

2-1 jQuery 주요 메서드

2-1-1 each() 메서드

- 배열관련 메서드 입니다.
 - 행 또는 열로 이루어진 데이터의 배열을 말합니다.
 - 1차원 배열 행 - 예] sungjuk[50]
 - 2차원 배열 행, 열 - 예] sungjuk[50][20]
 - 3차원 배열 면, 행, 열 - 예] sungjuk[10][50][20]
- 배열(Array())과, 객체(Object()), 제어문(반복문, 조건문)에 대해서 선수 교육이 필요합니다.

2-1-1-1 Array(); 배열객체

2-1-1-1-1 배열선언 방법 1

```
var cars = new Array();
```

```
cars[0] ="Saab";
```

```
cars[1] ="Volvo";
```

```
cars[2] ="BMW";
```

실습 예제]

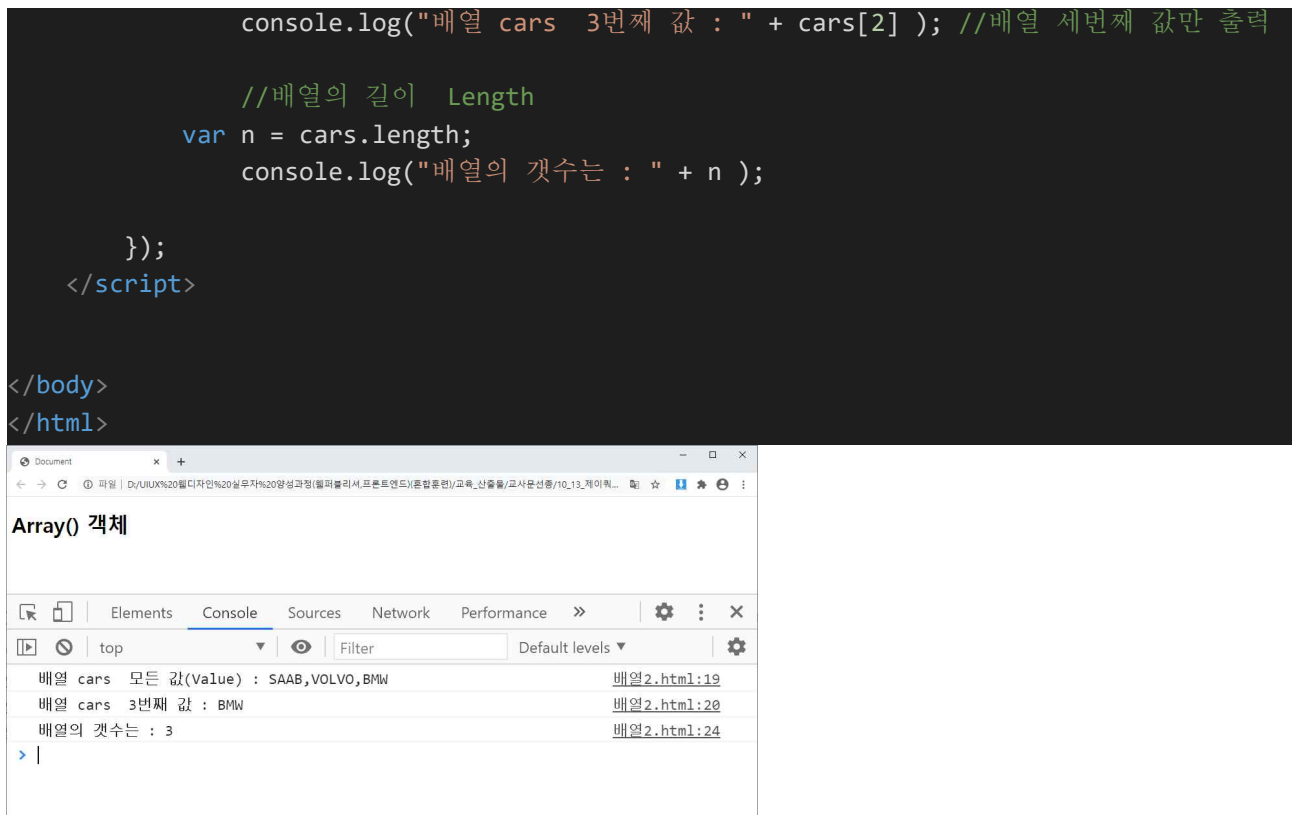
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="./js/lib/jquery-1.12.4.min.js"></script>
</head>
<body>

