Ch06 객체

서로 연관된 변수와 함수를 그룹핑하고 이름을 붙인 것

* 1. 객체 개요

**var array = [‘사과’, ‘바나나’, ‘망고’,’딸기’]; /\* 여러가지 변수를 모아놓은 자료형 typeof() 이용시 object \*/**

**array[0] : ‘사과’**

**var product = {제품명 : ‘7D 건조 망고’,**

**유형 : ‘당절임’,**

**성분 : ‘망고, 설탕, 메타중아황산나트륨, 치자황색소’,**

**원산지 : ‘필리핀’**

**};**

**product[‘제품명’] : ‘7D 건조 망고’**

**product.제품명 : ‘7D 건조 망고’**

* 1. 속성과 메서드

※ **배열 내부에 있는 값을 요소(element). 객체 내부에 있는 값은 속성(property). 자바 스크립트에서는 요소와 속성을 구분하지 않고 쓴다. 객체 속성 중 함수형 자료형인 속성을 특별히 메서드라고 부른다.**

* This 키워드 생략 불가

<script>

var person = {

name: '홍길동',

eat: function (food) {alert(this.name + '이 ' + food + '을/를 먹습니다.'); }

};

person.eat('밥');

</script>

* 1. 객체와 반복문

배열은 단순 for 반복문과 for in 반복문으로 요소에 쉽게 접근. 객체는 단순 for문은 불가. for in반복문을 사용한다

* 1. 객체의 속성 추가와 제거

6.5.1 속성 추가

<script>

// 변수를 선언합니다.

var student = {};

student.name = '윤인성';

student.hobby = '악기';

student.talent = '프로그래밍';

student.hope = '생명공학자';

// toString() 메서드를 만듭니다.

student.toString = function () {

var output = '';

for (var key in this) {

// toString() 메서드는 출력하지 않게 합니다.

if (key != 'toString') {

output += key + '\t' + this[key] + '\n';

}

}

return output;

};

// 출력합니다.

alert(student.toString());

</script>

6.5.2 속성 제거 ; delete 이용

<script>

var student = {};

student.name = '윤인성';

student.hobby = '악기';

student.talent = '프로그래밍';

student.hope = '생명공학자';

// toString() 메서드를 만듭니다.

student.toString = function () {

var output = '';

for (var key in this) {

// toString() 메서드는 출력하지 않게 합니다.

if (key != 'toString') {

output += key + '\t' + this[key] + '\n';

}

}

return output;

}; // 출력합니다.

alert(student.toString());

delete (student.hope); // 속성을 제거합니다.

// toString() 메서드를 사용하지 않아도 toString() 메서드를 사용합니다

alert(student);

</script>

* 1. 객체와 배열을 사용한 데이터 관리

<script>

// 학생 정보 배열을 만듭니다.

var students = [];

students.push({ name: '윤인성', kor: 87, mat: 98, eng: 88, sci: 95 });

students.push({ name: '연하진', kor: 92, mat: 98, eng: 96, sci: 98 });

students.push({ name: '구지연', kor: 76, mat: 96, eng: 94, sci: 90 });

students.push({ name: '나선주', kor: 98, mat: 92, eng: 96, sci: 92 });

students.push({ name: '윤아린', kor: 95, mat: 98, eng: 98, sci: 98 });

// 모든 students 배열 내의 객체에 메서드를 추가합니다.

for (var i in students) {

// 총점 구하는 메서드를 추가합니다.

students[i].getSum = function () {

return this.kor + this.mat + this.kor + this.sci;

};

// 평균을 구하는 메서드를 추가합니다.

students[i].getAverage = function () {

return this.getSum() / 4;

};

}

// 출력합니다.

var output = '이름\t총점\t평균\n';

for (var i in students) {

with (students[i]) {

output += name + '\t' + getSum() + '\t' + getAverage() + '\n';

}

}

alert(output);

</script>

* 1. 함수를 사용한 객체 생성

<script>

function makeStudent(name, korean, math, english, science) {

var willReturn = {

// 속성

name: name,

kor: korean,

mat: math,

eng: english,

sci: science,

// 메서드

getSum: function () {

return this.kor + this.mat + this.eng + this.sci;

},

getAverage: function () {

return this.getSum() / 4;

},

toString: function () {

return this.name + '\t' + this.getSum() + '\t' + this.getAverage();

}

};

return willReturn;

}

// 학생 정보 배열을 만듭니다.

var students = [];

students.push(makeStudent('윤인성', 87, 98, 88, 95));

students.push(makeStudent('연하진', 92, 98, 96, 98));

students.push(makeStudent('구지연', 76, 96, 94, 90));

students.push(makeStudent('나선주', 98, 92, 96, 92));

students.push(makeStudent('윤아린', 95, 98, 98, 98));

students.push(makeStudent('윤명월', 64, 88, 92, 92));

students.push(makeStudent('김미화', 82, 86, 98, 88));

students.push(makeStudent('김연화', 88, 74, 78, 92));

students.push(makeStudent('박아현', 97, 92, 88, 95));

students.push(makeStudent('서준서', 45, 52, 72, 78));

// 출력합니다.

var output = '이름\t총점\t평균\n';

students.forEach(function (data) {

output += data;

});

Document.write(output);

alert(output);

</script>