자바스크립트 특징

자바스크립트 특징

- 자바스크립트는 스크립팅Scripting 언어
- 스크립팅 언어 특징
  - 소스 파일의 코드를 사전에 컴파일하여
    실행 파일을 만들지 놓지 않고
  - 사용하는 시점에 컴파일하고 실행 코드 1
  - 장점을 활용하는 지혜 필요
- 컴파일 순서
  - 소스 파일의 위에서부터 아래로 컴파일
  - function 키워드를 만나면
    function 오브젝트를 생성
  - 이때, 함수 안의 코드는 컴파일하지 않음
    함수가 호출되었을 때, 위의 방법으로 컴파일

JS와 객체지향

- 객체 지향 프로그래밍 언어
  OOP: Object Oriented Programming
- 자바스크립트는 객체 지향 언어
  - ES5 스펙에 아래와 같이 기술되어 있음

ECMAScript is an object-oriented programming language for performing computations and manipulating computational objects within a host environment. ECMAScript as defined here is not intended to be

- 자바스크립트 OOP 구현
  - 다른 언어의 OOP 구현과 차이가 있으므로
    비교하는 것은 의미 없음
  - JS 특징이 반영된 OOP 구현

OOP의 객체

- 강좌에 필요한 것만 간략하게 다룹니다.
- 객체Object
  - 개념적 접근
  - 행위Behavior와 속성Attribute으로 구성
  - 행위: 먹다, 마시다
  - 속성: 밥, 사이다 ++

자바스크립트 객체 형태

JS 객체 형태

- Object 오브젝트 형태
  var book = {
    read: function(param){코드}
  };
  - 인스턴스를 생성할 수 없음
- Function 오브젝트 형태
  - function readBook(param){코드};
  - 객체이지만, OOP의 객체라고 하기에는 부족
  - 객체에 메소드가 하나 있는 모습

prototype

- JS의 OOP 구현 방법
  - prototype에 메소드 연결
  - prototype 형태
- prototype에 연결하여 작성 코드 1
- 다른 언어는 class 안에
  - 메소드와 프로퍼티를 작성하지만
- 자바스크립트는 prototype에
  - 메소드를 연결하여 작성합니다.
  - ES6에서 class에 메소드를 작성 코드 2
  - 하지만, 내부에서 prototype에 연결합니다.

자바스크립트 인스턴스

인스턴스

- Instance
  - Class를 new 연산자로 생성한 것
- 인스턴스 목적
  - Class에 작성된 메소드 사용
  - 인스턴스마다 프로퍼티 값을 유지
  - 예: 축구 경기에서
    각 팀에 적용되는 규칙은 같지만
    팀마다 점수는 다르다.

new 연산자

구분	데이터(값)
constructor	생성자
파라미터	값opt
반환	생성한 인스턴스

- 인스턴스를 생성하여 반환
  - new Book("JS책");
  - 인스턴스를 생성하므로 Book()을 생성자 함수라고 부름
  - 코딩 관례로 첫 문자를 대문자로 작성
- 파라미터
  - 생성자 함수로 넘겨 줄 값을 소괄호()에 작성 코드 1

instanceof

- 오브젝트로 생성한 인스턴스 여부 반환
  - instance instanceof object 코드 1
  - instance 위치에 인스턴스 작성
  - object 위치에 비교 기준 오브젝트 작성
- instance가 object로 생성한
  인스턴스이면 true, 아니면 false 반환

메소드 형태

메소드 호출 형태

- 데이터 형태
  - 배열: ["book", "책", 123]
- OOP의 일반적인 형태 코드 1
- 자바스크립트 형태
  - 데이터로 메소드 호출 코드 2
  - 오브젝트의 함수 호출 코드 3
  - 인스턴스의 메소드 호출 코드 4
- Object 확장
  - 네임스페이스NameSpace로 사용
  - Book = {};
  - Book.Javascript = {}; Book.Html = {};

메소드 사용 팁

- 메소드를 알 수 없을 때
  - MDN 활용
  - MDN에서 "오브젝트 이름"으로 검색
  - 왼쪽의 리스트에서 메소드 이름을 찾습니다.
  - 메소드 이름이 시맨틱을 갖고 있으므로
    메소드 이름으로 기능 예측 가능
- 메소드를 알고 있을 때
  - 기능을 정확하게 사용하기 위해 사용