  <h1>Array() 객체</h1>

  <script>
    $(function(){
      //1-1 배열 생성자(new) 선언방법
      var cars = new Array();
      cars[0]="SAAB";
      cars[1]="VOLVO";
      cars[2]="BMW";

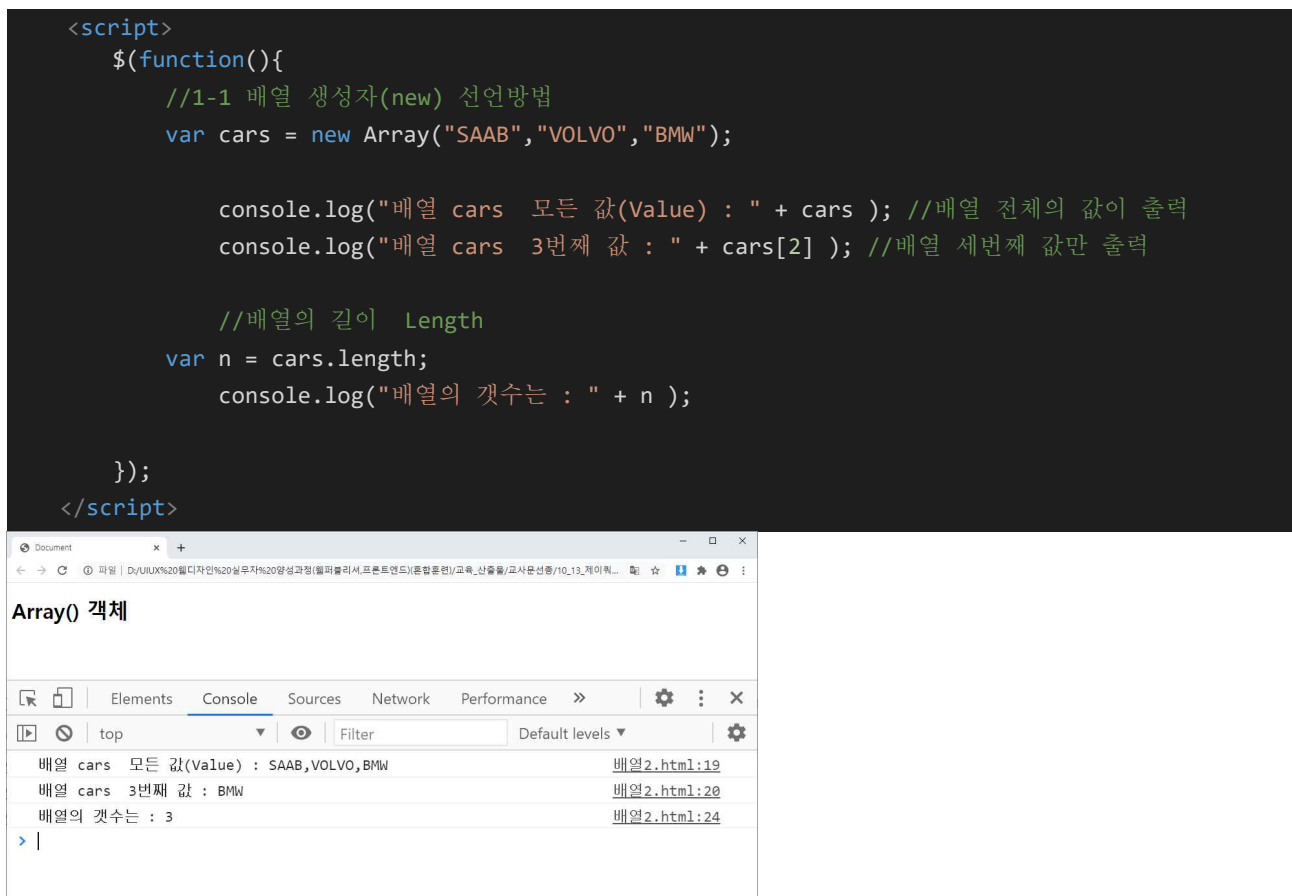
      console.log("배열 cars 모든 값(Value) : " + cars ); //배열 전체의 값이 출력
```

력



2-1-1-1-2 배열선언 방법 2

var cars = new Array("Saab", "Volvo", "BMW");



2-1-1-1-3 배열선언 방법 3 - 배열선언 권장 방법

