

5강. 배열(Array)



배열(객체)이란?

배열(Array)이란?

- 여러 개(다량)의 데이터를 저장하여 관리할 수 있는 자료 구조이다.
- 인덱스를 이용하여 순차(순서)적으로 관리할 수 있다 -> 효율적이다.

var num1

var num2

var num3

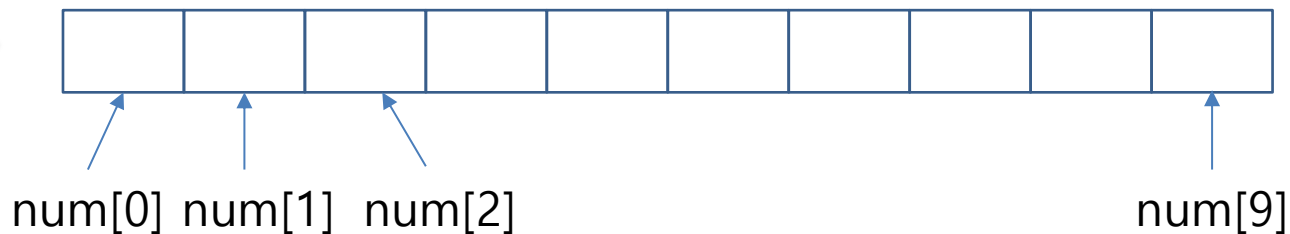
.....

var num9

var num10



var num = new Array(10)



배열의 생성

배열(Array)의 생성

방법 1 - 리터럴을 사용한 배열 객체 생성

```
<script>
    var animal = []; //빈 인덱스
    console.log("배열의 길이 : " + animal.length + "개");
    //저장
    animal[0] = 'cow';
    animal[1] = 'dog';
    animal[2] = 'mouse';

    //전체 조회
    for(var i=0; i<animal.length; i++){
        console.log(animal[i]);
    }

    //생성과 함께 초기화
    var comLang = ['Java', 'Html + Css', 'Javascript', 'Jsp'];
    console.log("Computer Language : " + comLang);
</script>
```

배열의 길이 : 0개

cow

dog

mouse

Computer Language : Java,Html + Css,Javascript,Jsp

배열의 조회 - for in 반복문

for in 반복문

for(var 변수 in 배열이름){ }

```
<script>
  var array = ['딸기', '귤', '사과', '바나나'];

  for(var i=0; i<array.length; i++){
    console.log(array[i]);
  }

  //for in 반복문
  for(var item in array){
    console.log(array[item]);
  }
</script>
```

딸기
귤
사과
바나나
딸기
귤
사과
바나나

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>
```

최대값 : 92
최소값 : 17

배열의 생성2

배열(Array) 생성

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 + "개");

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

```
//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 연산

배열(Array) 연산

```
<script>
    var num = new Array(); //num 이름의 인스턴스 생성
    //인덱스로 데이터 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 - 같지않다

    //합계와 평균
    var sum = 0;
    var avg = 0.0;

    for(var i=0; i<num.length; i++){
        sum += num[i]; //누적 합계
    }
    avg = sum / num.length;
    console.log("sum = " + sum);
    console.log("avg = " + avg);
</script>
```

35

false

true

sum = 105

avg = 26.25

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] + ' ');
}
```

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

실습 문제

좋은 글귀를 배열에 저장해 두고 무작위로 출력하는 프로그램

성공하는 사람은 실패하는데 익숙한 사람이다.

```
// 좋은 글귀
var words = []

words[0] = "당신은 지금도 최고고, 이전에도 최고였으며 앞으로도 최고일 것이다";
words[1] = "성공하는 사람은 실패하는데 익숙한 사람이다.";
words[2] = "후회를 최대한 이용하라. 깊이 후회한다는 것은 새로운 삶을 산다는 것이다.";
words[3] = "가짜 친구는 소문을 믿고, 진짜 친구는 나를 믿는다.";
words[4] = "성공이라는 뜻을 박으려면 끈질김이라는 망치가 필요하다.";
```

```
document.write(words[rand]);
```

실습 문제

words.html

```
<link href="https://fonts.googleapis.com/css2?family=Nanum+Pen+Script&display=swap"
      rel="stylesheet">
<style>
  body{
    margin-top: 50px;
    font-size: 2.5em;
    text-align: center;
    font-family: 'Nanum Pen Script', 바탕;
    background: ■rgb(1, 1, 151);
    color: □white;
  }
</style>
</head>
<body>
  <script src="../js/word.js"></script>
</body>
```

배열 객체의 함수

push() 함수 - 새로운 요소를 추가하는 기능(단, 뒤에서 추가됨)

pop() 함수 - 끝 요소 제거하는 기능

