

# 웹 앱 개발을 위한 JavaScript 기초 강의 노트

## 제 8회차 객체의 개요

### ■ 학습목표

- 객체에 대해 이해하고 생성할 수 있다.
- 속성과 메서드에 대해 이해하고, 속성값에 접근하여 수정할 수 있다.
- for in문과 in과 with 키워드를 이해하고 사용할 수 있다.

### ■ 학습내용

- 객체의 개요
- 객체 관련 키워드

## 1. 객체의 개요

### 1) 객체란?

#### - 객체

- 변형 가능한 속성들의 집합
- 이름과 값이 있는 속성들을 포함하는 컨테이너
- 데이터를 모으고 구조화 하는데 유용
- JavaScript는 다른 객체지향 언어와 달리 클래스 없이 객체 생성
  - 객체 기반의 스크립트 언어
  - 많은 부분이 객체로 존재하므로 객체가 매우 중요

#### - 객체의 생성

- 중괄호({ })를 이용하여 생성
- 중괄호({ })안에 아무것도 적지 않으면 빈 객체 생성
- 중괄호({ })안에 속성값 표현
  - “속성이름” : “속성값”(속성 이름 : ‘속성값’) 형태로 표기
  - 쉼표를 이용하여 구분

#### - 속성이름과 속성값의 구분

속성이름	속성값
<ul style="list-style-type: none"> <li>• 빈 문자열을 포함하여 어떤 문자열이나 숫자 가능</li> <li>• 사용 가능한 식별자</li> <li>• 예약어가 아닐 경우 속성이름을 감싸고 있는 따옴표(“ or ‘) 생략 가능</li> </ul>	<ul style="list-style-type: none"> <li>• 객체를 포함하여 값을 나타내는 어떤 표현식도 가능</li> <li>• 이 값이 함수일 경우, 해당 속성을 메서드라 함</li> </ul>

## 1. 객체의 개요

### 1) 객체란?

- 속성값 읽기

- 대괄호([ ]) 표기법  
→ 객체 뒤의 대괄호 안에 속성 이름을 적어 속성에 접근
- 마침표(.) 표기법  
→ 객체 뒤에 마침표를 찍고, 속성 이름을 적어 속성에 접근  
→ 대괄호 표기법보다 간단하고 읽기가 편해 마침표 표기법을 더 많이 사용
- 예

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

  alert(car["name"]);
  alert(car.model);
  alert(car.color);
  alert(car.weight);
</script>
```

## 1. 객체의 개요

### 2) 속성과 메서드

#### - 속성

- 모든 형태의 자료형을 가질 수 있음  
→ 숫자, 문자열, 불린, null, undefined, 배열, 함수, 객체 등

#### - 메서드

- 속성값이 함수인 속성
- 객체의 속성이름 뒤에 괄호를 열고 닫아 접근

#### - 자기 자신이 가진 속성을 이용하고자 할 때

- 자신이 가진 속성임을 분명하게 표시해야 함
- this 키워드를 이용하여 정확히 표시
- JavaScript에서는 같은 객체 내부에서 this 키워드 생략할 수 없음

#### - 예

- car is running을 car1 is running으로 수정하려는 경우

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

  company: {
    start:1997,
    logo:'star'
  },

  run : function(){
    alert(name+' is running.');
```

```
    }
  };

  alert(car.company.logo);
  car.run();
</script>
```

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

  company: {
    start:1997,
    logo:'star'
  },

  run : function(){
    alert(this.name+' is
    running.');
```

자기 자신이 가진 속성을  
이용한다는 것 표시

```
    }
  };

  alert(car.company.logo);
  car.run();
```

## 2. 객체 관련 키워드

### 1) in 키워드

#### - for in문

- 객체에 포함된 모든 속성에 대해 반복을 실행
- 단순 for문을 이용하여 객체의 속성을 확인하기 어려움
- 형식

```
for(var i in [객체 / 배열] ){  
    }  
}
```

- 각 반복의 시작 부분에서 변수의 값은 객체의 다음 속성이거나 배열의 인덱스
- 반복문 안에서 변수를 사용하여 객체의 속성 또는 배열 요소를 참조 가능

## 2. 객체 관련 키워드

### 1) in 키워드

- in 키워드

- for문과 독립적으로 사용하면 확인하고자 하는 속성이 객체 안에 있는 지 알 수 있음
- 형식

속성 in 객체

❖ 존재하면 **true 반환**

❖ 존재하지 않으면 **false 반환**

- 예

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

  company: {
    start:1997,
    logo:'star'
  },

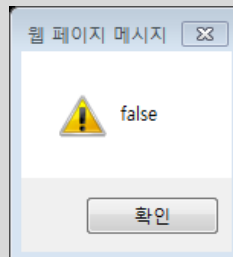
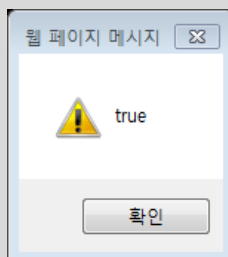
  run : function(){
    alert(this.name+' is running. ');
  }
};

alert('name' in car);
alert('company' in car);
alert('run' in car);

alert('test' in car);

</script>
```

