- **06. 객체(object)** // src: 5_JavaScripts/ch06_객체 참고
- 1) 객체(object): 변수를 모아 놓은 자료형 type //{}; 내부에 변수가 없어도 object
- 2) 속성(property)과 메소드(method);
- (1) 요소(element): 배열 내부 값 var arr = [1, 2, 3]; //1, 2, 3:element
- (2) 속성(property): 객체 내부 값 var person = {name:'kong'}; //'kong': property, name: key ※자바스크립트에서는 요소와 속성을 구분하지 않고 사용 // [1, 2, 3] = {0:1, 1:2, 2:3}
- (3) 메소드(method): 객체 속성 중 함수형 자료형 속성 //객체 속성 값으로 함수 가능
- 3) 객체 출력
- (1) 배열은 일반 for문, for-in문, forEach()함수 모두 사용 가능
- (2) 객체는 for-in문 사용해서 출력
- 4) 객체 관련 키워드: in, with
- (1) in 키워드: 해당 key가 객체 안에 있는지 확인
- (2) with 키워드: 반복되는 변수명을 생략 가능하게 해줌, 동일 객체명과 충돌시 'window.' 이용
- 5) 객체 속성의 추가 및 제거
- (1) 속성 추가: 변수.속성명(key) = '속성값';
- (2) 속성 제거: delete(변수.속성명);
- 6) 객체와 배열을 사용한 데이터 관리; 모든 배열 내 객체에 메소드 추가 가능
- 7) 함수를 이용한 객체 생성; 데이터 입력시 오탈자 방지(특히 속성명(key) 오탈자)
- **07. 생성자 함수(constructor)** // src: 5_JavaScripts/ch07_생성자함수 참고
 - 1) 생성자 함수: 생성자란 객체지향 프로그램에서 객체를 생성할 때 호출되는 단위 메소드 function 생성자함수명(속성명들) { };

var 객체명 = new 생성자함수명(속성값들); //new 키워드로 객체(instance) 생성

- 2) 프로토 타입(prototype): 생성자 함수로 생성된 객체가 공통으로 가지는 공간
- ⇒ 메소드는 객체가 아닌 프로토 타입 안에 넣어줌 (∵객체 생성시마다 동일 함수 생성 방지) 생성자함수명.prototype.메소드명 = function() { };
- 3) new 키워드: new 키워드로 함수 호출시 객체 위한 공간 생성

※this 키워드: new 키워드 사용시 생성된 해당 공간 의미, 그렇지 않은 경우 window객체 나타냄

- 4) 캡슐화: 특정 속성이나 메소드를 사용자가 바로 사용할 수 없게 막는 것
- ⇒ 숨겨놓은 특정 속성은 getter와 setter를 이용해서 간접적으로 접근 가능

※예외처리: throw, try ~ catch 등 이용 //다른 페이지로 이동시: location.href=";

5) 상속: 기존 생성자 함수나 객체를 기반으로 새로운 생성자 함수나 객체를 쉽게 만듦