata.unshift('청주', '세종');
    document.write("데이터 : " + p1 + "<br>");
    document.write("배열 데이터 : " + kdata + "<p/>");
    var p2=kdata.shift();
    document.write("데이터 : " + p2 + "<br>");
    document.write("배열 데이터 : " + kdata + "<p/>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

## ◆ forEach

- 배열을 반복하며 저장된 데이터를 조회

```
<script>
    var kdata=['서울', '부산', '청주', '대구'];
    function printArr(item, index) {
        document.write("배열 데이터 [" + index + "] : " + item + "<br>");
    }
    kdata.forEach(printArr);
</script>
```

```
<script>
    var data=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
    var sum=0;
    function addArr(value) {
        sum+=value;
    }
    data.forEach(addArr);
    document.write("배열 데이터 합 :" + sum + "<p/>");
</script>
```

# ❖ 배열

## ◆ map

- 배열의 데이터를 함수의 인자로 전달하고 함수의 수행 결과를 반환 받아 새로운 배열 생성

```
<script>
    var data=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
    function mapArr(value) {
        return value*value;
    }
    var mapdata=data.map(mapArr);
    document.write("원래 배열 :" + data + "<p/>");
    document.write("map 메소드 적용 배열 :" + mapdata + "<p/>");
</script>
```

## ◆ filter

- 배열의 데이터 중에 조건이 참인 데이터만 반환하여 새로운 배열 생성

```
<script>
    var data=[21, 42, 33, 14, 25, 12, 37, 28, 16, 11];
    function filterArr(value) {
        return value>=18;
    }
    var fdata=data.filter(filterArr);
    document.write("필터 전 배열 : " + data + "<p/>");
    document.write("필터 후 배열 : " + fdata + "<p/>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 배열

## ◆ indexOf & lastIndexOf

- 배열의 데이터를 검색하여 인덱스 위치를 반환

  - **indexOf 메소드**: 검색 시작 위치를 지정할 수 있음
  - **lastIndexOf 메소드**: 배열의 맨 마지막 원소부터 검색 시작

```
<script>
    var data=[10, 20, 30, 40, 30, 60, 70, 30, 90,100];
    document.write("배열 데이터 : [" + data + "]<p/>");
    document.write("처음부터 검색한 30의 인덱스 : " + data.indexOf(30) + "<p/>");
    document.write("마지막에서 검색한 30의 인덱스 : " + data.lastIndexOf(30) + "<p/
>");
    document.write("세 번째부터 검색한 30의 인덱스 : " + data.indexOf(30, 3) + "<p/
>");
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 연관 배열

## ◆ 연관 배열 생성 방법

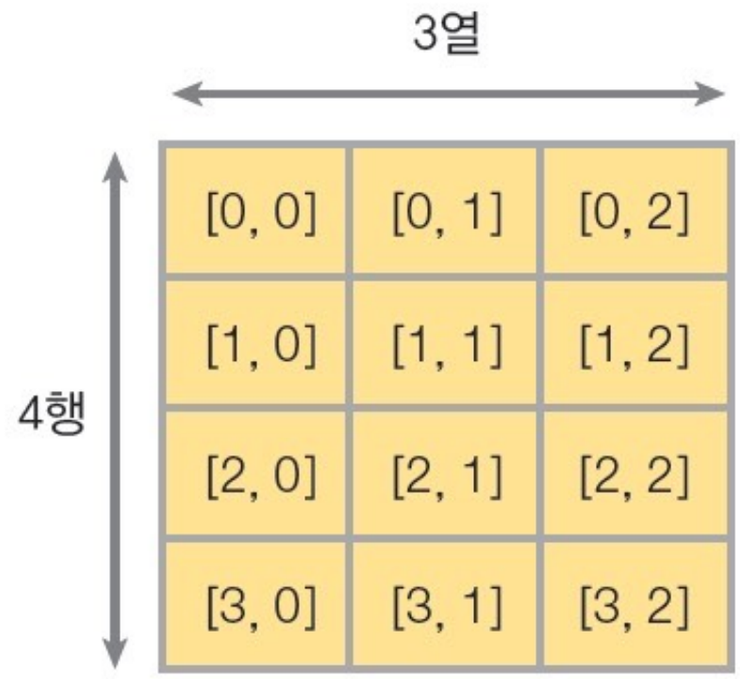
```
arr={key_1:value1, key_2:value2, ..... , key_n:value_n};
```

```
<script>
    var data={'f0':100, 'f1':200, 'f2':300};
    data['f3']=400;    // 배열 데이터 저장
    data.f4=500;       // 배열 데이터 저장
    document.write(data.f0 + "<br>");    // 'f0'키 데이터 조회
    document.write(data.f1 + "<br>");    // 'f1'키 데이터 조회
    document.write(data['f2'] + "<br>");
    document.write(data['f3'] + "<br>");
    document.write(data['f4'] + "<br>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

## ❖ 2차원 배열

### ◆ 2차원배열의 구조




```
<script>
    var d2data=[[10, 20, 30, 40, 0], [60, 70, 80, 90, 0]];
    d2data[0][4]=50;
    d2data[1][4]=100;
    document.write("2차원 배열 첫 번째 데이터 : " + d2data[0][0] + "<br>");
    document.write("2차원 배열 마지막 데이터 : " + d2data[1][4] + "<br>");
    document.write("2차원 배열 행 길이 : " + d2data.length + "<br>");
    document.write("2차원 배열 열 길이 : " + d2data[0].length + "<br>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

# ❖ 2차원 배열

## ◆ 1차원 배열로 2차원 배열 생성하고 조회하기

```
<script>
    var arr0=[10, 20, 30, 40, 50];
    var arr1=[11, 21, 31, 41, 51];
    var arr2=[12, 22, 32, 42, 52];
    var arr3=[13, 23, 33, 43, 53];
    var allArr=[arr0, arr1, arr2, arr3];    // 2차원 배열 생성
    var partArr=[arr1, arr3];               // 2차원 배열 생성
    function printAll() {
        for(var x=0; x<allArr.length; x++) {
            for(var y=0; y<allArr[x].length; y++) {
                document.write(allArr[x][y] + " ");
            }
            document.write("<p/>");
        }
        document.write("<a href='39_arr.html'>돌아가기</a>");
    }
    function printPart() {
        for(var x=0; x<partArr.length; x++) {
            for(var y=0; y<partArr[x].length; y++) {
                document.write(partArr[x][y] + " ");
            }
            document.write("<p/>");
        }
        document.write("<a href='39_arr.html'>돌아가기</a>");
    }
</script>
<button type="button" onclick="printAll()">전체 배열 데이터 보기</button></p>
<button type="button" onclick="printPart()">홀수 배열 데이터 보기</button>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

## ❖ 2차원 배열

### ◆ for 문 이용 2차원 배열 만들기

```
<script>
    var data=[];
    for(var i=0; i<10; ++i) {
        data[i]=[String(i+"-"+0), String(i+"-"+1), String(i+"-"+2)];
    }
    function printData() {
        for(var x=0; x<data.length; x++) {
            for(var y=0; y<data[x].length; y++) {
                document.write(data[x][y] + " ");
            }
            document.write("<p/>");
        }
        document.write("<a href='40_arr.html'>돌아가기</a>");
    }
</script>
<button type="button" onclick="printData()">전체 배열 데이터 보기</button>
```