

웹 앱 개발을 위한 JavaScript 기초 강의 노트

제 13회차

기본 내장 객체 - ECMAScript 5의 객체

■ 학습목표

- ECMAScript 5에서 추가된 Array 객체의 메서드를 이해하고 사용할 수 있다.
- ECMAScript 5에서 추가된 JSON 객체의 메서드를 이해하고 사용할 수 있다.
- ECMAScript 5에서 추가된 String 객체의 메서드를 이해하고 사용할 수 있다.
- ECMAScript 5에서 추가된 Object 객체의 메서드를 이해하고 사용할 수 있다.

■ 학습내용

- ECMAScript 5 Array 객체
- ECMAScript 5 JSON 객체
- ECMAScript 5 String 객체
- ECMAScript 5 Object 객체

1. ECMAScript 5 Array 객체

1) 개요와 추가 메서드

- 개요

- ECMAScript 5
 - JavaScript 표준안
 - 2009년 12월 객체 관련 부분을 보완한 ECMAScript 5 발표
 - Internet Explorer 8 이하에서는 지원하지 않음

- 추가 메서드

- ECMAScript 5에서 추가된 Array 객체 메서드

메서드	설명
Array.isArray()	배열 여부를 확인하여 불린 값을 리턴
indexOf()	특정 배열 요소를 앞에서부터 검색하여 해당 배열 요소가 위치하는 인덱스 리턴
lastIndexOf()	특정 배열 요소를 뒤에서부터 검색하여 해당 배열 요소가 위치하는 인덱스 리턴
forEach()	배열의 각각의 요소에 대해 지정된 작업을 실행
map()	배열의 각 요소를 변경해 새로운 배열을 생성
filter()	특정 조건을 만족하는 배열 요소를 추출해 새로운 배열 생성
every()	배열 요소가 특정 조건에 모두 만족하는지 확인
some()	배열 요소가 특정 조건에 적어도 하나 만족하는지 확인
reduce()	배열 요소가 하나가 될 때까지 왼쪽부터 배열 요소를 두 개씩 묶는 함수 실행
reduceRight()	배열 요소가 하나가 될 때까지 오른쪽부터 배열 요소를 두 개씩 묶는 함수 실행

1. Array 객체

2) 배열 확인과 배열 요소 위치 확인

- isArray()

- 배열 여부 확인하는 메서드로써, 입력된 매개변수가 배열이면 true, 배열이 아니면 false 리턴

- indexOf(), lastIndexOf()

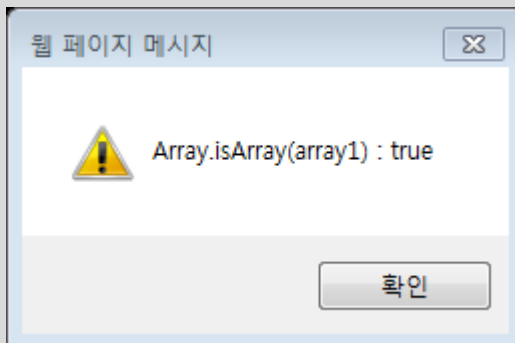
- 매개 변수로 입력된 배열 요소를 앞/뒤에서부터 검색하여 해당 배열 요소의 인덱스를 리턴
- 배열 요소가 존재하지 않으면 -1 리턴

- 예

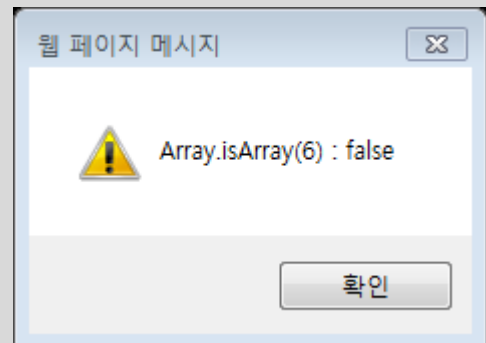
```
<script>
  var array1 = [1, 2, 3, 4, 5];

  ① alert('Array.isArray(array1) : ' + Array.isArray(array1));
  ② alert('Array.isArray(6) : ' + Array.isArray(6));
    alert('array1.indexOf(4) : ' + array1.indexOf(4));
    alert('array1.lastIndexOf(2) : ' + array1.lastIndexOf(2));
    alert('array1.lastIndexOf(6) : ' + array1.lastIndexOf(6));
</script>
```

