

# 객체





객체



# 객체

## 객체 선언/호출

- 객체는 키값을 사용하여 속성(멤버변수) 식별
- 중괄호를 사용하여 객체 생성
- 대괄호([]) 또는 점(.)으로 요소의 값에 접근
- 속성에 모든 자료형이 올 수 있으며, 그 중 함수 자료형인 요소를 메소드라고 함
- 객체 내에서 자신의 속성을 호출할 때 반드시 this 키워드 사용
- 객체의 모든 속성을 출력하기 위해서는 for in 문을 사용
  - 일반 for문이나 while문으로는 전체 출력 불가



# 객체

## 객체 선언

```
var 변수명(객체명) = {  
    속성1(키값) : 값1,  
    속성2(키값) : 값2,  
    속성3(키값) : 값3  
};
```

## 속성값 접근

```
변수명(객체명)['키값'];  
또는  
변수명(객체명).키값;
```



# 객체

## in / with 키워드

- in : 객체 내부에 해당 속성이 있는지 확인하는 키워드
- with : 코드를 줄여주는 키워드, 호출 시 객체명 생략 가능

### in

속성명 in 변수명(객체명)                      //객체에 해당 속성이 있으면 true

### with

```
with(변수명(객체명)){  
    속성명;  
    속성명;  
}
```



# 객체

## 객체 속성 추가 및 삭제

- 이미 생성된 객체에 추가적인 속성 및 메소드를 동적으로 추가 및 삭제가 가능

### 추가

변수명(객체명).추가속성명='값'    //객체에 해당 속성이 있으면 true

### 삭제

delete(변수명(객체명).속성명);



# 객체

## 객체 배열 활용

- 생성된 객체를 배열에 넣어 활용 가능

## 사용

```
var 변수명 = new Array();  
변수명.push({속성명:'값', 속성명:'값', 속성명:'값'});  
변수명.push({속성명:'값', 속성명:'값', 속성명:'값'});  
변수명.push({속성명:'값', 속성명:'값', 속성명:'값'});
```



# 객체

## 함수 활용 객체 생성

- 함수의 매개변수에 필요한 속성값을 다 받아서 객체를 생성 후 리턴

## 사용

```
function 함수명(값1,값2,값3,...){  
    var 변수명(객체명) = {  
        속성:값1,  
        속성:값2,  
        속성:값3  
    }  
    return 변수명(객체명);  
}
```





# 객체

## 생성자 함수

- this 키워드를 사용하여 속성을 생성하는 함수
- new라는 키워드를 사용하여 객체 생성
- 생성자명의 첫 글자는 대문자로 시작

## 사용

```
function 생성자명(값1,값2,값3,...){  
    this.속성 = 값1;  
    this.속성 = 값2;  
    this.속성 = 값3;  
}
```