```
<script>
  var nums = [1, 2, 3];

  //요소 추가 - push() 함수
  nums.push(4, 5, 6);

  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] + '&nbsp;');
  }
</script>
```

배열의 요소 추가, 삭제

1 2 3 4 5 6

1 2 3 4 5

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>
```

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

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

```
<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>
```

배열 객체의 함수

indexOf() 함수 – 특정한 문자열의 첫번째 위치를 반환

대, 소문자 구분하며, 일치하지 않으면 -1 반환

lastIndexOf() 함수 – 특정한 문자열의 뒤에서 첫번째 위치를 반환

```
<h3>문자열의 위치 찾기</h3>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
  function myFunction(){
    var str = "Hello world, welcome to the world";
    var n1 = str.indexOf("world");
    //앞에서 찾아서 인덱스 반환
    //대소문자 구분하고 없으면 -1을 반환함

    var n2 = str.lastIndexOf("world");
    //뒤에서 부터 찾아서 인덱스를 반환함
    document.getElementById("demo").innerHTML = n1 + '&nbsp;' + n2;
  }
</script>
```

문자열의 위치 찾기

Try it

6 28

배열 객체의 함수

indexOf() 함수 예제

127.0.0.1:5500 내용:

입력해주세요

127.0.0.1:5500 내용:

안녕하세요

단어를 검색하면 대답하는 프로그램

- '안녕'이 있으면 '안녕하세요' 출력
- 시간을 검색하면 '현재 시간' 출력
- '잘있어' 또는 '잘가'를 검색하면
"안녕히 가세요"를 출력하고 프로그램
이 종료됨
- 찾는 단어가 없으면 "모르는 단어입니다" 출력

배열 객체의 함수

indexOf() 함수 예제

```
<script>
  while(true){
    var input = prompt("입력해 주세요.");
    if(input.indexOf("안녕") >= 0){
      alert("안녕하세요");
    }else if(input.indexOf('시') >= 0){
      var now = new Date();
      var hour = now.getHours();
      var minute = now.getMinutes();
      alert(hour + '시' + minute + '분 입니다.');
```

```
    }else if(input.indexOf('잘가') >= 0 || input.indexOf('잘있어') >= 0){
      alert("안녕히 가세요!");
    }else{
      alert("모르는 단어입니다.");
    }
  }
</script>
```


배열의 연산 연습문제

1부터 10까지의 자연수 중 홀수를 사용하여 합을 구하고, 배열의 맨 뒤에 추가하는 프로그램을 작성하세요

👉 출력 결과

1 3 5 7 9 25

```
<script>
  var numbers = [1, 3, 5, 7, 9];
  var sum = 0;

  //홀수들의 합 구하기
  

  numbers.push(sum); //sum 추가 - 맨 뒤에 추가

  // numbers 출력하기
  
</script>
```

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 배열 예제

```
//이차원 배열 - 2행 3열
var numbers = [
  [1, 2, 3],
  [10, 20, 30]
]
```

```
//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>');
```

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

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

다차원 배열

2차원 배열 연산

```
//이차원 배열 - 2행 3열
var numbers = [
  [1, 2, 3, 4],
  [10, 20, 30]
]

var sum = 0;      //합
var count = 0;    //개수
var avg;          //평균
var i, j;
```

```
//합계 계산
for(i = 0; i < numbers.length; i++){
  for(j = 0; j < numbers[i].length; j++){
    sum += numbers[i][j];
    count++;
  }
}

//평균 계산
avg = sum / count;

//출력
document.write("합계 : " + sum + '<br>');
document.write("개수 : " + count + '<br>');
document.write("평균 : " + avg + '<br>');
```

다차원 배열

시험 성적 관리 - 학생 5명의 국어와 수학점수 계산

```
//학생 5명의 국어, 수학 점수
```

```
var score = [
```

```
    [80, 70],
```

```
    [90, 60],
```

```
    [70, 70],
```

```
    [100, 90],
```

```
    [80, 20]
```

```
]
```

```
var i, j;
```

```
var kor_sum = 0;
```

```
var math_sum = 0;
```

```
var kor_avg, math_avg;
```

```
// 점수 합계
```

```
for(var i = 0; i < score.length; i++){
```

```
    kor_sum += score[i][0];
```

```
    math_sum += score[i][1];
```

```
}
```

```
// 점수 평균
```

```
kor_avg = kor_sum / score.length;
```

```
math_avg = math_sum / score.length;
```

```
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>');
```