```
var cars = [];  
cars[0] = "Saab";  
cars[1] = "Volvo";  
cars[2] = "BMW";
```

```
<script>  
  $(function(){  
    //1-1 배열 생성자(new) 제외 선언방법 - 권장방법  
    var cars = [];  
    cars[0]="SAAB";  
    cars[1]="VOLVO";  
    cars[2]="BMW";  
  
    console.log("배열 cars 모든 값(Value) : " + cars ); //배열 전체의 값이 출력  
    console.log("배열 cars 3번째 값 : " + cars[2] ); //배열 세번째 값만 출력  
  
    //배열의 길이 Length  
    var n = cars.length;  
    console.log("배열의 갯수는 : " + n );  
  
  });  
</script>
```

Object(객체) 메서드

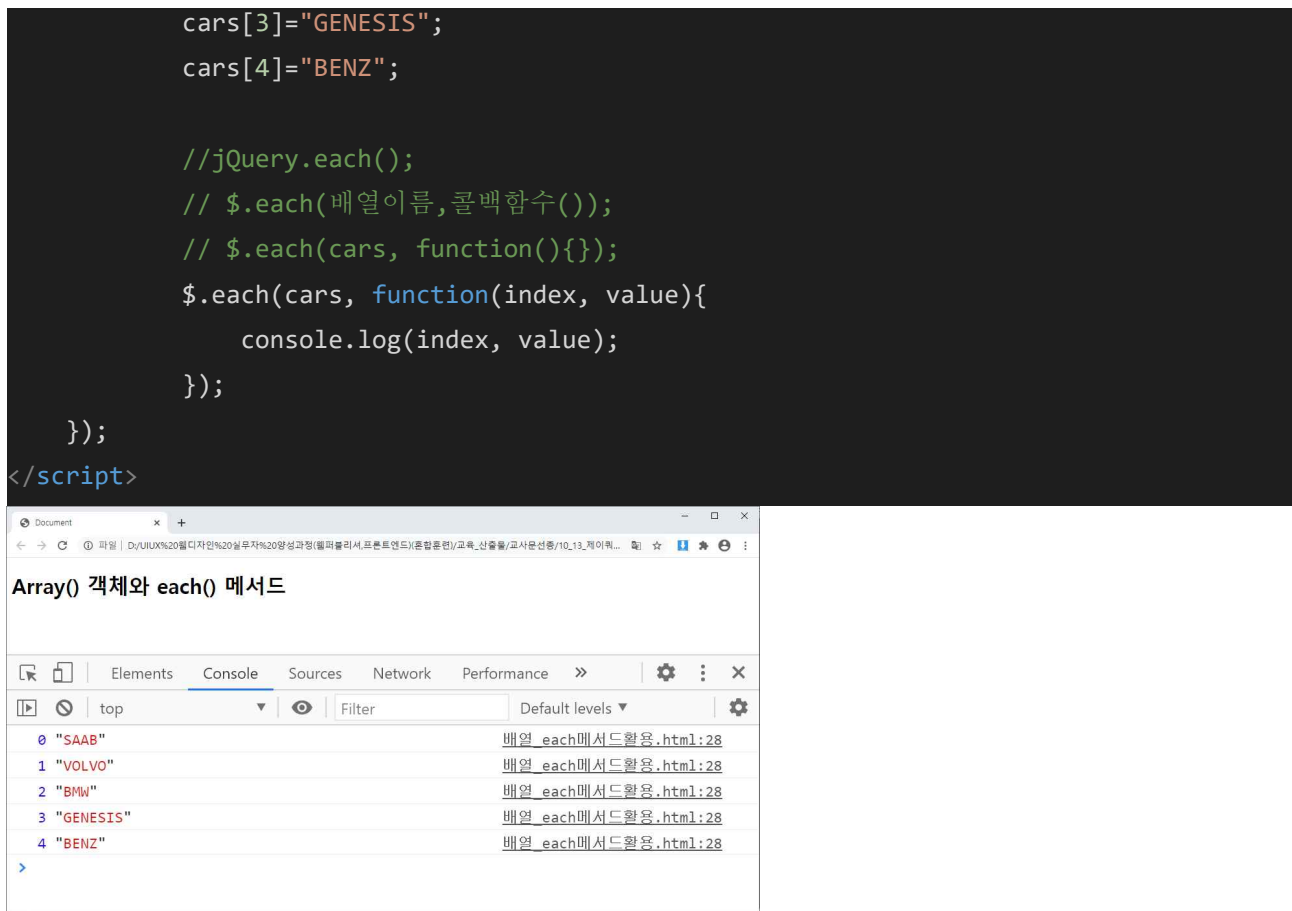
Elements	Console	Sources	Network	Performance	»	⚙	⋮	×
top	Filter	Default levels	⚙					
GENESIS	객체1.html:24							
▶ Object	객체1.html:25							
irum GENESIS	객체1.html:29							
year 2020	객체1.html:29							
weigh 3000	객체1.html:29							
price 6000	객체1.html:29							
>								

2-1-1-1-3-1 배열선언 방법

