

객체

▶ 객체 선언 및 호출

객체는 키 값을 사용하여 속성(멤버 변수) 식별
중괄호를 사용하여 객체를 생성하고 [] 또는 .으로 요소 값에 접근
속성에 모든 자료형이 올 수 있고 그 중 함수 자료형 요소를 메소드라고 함
객체 내에서 자신의 속성을 호출할 때 반드시 this키워드를 사용하고
객체의 모든 속성을 출력하려면 단순 for문이나 while문으로는 출력이 불가
하니 for in문을 사용해야 함

* 식별자로 사용할 수 없는 문자(띄어쓰기, 특수문자)를 속성으로 사용할 경우
' '로 묶어서 선언하고 접근 시에는 []만 가능

▶ 객체 선언 및 호출

✓ 선언 방법

```
var 변수 명(객체 명) = {  
    속성(키 값) : 값,  
    속성(키 값) : 값,  
    속성(키 값) : 값  
};
```

✓ 속성 값 접근

```
변수 명(객체 명)['요소 명(키 값)'];  
또는  
변수 명(객체 명).요소 명(키 값);
```

▶ 함수 활용 객체 생성

함수의 매개변수에 필요한 속성 값을 다 받아서 객체 생성 후 리턴

```
function 함수 명(value1, value2, value3, ...){  
    var 변수 명(객체 명) = {  
        속성 명: 'value1';  
        속성 명: "value2";  
        ...  
    }  
    return 변수 명(객체 명);  
}
```

▶ 객체 속성 추가 및 삭제

속성 및 메소드를 동적으로 추가/삭제 가능

추가

```
변수 명(객체 명).속성 명 = '값';  
변수 명(객체 명).속성 명 = "값";  
변수 명(객체 명).속성 명 = function(){  
    메소드 로직;  
    [return [리턴 값];]  
};
```

삭제

```
delete(변수 명(객체 명).속성 명);
```

▶ 객체 배열 활용

생성된 객체를 배열에 넣어 활용 가능

```
var 변수 명 = [ ]; // 배열 생성
```

```
데이터 대입 : 변수 명.push({속성 명: "값", 속성 명: "값", ...});
```