

# 객체

- Object가 객체
- 배열과 객체는 비슷
- 배열은 요소에 접근할 때 인덱스 사용, 객체는 키를 사용

## 객체 요소 접근 방법

- product['제품명']
- product.제품명

→ 문자열을 키로 사용하는 경우 무조건 대괄호를 사용해야 객체의 요소에 접근 가능

## 속성과 메서드

- 배열과 마찬가지로 객체의 속성도 모든 형태의 자료형을 가질 수 있다. (number, String, boolean, array, method)
- 객체의 속성 중 함수 자료형인 속성을 특별히 메서드라 부름

### 속성과 메서드의 구분

```
<script>
    var person = {
        name: "",
        eat: function (food) {
            alert(this.name + ' ' + food + ' ');
        }
    };

    person.eat(' ');
</script>
```

→ eat속성은 함수 자료형이므로 특별히 eat()메서드라고 부름

→ 자바스크립트는 다른 언어와 다르게 같은 객체 내부에서 this 키워드를 생략 할 수 없다.

## 객체와 반복문

객체는 for문으로 객체의 속성 탐색 불가 → for in 반복문 사용해야 함

### 객체의 반복문 for in

```
<script>
    var product = {
        name: "MicroSoft",
        price: '15000',
        language: '',
        supportOS: 'Win 32/64',
        subscription: true
    };

    var output = '';

    for(var key in product){
        output += '- ' + key + ': ' + product[key] + '\n';
    }
    alert(output);

</script>
```

→ 객체의 요소만큼 반복문 실행

## in키워드, with키워드

## in

```
var output = '';
var student = {
  : "",
  : 92, : 98,
  : 96, : 98
};

output += ' ' in student: " + (' ' in student) + '\n'; //true
output += ' ' in student: " + (' ' in student);          //false

alert(output);
```

## with

```
<script>
  var output = '';
  var student = {
    : "",
    : 92, : 98,
    : 96, : 98
  };

  with (student) { //student. -->
    output += " : " + + '\n';
    output += " : " + + '\n';
  }

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

## 동적으로 메서드 추가

```
<script>
  var student = {};

  student. = "";
  student. = "";
  student. = "";
  student. = "";

  student.toString = function () {
    var output="";
    for(var key in this){
      if(key != "toString"){
        output += key + '\t' + this[key] + '\n';
      }
    }
    return output;
  };

  alert(student.toString());

</script>
```

- delete(student.장래희망) 으로 속성 삭제 가능

## 전개 연산자를 사용한 배열 테크닉

포인터스러운 배열을 for문으로 일일이 안돌려도 깊은 복사 가능함

배열 병합에도 쉬움

#### 전개 연산자

```
<script>

  //
  const arrayA = [1,2,3,4,5];
  const arrayB = [6,7,8,9,10];

  const newArray = [...arrayA];
  newArray[0] = 0;
  alert(newArray);

  //
  const mergedArray = [...arrayA, ...arrayB];
  alert(mergedArray);

</script>
```