

5강. 배열(Array)



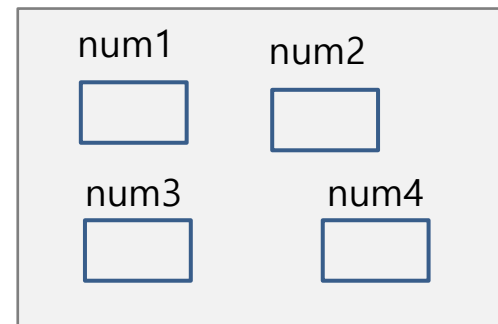
배열(객체)이란?

배열 사용의 필요성

- 정수 10개를 이용한 학생의 성적 프로그램을 만들때
10개의 변수를 선언

`var num1, var num2, var num3... var num10;`
정보가 흩어진 채 저장되고, 변수 이름이 많아
비효율적이고 관리하기 어렵다.

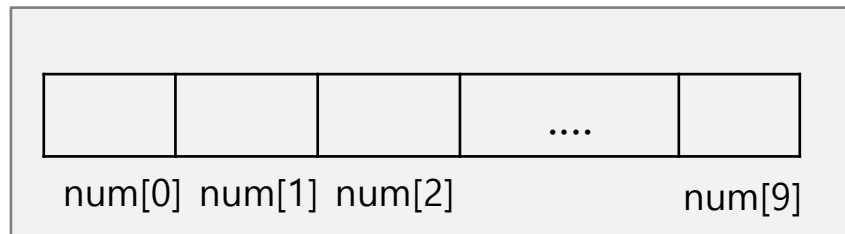
메모리



배열 사용의 장점

- 인덱스를 이용하여 순차(순서)적으로 관리할 수 있다 -> 효율적이다.

메모리



배열(객체)이란?

배열(Array)이란?

- 여러 개의 연속적인 값을 저장하고자 할 때 사용하는 자료형이다.

배열의 생성

방법1 : `var 배열 이름 = [값1, 값2, 값3...]`

방법2 : `var 배열 이름 = new Array(5)`

배열의 생성

방법 1 - 리터럴을 사용한 배열 객체 생성

```
animal[0] = cow  
animal[1] = dog  
animal[2] = mouse  
animal[3] = undefined  
배열의 길이 : 3개  
cow  
dog  
mouse
```

```
//배열 생성 첫번째 방법 - 리터럴 방식  
let animal = []; //비어있는 배열 생성
```

```
//입력(저장)  
animal[0] = 'cow';  
animal[1] = 'dog';  
animal[2] = 'mouse';
```

array1.html

```
//특정 요소 출력  
document.write("animal[0] = " + animal[0] + '<br>');  
document.write("animal[1] = " + animal[1] + '<br>');  
document.write("animal[2] = " + animal[2] + '<br>');  
document.write("animal[3] = " + animal[3] + '<br>');
```

```
//배열의 크기(길이)  
document.write("배열의 길이 : " + animal.length + "개<br>");
```

```
//조회 - 일반 for문  
//주의 - 0번 인덱스부터 시작하고 배열의 길이 미만일 것.  
for(let i = 0; i < animal.length; i++){  
    document.write(animal[i] + '<br>')  
}
```

배열의 조회 - for in 반복문

배열 조회 - for in 반복문으로 출력하기

```
for(var 변수 in 배열이름){ }
```

```
//배열 선언과 동시에 저장
let lang = ['Java', 'Javascript', 'Python'];           array2.html

document.write("배열의 길이 : " + lang.length + "개<br>");

//조회 - 일반 for
for(let i = 0; i < lang.length; i++){
    document.write(lang[i] + " ")
}

document.write('<br>')

//for ~ in 출력
for(let i in lang)
    document.write(lang[i] + " ")
```