- 결과



## 2. 객체 관련 키워드

### 2) with 키워드

#### - with 키워드

- 반복할 객체를 생략하고 속성이나 메서드를 사용할 수 있도록 함
- 코드를 더욱 간결하게 작성할 수 있음
- 형식

```
with (객체){  
    실행할 코드  
}
```

- 사용 시 자바스크립트를 최적화하기 어려워, 사용하지 않은 코드보다 실행 속도 떨어짐
- 사용 자제를 추천
- 예 1

```
<script>  
    var car = {  
        name : 'car1',  
        model : 400,  
        color: 'black',  
  
        company: {  
            start:1997,  
            logo:'star'  
        },  
  
        run : function(){  
            alert(this.name+' is running.');        }  
    };  
    var out = "";  
    with (car) {  
        out += 'name : Wt' + name + 'Wn';  
        out += 'model : Wt' + model + 'Wn';  
        out += 'color : Wt' + color + 'Wn';  
        out += 'company start : Wt' + company.start + 'Wn';  
        out += 'company logo : Wt' + company.logo + 'Wn';  
    }  
    alert(out);  
</script>
```



## 2. 객체 관련 키워드

### 2) with 키워드

- with 키워드

#### • 예 2

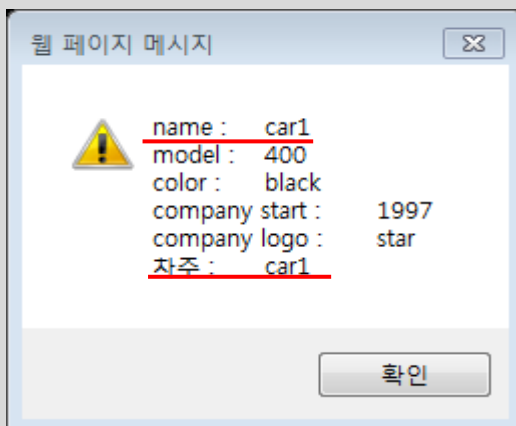
```
<script>
  var car = {
    name : 'car1',
    model : 400,
    color : 'black',

    company : {
      start : 1997,
      logo : 'star'
    }
  };
  var name = '홍길동';
  var out = '';
  with (car) {
    out += 'name : ' + name + '\n';
    out += 'model : ' + model + '\n';
    out += 'color : ' + color + '\n';
    out += 'company start : ' + company.start + '\n';
    out += 'company logo : ' + company.logo + '\n';
    out += '차주 : ' + name + '\n';
  }
  alert(out);
</script>
```

→ 객체 속성 이름

→ 차주를 저장하는 외부 변수명

↓  
차주를 출력에 포함시키기 위해  
out에 name값을 더함



[ 객체 속성 이름을 외부 변수보다 우선 ]

## 2. 객체 관련 키워드

### 2) with 키워드

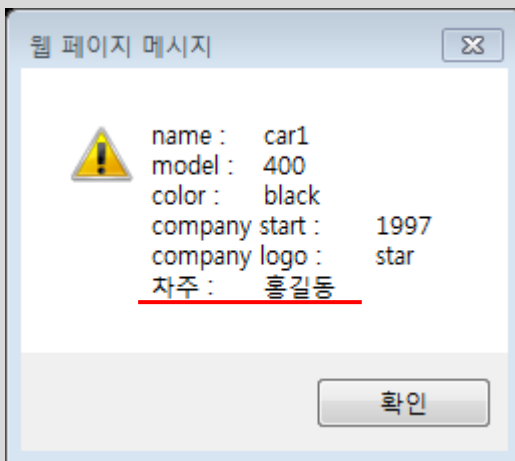
- with 키워드

#### • 예 3

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

    company: {
      start:1997,
      logo:'star'
    }
  };
  var name = '홍길동';
  var out = '';
  with (car) {
    out += 'name : Wt' + name + 'Wn';
    out += 'model : Wt' + model + 'Wn';
    out += 'color : Wt' + color + 'Wn';
    out += 'company start : Wt' + company.start + 'Wn';
    out += 'company logo : Wt' + company.logo + 'Wn';
    out += '차주 : Wt' + name + 'Wn';
  }
  alert(out);
</script>
```

window.name 지정



## ▣ 정리하기

### 1. 객체의 개요

- **객체**는 JavaScript의 기본 자료형을 제외한 모든 값이며, 이름과 값이 있는 속성들을 포함하는 컨테이너라고 할 수 있음
- 속성은 모든 자료형의 값을 가질 수 있으며, 특히 함수가 속성값일 때는 **메서드**라고 함
- 속성값은 **대괄호([ ]) 표기법**과 **마침표(. ) 표기법**을 이용하여 접근하여 읽고 수정할 수 있음

### 2. 객체 관련 키워드

- **for in 문**은 객체에 포함된 속성을 살펴보기 위해 사용함
- **in**은 속성이 객체에 있는지 확인하기 위해 사용하는 키워드임
- **with**는 반복할 객체를 생략하고 속성이나 메서드를 사용할 수 있게 하는 키워드임