```
var cars = ["Saab", "Volvo", "BMW"];
```

2-1-1-4 Array(); 배열객체와 each() 메서드

```
<script>  
  $(function(){  
    //1-1 배열 생성자(new) 제외 선언방법 - 권장방법  
    var cars = [];  
    cars[0]="SAAB";  
    cars[1]="VOLVO";  
    cars[2]="BMW";
```



2-1-1-1-5 Array(배열) 길이 속성 length

배열 의 속성은 배열의 길이 (배열 요소의 수)를 반환합니다.

```
var n = cars.length;
```

```
console.log(n);
```

결과(Result) : 3

배열 인덱스(Index) : 0 ~ 2 최대 사용 인덱스번호 (n-1)

예]

```
console.log(cars[0]);
```

```
console.log(cars[1]);
```

```
console.log(cars[2]);
```

2-1-1-3 Object() 객체

2-1-1-3-1 객체선언 방법 1 : 생성자 방식 - 권장하지 않음

```
var person = new Object();  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```

```
<script>  
  $(function(){  
    //1-1 Object(객체) 생성자(new) 선언방법  
    var cars = new Object(); //선언문  
  
    cars.irum="GENESIS"; //객체(Object).속성(key)=값(Value)데이터 입력  
    cars.year=2020; //데이터 입력  
    cars.weigth=3000; //데이터 입력  
    cars.price=6000; //데이터 입력  
  
    console.log(cars.irum);  
    console.log(cars);  
  
    //each()메서드  
    $.each(cars, function(key,value){  
      console.log(key, value);  
    });  
  });  
</script>
```