배열의 생성2

방법2 - Array 객체를 사용한 배열

```
<script>
var num = new Array(); //num 이름의 배열 생성
console.log("배열의 길이 : " + num.length + "개");
console.log("num : " + num);

//인덱스로 데이터 저장
num[0] = 11;
num[1] = 24;
num[2] = 30;
num[3] = 40;

console.log("배열의 길이 : " + num.length);

//조회 - 리스트 목록
console.log("num : " + num);
```

배열의 길이 : 0개
num :
배열의 길이 : 4개
num : 11,24,30,40
11
24
30
40
num : 11,33,,40
11
33
40

```
newArray.html
//조회2 - 값 출력
for(var i=0; i<num.length; i++){
    console.log(num[i]);
}

//변경, 수정
num[1] = 33;

//삭제
delete num[2];

//조회
console.log("num : " + num);
</script>
```

배열의 생성과 초기화

배열(Array) 생성2 – 생성자로 초기화

```
var arr = new Array(4);
document.write("배열의 길이 : " + arr.length + "개<br>");
document.write("arr : " + arr + '<br>');

//저장
var carts = new Array("생수", "라면", "커피", "계란");
document.write("carts : " + carts + '<br>');

//조회
for(var i = 0; i < carts.length; i++){
    document.write(carts[i] + ' ');
}
document.write('<br>')

//변경
carts[2] = "녹차";

//for ~ in 반복문
for(var i in carts){
    document.write(carts[i] + ' ');
}
```

newArray2.html

배열의 길이 : 4개
arr : ,,,
carts : 생수,라면,커피,계란
생수 라면 커피 계란
생수 라면 녹차 계란

배열의 연산

배열(Array) 연산 – 총점과 평균 구하기

```
let num = new Array(); //num 이름의 인스턴스 생성
```

arrayCalc.html

```
//저장
```

```
num[0] = 11;
```

```
num[1] = 24;
```

```
num[2] = 30;
```

```
num[3] = 40;
```

```
//배열의 연산
```

```
console.log(num[0] + num[1]); //산술
```

```
console.log(num[0] > num[1]); //비교
```

```
console.log(num[0] !== num[1]); //비교
```

```
//합계와 평균
```

```
let sum = 0;
```

```
let avg = 0.0;
```

```
for(let i = 0; i < num.length; i++){
```

```
    sum += num[i]; //누적 합계
```

```
}
```

```
avg = sum / num.length; //평균 = 합계 / 개수
```

```
console.log("sum = " + sum);
```

```
console.log("avg = " + avg);
```


배열의 사용

1부터 10까지의 자연수를 배열에 저장

```
//1부터 10까지의 자연수를 배열에 저장하세요
let number = new Array(10);

//저장
for(var i = 0; i < number.length; i++){
    number[i] = i + 1;
}

//출력
for(var i = 0; i < number.length; i++){
    console.log(number[i]);
}
```

number.html

배열의 사용

배열에 알파벳 저장하기

```
//알파벳 대문자를 저장할 배열을 선언하고 출력하기
let alphabet = new Array(26);
let ch = 'A';
ch = ch.charCodeAt(0);
//console.log(ch);
//let ch2 = String.fromCharCode(ch)
//console.log(ch2);

for(var i = 0; i < alphabet.length; i++){
    alphabet[i] = ch;
    ch++;    //코드(숫자) 1 증가
}

for(var i = 0; i < alphabet.length; i++){
    //코드와 문자로 출력
    console.log(alphabet[i] + ", " + String.fromCharCode(alphabet[i]));
}
```

alphabet.html

실습 문제

1부터 10까지의 자연수 중 짝수를 배열에 저장하고 출력하세요
(파일 : evenNumber.html)

