

10장 객체 리터럴

10.1 객체란 ?

- 자바스크립트를 구성하는 거의 모든 것이 객체이다. 객체는 0개 이상의 프로퍼티로 구성된 집합이며 프로퍼티는 키와 값으로 구성
- 원시 값을 제외한 나머지 값은 모두 객체이다. 객체는 변경 가능한 값 (mutable value)
- 함수는 일급객체로 값으로 취급 가능하다. 따라서 함수도 프로퍼티 값으로 사용가능하다. 객체내의 함수를 메서드라고 부른다.
- 객체의 집합으로 프로그램을 표현하려는 프로그래밍 패러다임을 객체지향 프로그래밍이라 한다.

10.2 객체 리터럴에 의한 객체 생성

- **인스턴스란?** 클래스에 의해 생성되어 메모리에 저장된 실체를 말한다. 인스턴스는 객체가 메모리에 저장되어 실제로 존재하는 것에 초점을 맞춘 용어
- 객체 지향 언어**
 - C++, Java 클래스 기반 객체지향 언어 클래스를 사전 정의 후 new 연산자와 함께 생성자를 호출하여 인스턴스 생성
 - JavaScript 프로토타입 기반 객체지향 언어 클래스로만 객체 생성 다양한 객체 생성 방법을 지원
- JS 객체 생성방법**
 - 객체 리터럴
 - Object 생성자 함수
 - 생성자 함수
 - Object.create 메서드
 - 클래스(ES6)
- 객체 리터럴은 객체를 생성하기 위한 표기법이며 가장 일반적이고 간단한 방법 이 외 다른 생성 방식은 함수를 사용해 객체를 생성한다.
- 객체 리터럴의 종괄호는 코드 블록을 의미하지 않는다!! 따라서 객체 리터럴의 닫는 종괄호 뒤에는 세미콜론을 꼭 붙여야한다.

10.3 프로퍼티

- 객체는 프로퍼티의 집합이며, 프로퍼티는 키와 값으로 구성된다.
- 프로퍼티는 심표로 구분 맨 마지막에 심표를 붙여도 상관없다.
- 프로퍼티 이름은 식별자 네이밍 규칙을 따르며 " " 따옴표를 안써도 된다. 네이밍 규칙을 따르지 않는다면 "" 를 붙여야 한다.
- 빈 문자열을 프로퍼티 키로 사용해도 에러는 발생 x
- 프로퍼티 키에 문자열이나 심벌 값 외의 값을 사용하면 암묵적 타입 변환으로 문자열로 변경
- 예약어를 키로 사용해도 에러 x 하지만 안쓰는것이 좋다
- 이미 존재하는 프로퍼티 키를 중복 선언하면 나중에 선언한 프로퍼티가 덮어쓴다. 에러 발생 x

10.4 메서드

- 메서드는 객체에 묶여있는 함수

10.5 프로퍼티 접근

- 마침표 표기법** 객체명.프로퍼티 키 식별자 네이밍 규칙 따른 프로퍼티
- 대괄호 표기법** ['프로퍼티 키'] 식별자 네이밍 규칙 따른 프로퍼티
식별자 네이밍 규칙 따르지 않은 프로퍼티
- 객체에 존재하지 않는 프로퍼티에 접근하면 undefined 반환 에러 x

10.6 프로퍼티 값 갱신

- 할당 연산자 사용

10.7 프로퍼티 동적 생성

- 할당 연산자 사용

10.8 프로퍼티 삭제

- Delete 연산자 사용
- 만약 존재하지 않는 프로퍼티 삭제하면 아무런 에러 없이 무시

10.9 ES6에서 추가된 객체 리터럴의 확장 기능

10.9.1 프로퍼티 축약 표현

- 변수 이름과 프로퍼티 키가 동일한 이름일 때 프로퍼티 키를 생략할 수 있다. 이때 프로퍼티 키는 변수 이름으로 자동 생성된다.

10.9.2 계산된 프로퍼티 이름

- 객체 리터럴 내부에서 계산된 프로퍼티 이름으로 키를 동적 생성

10.9.3 메서드 축약 표현

- Function 키워드 생략가능
- But, 축약표현은 함수로 정의한 메서드와 다르게 동작한다. 인스턴스를 생성할 수 없는 non-constructor 프로토타입을 갖지 않음