2-1-1-3-2 객체 방법 2 - 권장 방법 : 리터널 방식

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

또는

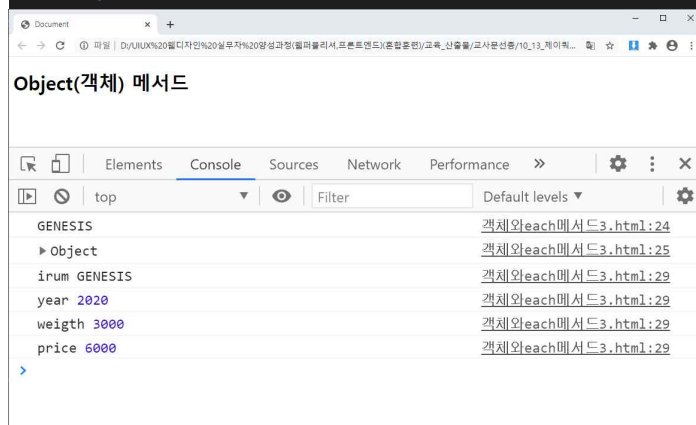
```
var person = {  
    firstName:"John",  
    lastName:"Doe",  
    age:50,  
    eyeColor:"blue"  
};
```

```
<script>  
    $(function(){  
        //1-1 Object(객체) 리터널 선언방법 - 권장 방법  
        var cars = {  
            irum:"GENESIS",  
            year:2020,  
            weigth:3000,  
            price:6000  
        };  
        console.log("객체.이름 = " + cars.irum);  
        console.log("모든 객체.키:값 = " + cars);  
  
        //each()메서드  
        $.each(cars, function(a,b){ //key, value  
            console.log(a, b);  
        });  
    });  
</script>
```

2-1-1-3-3 객체 선언 방법 3

```
var person = {};  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```

```
<script>  
  $(function(){  
    // Object(객체) 리터럴 방식2 선언방법  
    var cars = {};  
  
    cars.irm="GENESIS"; //객체(Object).속성(key)=값(Value)데이터 입력  
    cars.year=2020; //데이터 입력  
    cars.weigth=3000; //데이터 입력  
    cars.price=6000; //데이터 입력  
  
    console.log(cars.irm);  
    console.log(cars);  
  
    //each()메서드  
    $.each(cars, function(key,value){  
      console.log(key, value);  
    });  
  });  
</script>
```



2-1-1-3-4 객체 선언 방법 4 : 리터럴방식과 콜백함수

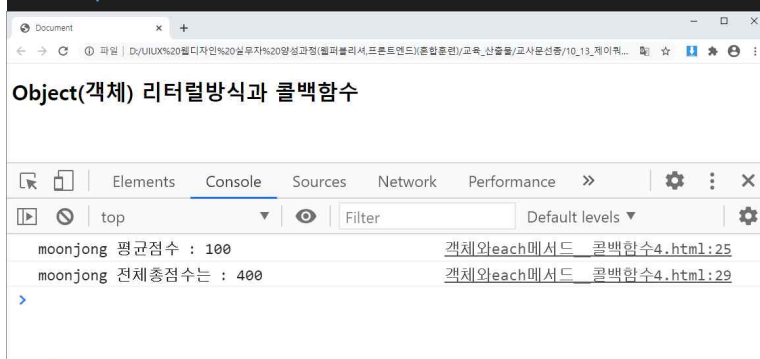
```
<script>  
  $(function(){  
    // Object(객체) 리터럴 방식2 선언방법  
    var sunbjuk = {  
      irm:"moonjong",  
      kor:100,  
      mat:100,  
      eng:100,  
      edps:100,
```

```

    avg:function(){
        var x = (sunbjuk.kor+sunbjuk.mat+sunbjuk.eng+sunbjuk.edps)/4;
        return console.log(sunbjuk.irim + " 평균점수 : " + x);
    },
    tot:function(){
        var x = (sunbjuk.kor+sunbjuk.mat+sunbjuk.eng+sunbjuk.edps);
        return console.log(sunbjuk.irim + " 전체총점수는 : " + x);
    }
};
sunbjuk.avg();
sunbjuk.tot();

});
</script>

```



2-1-1-3-4 객체 선언 방법 5 : Object(객체) 리터럴 방식과 콜백함수, this 객체

```

<script>
    $(function(){
        // Object(객체) 리터럴 방식과 콜백함수, this 객체
        var sunbjuk = {
            irum:"moonjong",
            kor:100,
            mat:100,
            eng:100,
            edps:100,
            avg:function(){
                var x = (this.kor+this.mat+this.eng+this.edps)/4;
                return console.log(this.irim + " 평균점수 : " + x);
            },
            tot:function(){
                var x = (this.kor+this.mat+this.eng+this.edps);
                return console.log(this.irim + " 전체총점수는 : " + x);
            }
        };
        sunbjuk.avg();
        sunbjuk.tot();
    });
</script>

```