👉 실행 결과

2 4 6 8 10

배열의 활용

배열에서 최대값과 최소값 찾기

```
<script>
    var nums = [17, 92, 33, 25, 56, 81];
    var max = nums[0]; //첫번째 값을 max에 저장
    var i;
    for(i=0; i<nums.length; i++){
        if(max < nums[i]){
            max = nums[i]; //nums[i]를 max에 저장
        }
    }
    document.write("최대값 : " + max);
    document.write("<br>");

    var min = nums[0];
    for(i=0; i<nums.length; i++){
        if(min > nums[i]){
            min = nums[i];
        }
    }
    document.write("최소값 : " + min);
</script>
```

findMax.html

최대값 : 92
최소값 : 17

배열 객체의 함수

push() 함수 - 새로운 요소를 추가(맨 뒤에서 추가됨)

pop() 함수 - 끝 요소 제거하는 기능(맨 뒤에서 삭제)

```
<h3>배열의 요소 추가, 삭제</h3>
```

```
<script>
```

arrayFunction.html

```
var nums = [1, 2, 3];
```

```
//요소 추가 - push(1) : 맨 뒤에 추가됨  
nums.push(4);
```

```
for(var item in nums){  
    document.write(nums[item] + ' ');  
}  
document.write('<br>');
```

```
//요소 삭제 - pop() : 맨 뒤(끝) 요소  
nums.pop();
```

```
for(var i in nums){  
    document.write(nums[i] + ' ');  
}  
document.write('<br>');
```

배열 객체의 함수

unshift() 함수 - 새로운 요소를 추가(맨 앞에서 추가됨)

shift() 함수 - 끝 요소 제거하는 기능(맨 앞에서 삭제)

```
//맨 앞에 추가 - unshift(1)
nums.unshift(0);

for(var item in nums){
    document.write(nums[item] + ' ');
}
document.write('<br>');

//맨 앞 삭제 - shift()
nums.shift();

for(var item in nums){
    document.write(nums[item] + ' ');
}
```

[arrayFunction.html](#)

배열 객체의 함수

splice() 함수

- ▶ 추가 기능 – 특정 인덱스에 요소를 추가할 수 있다.
- ▶ 삭제 기능 – 특정 인덱스의 요소를 1개 이상 삭제할 수 있다.

```
<script>
  var animal = ['bear', 'chicken', 'dog', 'horse'];

  //특정 인덱스에서 제거(삭제)
  animal.splice(2, 2); //2번 인덱스에서 2개 삭제

  document.write("animal : " + animal + '<br>');

  //특정 인덱스에서 요소 추가
  animal.splice(1, 0, 'cow'); //1번 인덱스에 삭제하지 않고 'cow' 추가

  document.write("animal : " + animal + '<br>');

  for(var i in animal){
    |   document.write(animal[i] + ' ');
  }
</script>
```

splice.html

배열 요소를 특정한 위치에서 추가, 삭제

animal : bear,chicken
animal : bear,cow,chicken
bear cow chicken

배열 객체의 함수

둘 이상의 배열을 합치는 `concat()` 함수

배열 요소를 연결하는 `join()` 함수

concat.html

```
<script>
  var numbers = [1, 2, 3];
  var chars = ['a', 'b', 'c'];

  //두 배열 합치기 - concat()
  var list = numbers.concat(chars); //배열의 순서 주의!!
  document.write(list + '<br>');

  //구분기호 연결하기 - join() -> 매개변수로 구분기호 전달
  document.write(numbers.join() + '<br>'); //기본 콤마로 구분 1,2,3
  document.write(numbers.join("-") + '<br>'); //1-2-3
  document.write(chars.join("-") + '<br>'); //a-b-c

  document.write(list.join(":"));
</script>
```

배열의 요소를 합치기, 구분 기호 연결하기

1,2,3,a,b,c
1,2,3
1-2-3
a-b-c
1:2:3:a:b:c

배열의 객체 함수 연습문제

1부터 10까지의 자연수 중 홀수를 사용하여 합을 구하고, 그 합을 배열의 맨 뒤에 추가하여 출력하는 프로그램을 작성하세요(파일 : sum-push.html)