① 출력 화면



② 출력 화면



1. Array 객체

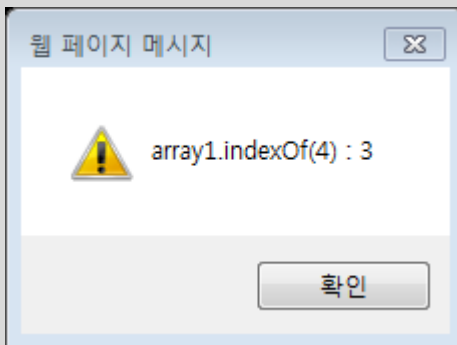
2) 배열 확인과 배열 요소 위치 확인

- 예

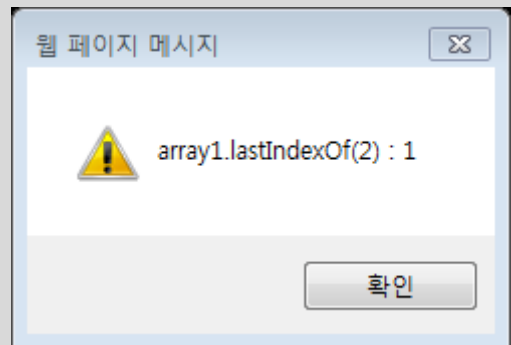
```
<script>
  var array1 = [1, 2, 3, 4, 5];

  alert('Array.isArray(array1) : ' + Array.isArray(array1));
  alert('Array.isArray(6) : ' + Array.isArray(6));
  ③ alert('array1.indexOf(4) : ' + array1.indexOf(4));
  ④ alert('array1.lastIndexOf(2) : ' + array1.lastIndexOf(2));
  ⑤ alert('array1.lastIndexOf(6) : ' + array1.lastIndexOf(6));
</script>
```

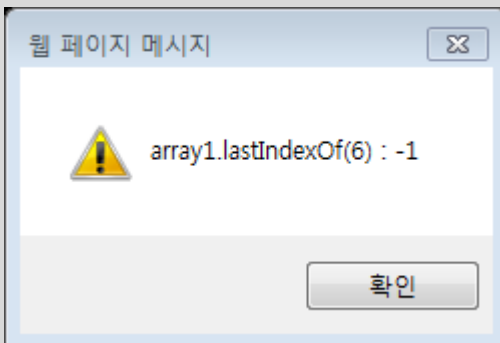
③ 출력 화면



④ 출력 화면



⑤ 출력화면



1. Array 객체

3) 반복

- forEach()

- 배열 각각의 요소에 대해 지정된 작업을 실행하는 메서드
- 함수를 매개변수로 입력하여 작업을 지정하며, for in 문과 유사
- 함수는 배열 요소의 값, 배열 요소의 숫자 인덱스, 배열 요소를 포함하는 배열을 의미하는 3개의 매개변수를 가짐

- map()

- 배열의 각각의 요소를 변경해 새로운 배열을 리턴하는 메서드
- 배열 요소를 매개변수로 가지는 함수를 입력하여 호출

- 예

• forEach() 메서드 동작 알아보기

```
<script>

var array1 = [1, 2, 3, 4, 5];
var sum=0;
var out = '';

array1.forEach(function (value, index, array) {
    sum += value;
    out += 'index ' + index + '의 배열 요소 : ' + value + '\n';

    if (index == array.length-1) {
        out += '배열 요소들의 합은 ' + sum + '입니다';
    }
});
alert(out);

out = array1.map(function (value) {
    return value + 10;
});

alert(out);
</script>
```