👉 실행 결과

1 3 5 7 9 25

내장 객체 - Array

indexOf() 함수 - 특정한 문자열의 첫번째 위치를 반환

대, 소문자 구분하며, 일치하지 않으면 -1 반환

lastIndexOf() 함수 - 특정한 문자열의 뒤에서 첫번째 위치를 반환

```
let str = "Hello world, welcome world";                                indexof.html

//indexOf() - 문자열의 첫번째 위치 반환, 못 찾으면 -1 반환
let n1 = str.indexOf("world"); //6
let n2 = str.indexOf("asia");  //-1

//lastIndexOf() - 문자열의 뒤에서 찾아서 위치 반환
let n3 = str.lastIndexOf("world"); //21

document.write(n1 + "<br>");
document.write(n2 + "<br>");
document.write(n3 + "<br>");
```

다차원 배열

■ 2차원 이상의 배열

1. 지도, 게임 등 평면이나 공간을 구현할 때 많이 사용됨.
2. 이차원 배열의 선언과 초기화

```
var arr = [  
    [1, 2, 3],  
    [4, 5, 6]  
]
```

arr[0][0]	arr[0][1]	arr[0][2]
arr[1][0]	arr[1][1]	arr[1][2]

3. 배열의 개수

배열의 개수(행) : **arr.length** -> 2개

배열의 개수(1열) : **arr[0].length** -> 3개

다차원 배열

❖ 2차원 배열의 생성과 조회

```
let numbers = [  
  [1, 2, 3],  
  [4, 5, 6]  
]
```

//조회 - 인덱싱

```
document.write(numbers[0][0] + '<br>');  
document.write(numbers[0][1] + '<br>');  
document.write(numbers[0][2] + '<br>');  
document.write(numbers[1][0] + '<br>');  
document.write(numbers[1][1] + '<br>');  
document.write(numbers[1][2] + '<br>');
```

array_2d.html

//배열의 개수

```
document.write("배열의 개수(행) : " + numbers.length + '<br>');  
document.write("배열의 개수(1열) : " + numbers[0].length + '<br>');  
document.write("배열의 개수(2열) : " + numbers[1].length + '<br>');
```

//전체 출력

```
for(let i = 0; i < numbers.length; i++){  
  for(let j = 0; j < numbers[i].length; j++){  
    document.write(numbers[i][j] + ' ');  
  }  
}
```

다차원 배열

❖ 2차원 배열의 총점과 평균 구하기

```
let numbers = [ [1, 2, 3], [4, 5, 6, 7, 8] ]  
let sum_v = 0;      //합계  
let count = 0;      //개수  
let avg = 0.0;      //평균
```

array-2d-calc.html

//합계 계산

```
for(let i = 0; i < numbers.length; i++){  
  for(let j = 0; j < numbers[i].length; j++){  
    sum_v += numbers[i][j];  
    count++; //개수 1 증가  
  }  
}
```

//평균 계산

```
avg = sum_v / count;
```

//출력

```
document.write("합계 : " + sum_v + '<br>');  
document.write("개수 : " + count + '<br>');  
document.write("평균 : " + avg.toFixed(2) + '<br>');
```

다차원 배열

❖ 시험 성적 관리 - 학생 5명의 국어와 수학점수 계산

```
//배열 및 변수 선언 및 초기화(저장)
```

```
var score = [  
  [80, 70],  
  [90, 60],  
  [70, 70],  
  [100, 90],  
  [83, 35]  
]  
  
var kor_sum = 0;  
var math_sum = 0;  
var kor_avg, math_avg;
```

```
// 과목별 점수 합계
```

```
for(var i = 0; i < score.length; i++){  
  kor_sum += score[i][0];  
  math_sum += score[i][1];  
}
```

```
// 과목별 점수 평균
```

```
kor_avg = kor_sum / score.length;  
math_avg = math_sum / score.length;  
console.log(kor_avg);
```

```
document.write("국어 점수 합계 : " + kor_sum + '<br>');  
document.write("수학 점수 합계 : " + math_sum + '<br>');  
document.write("국어 점수 평균 : " + kor_avg + '<br>');  
document.write("수학 점수 평균 : " + math_avg + '<br>');
```

calcScore.html