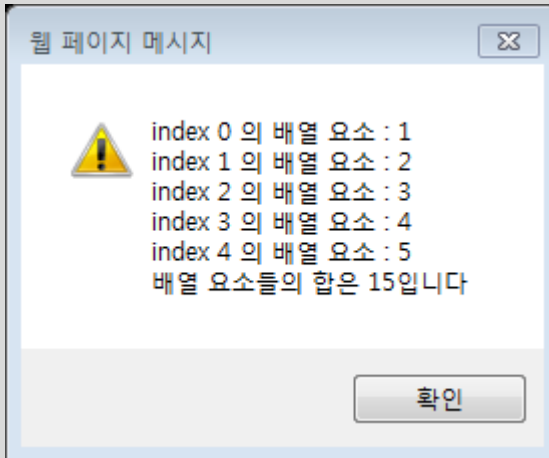
- 각 배열 요소들의 값을 sum에 더함
- 배열 요소의 인덱스와 배열 요소의 값을 out에 저장
- 배열의 마지막 인덱스까지 반복하면 배열 요소를 더한 값을 출력하도록 설정

1. Array 객체

3) 반복

- 예

- forEach() 메서드 호출 결과 : index 0부터 index4까지의 배열 요소 및 그 요소들의 합까지



- map() 메서드 동작 알아보기

<script>

var array1 = [1, 2, 3, 4, 5];

var sum=0;

var out = '';

```
array1.forEach(function (value, index, array) {
    sum += value;
    out += 'index ' + index + ' 의 배열 요소 : ' + value + 'Wn';
```

```
    if (index == array.length-1) {
        out += '배열 요소들의 합은 ' + sum + '입니다';
```

}

});

alert(out);

```
    out = array1.map(function (value) {
        return value + 10;
```

});

alert(out);

</script>

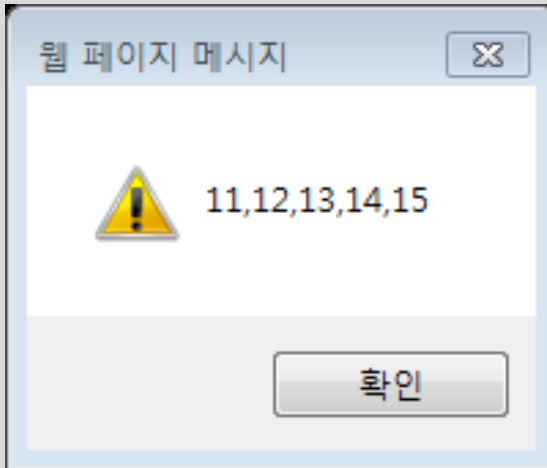
- 배열 요소의 값을 매개변수로 가지는 함수를 이용하여 호출
- 각각의 배열 요소에 10을 더한 후 리턴하도록 설정
- 리턴된 값은 out에 저장

1. Array 객체

3) 반복

- 예

- map() 메서드 호출 결과 : 각각의 배열 요소들에 10이 더해짐



4) 조건

- filter()

- 조건을 만족하는 요소를 추출해 새로운 배열을 생성하는 메서드
- 매개변수로 입력한 함수는 불린 값을 리턴해야 하며, 리턴 값이 true 인 것만을 모아 새로운 배열 생성

- every()

- 모든 배열 요소가 조건을 만족하는 확인하는 메서드
- 매개변수로 입력한 함수는 불린 값을 리턴해야 하며, 모든 배열 요소가 true를 리턴할 경우에만 true를 리턴

- some()

- 하나 이상의 배열 요소가 조건을 만족하는 확인하는 메서드
- 매개변수로 입력한 함수는 불린 값을 리턴해야 하며, 배열 요소 중 하나라도 true를 리턴하면 true를 리턴

1. Array 객체

5) 연산

- reduce()

- 배열 요소가 하나가 될 때까지 왼쪽부터 두 개씩 묶는 함수를 실행하는 메서드
- 매개 변수로 입력된 함수의 리턴 값이 다시 함수의 매개변수 입력

- reduceRight()

- 배열 요소가 하나가 될 때까지 오른쪽부터 두 개씩 묶는 함수를 실행하는 메서드

2. JSON 객체

1) 개요와 메서드

- 개요

- ECMAScript 5부터 정식으로 JSON(JavaScript Object Notation) 객체 지원
- JavaScript 객체의 형태를 가지는 문자열을 의미
- 인터넷에서 자료를 주고받을 때 그 자료를 표현하는 방법

- JSON.stringify()

- JavaScript 객체를 JSON 문자열로 변환하는 메서드
- 매개변수로 입력된 객체에 toJSON() 메서드가 없다면 객체 전체를 JSON 문자열로 변환

- JSON.parse()

- JSON 문자열을 JavaScript 객체로 변환하는 메서드

- 예

```
<script>
  var car = {
    name: 'car1',
    model: 400,
    color: 'black'
  };

  var student = {
    name: '홍길동',
    korean: 99,
    toJSON: function () {
      return {
        history: 98
      };
    }
  }

  ① alert(JSON.stringify(car));
  ② alert(JSON.stringify(student));

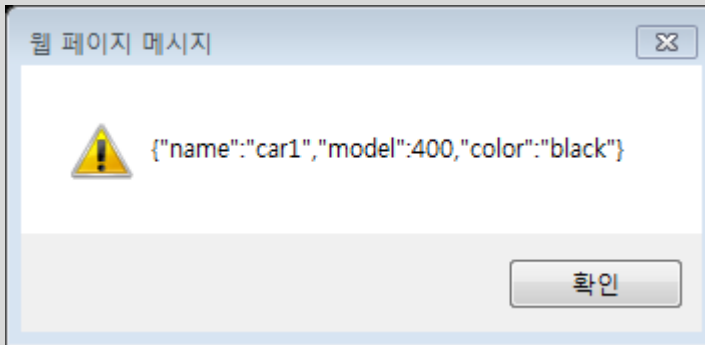
  ③ var car1 = JSON.parse('{"name":"car1","model":400,"color":"black"} ');
    alert(car1.name + ':' + car1.model + ':' + car1.color);
</script>
```

2. JSON 객체

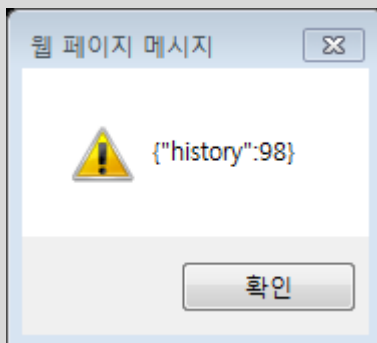
1) 개요와 메서드

- 예

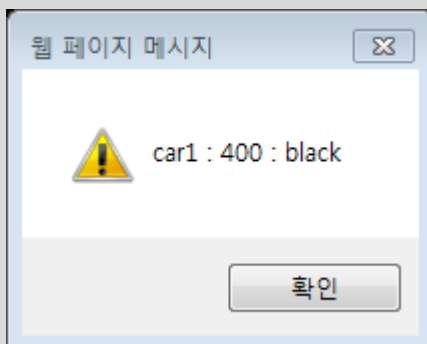
① 출력 화면 : car 객체 전체의 속성과 속성값



② 출력 화면 : student 객체에 toJSON() 메서드에서 리턴한 속성과 속성값만 문자열로 변환됨



③ 출력 화면 : JSON 문자열이 객체로 변환됨



3. String 객체

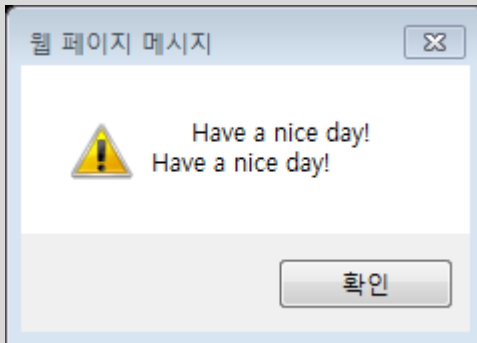
1) 추가 메서드

- trim()

- ECMAScript 5에서 String 객체에 추가된 메서드
- 문자열 양쪽 끝의 공백을 제거
- 문자열 자체가 변하지는 않음
- 예

```
<script>
  var out = '  Have a nice day!  ';
  alert(out + '\n' + out.trim());
</script>
```

- 출력 화면 : 문자열 양쪽 끝의 공백이 사라짐



4. Object 객체

1) 추가 메서드

메서드	설명
Object.defineProperty()	객체에 속성 추가
Object.defineProperties()	객체에 속성들을 추가
Object.create()	객체 생성
Object.preventExtensions()	객체 속성의 추가 제한
Object.isExtensible()	객체 속성의 추가가 가능한지 확인
Object.seal()	객체 속성의 삭제 제한
Object.isSealed()	객체 속성의 삭제가 가능한지 확인
Object.freeze()	객체 속성의 삭제와 수정을 제한
Object.isFrozen()	객체 속성의 삭제와 수정이 가능한지 확인
Object.keys()	순환 가능한 객체 자신의 소유의 속성을 배열로 생성
Object.getOwnPropertyNames()	모든 객체 자신 소유의 속성을 배열로 생성
Object.getOwnPropertyDescriptor()	특정 속성의 옵션 객체를 추출

4. Object 객체

2) 객체 속성 추가

- Object.defineProperty()

- 객체에 속성을 추가하는 메서드
- 속성을 추가하려는 객체, 속성의 이름, 속성 관련 옵션을 매개변수로 입력
- 객체 속성 관련 옵션

옵션	설명	기본값
value	속성의 값을 저장	undefined
set	세터를 정의 (value, writable 옵션과 함께 사용할 수 없음)	undefined
get	게터를 정의 (value, writable 옵션과 함께 사용할 수 없음)	undefined
configurable	속성의 옵션 값 설정을 변경할 수 있는지 여부를 결정	false
enumerable	for in 반복문으로 검사할 수 있는지 결정	false
writable	속성의 값을 변경할 수 있는지 여부를 결정	false

- Object.defineProperties()

- Object.defineProperty() 메서드는 객체에 하나의 속성만을 지정할 수 있지만, 여러 개의 속성을 지정할 수 있음
- 속성을 추가하려는 객체, 속성과 그에 대응하는 옵션 객체를 매개변수로 입력

4. Object 객체

3) 객체 생성

- Object.create()

- 객체를 생성하는 메서드
- 기존의 객체의 속성과 메서드를 포함하여 새로운 속성을 추가하여 객체 생성
- 원본이 되는 객체와 추가하고자 하는 속성과 옵션을 매개변수로 입력

4) 동적 속성 수정 제한

- Object.preventExtensions()

- 객체의 속성을 추가하는 것을 제한

- Object.isExtensible()

- 객체의 속성을 추가할 수 있는지 확인

- Object.seal()

- 객체의 속성을 삭제하는 것을 제한
- delete 키워드를 이용하여 객체의 속성을 삭제 (예: delete student.name)

- Object.isSealed()

- 객체의 속성을 삭제할 수 있는지 확인

- Object.freeze()

- 객체의 속성의 수정과 삭제를 제한

- Object.isFrozen()

- 객체의 속성을 수정과 삭제할 수 있는지 확인

4. Object 객체

5) 객체 보조

- Object.keys()

- 속성의 enumerable 옵션 값이 true인 속성만을 배열로 생성하는 메서드

- Object.getOwnPropertyNames()

- 속성의 enumerable 옵션 값이 false인 속성을 포함하여 모든 속성을 배열로 생성하는 메서드

- Object.getOwnPropertyDescriptor()

- 특정 속성의 옵션 객체를 추출하는 메서드

▣ 정리하기

1. ECMAScript 5 Array 객체

- Web Browser의 버전에 따라 ECMAScript 5를 지원하지 않을 수도 있음
- Array 객체에는 isArray(), forEach() 메서드를 포함하여 10개의 메서드가 추가됨

2. ECMAScript 5 JSON 객체

- JSON 객체는 JavaScript 객체의 형태를 가지는 문자열을 의미
- ECMAScript 5부터 정식으로 JSON 객체를 지원함

3. ECMAScript 5 String 객체

- String 객체에는 문자열 양쪽 끝의 공백을 제거하는 trim() 메서드가 추가됨

4. ECMAScript 5 Object 객체

- Object 객체에는 defineProperty(), create() 메서드를 포함하여 12개의 메서드가 추가됨