유노코딩과 함께하는 자바스크립트 기초 강의

객체가 뭐더라? 이전 설명 그대로 다시 보기

자바스크립트는 프로그래밍 언어로써, 소프트웨어적 대상에게 명령을 내리는 역할을 수행한다. 여기에서 소프트웨어적 대상이란, 웹브라우저, 웹 요소, 웹 스타일 등을 의미한다.
명령을 내리고픈 대상에 자바스크립트 명령을 전달하면, 다음 절차대로 작업이 이루어진다.

입력 → 처리 → 출력

자바스크립트 명령문으로 어떤 처리를 지시하고자 하는 대상을 가리켜 '객체'라 한다.

이젠 좀 더 깊게 파헤쳐보자

객체에 내릴 수 있는 명령의 유형은 두 가지다.

객체.데이터

객체가 가지고 있는 숫자, 문자 등의
다양한 데이터를 사용할 수 있다
객체가 가지고 있는 데이터(정보)를 가리켜
일반적으로 '속성'이라 한다

객체.기능( )

객체가 가지고 있는 다양한 기능을
수행할 수 있다(괄호 필수).
객체가 가지고 있는 기능을 가리켜
일반적으로 '메소드'라 한다

즉, 자바스크립트 코드 내에서 객체란 '값 또는 기능을 가지고 있는 데이터'이다.
그리고 개발자는 자신이 원하는 값이나 기능으로 구성된 객체를 직접 만들 수 있다!

객체를 만드는 생성자

생성자란 객체를 생성할 때 사용하는 함수로서, ’생성자 함수’라고도 한다.
사용할 기능을 정의한 일반 함수와 생성자 함수 사이에 특별한 문법적 차이는 존재하지 않는다. 즉, 함수는 생성자 역할을 할 수 있다. 단, 생성자 함수는 객체생성을 목적으로 만드는 것이다.

생성자의 쓸모는?
자바스크립트가 제공하지 않는 유형의 데이터를 창조할 수 있다!

키워드 this 사용하기

생성자 함수 정의 시 this 키워드는 객체 그 자신을 의미한다. this를 이용해 해당 객체의 속성이나 메소드를 추가할 수 있다.

```
// 강아지를 표현하는 'Dog'객체를 만들고 싶다
function Dog(){
    this.name = "콩이"
    this.breed = "시츄"
}
```

=> 일반적으로, 생성자 함수명의 첫글자는 대문자로 한다(필수가 아닌 관례)

객체 생성은 new 연산자로

생성자 함수는 '객체를 이렇게 만들겠습니다'에 대한 정의일 뿐이며, 실제 객체가 생성되기 위해서는 new 연산을 통해 객체를 반환해야 한다.

```
function Dog(){
  this.name = "콩이"
  this.breed = "시츄"
}

// Dog 객체를 생성하고, myDog 이라는 이름을 붙이겠다
const myDog = new Dog();
console.log(myDog.name) // 콩이
console.log(myDog.breed) // 시츄
```

생성자 하나로 여러 개 만들기

생성자 함수는 '객체를 이렇게 만들겠습니다'에 대한 정의, 즉 '설계도'의 역할을 한다.
따라서 생성자 함수 하나로 객체를 여러 개 만들 수도 있다.

```
// 생성자에도 매개변수를 정의할 수 있다.
function Dog(eachName, eachBreed){
  this.name = eachName
  this.breed = eachBreed
}

// 생성 시에도 인자를 전달할 수 있다.
const one = new Dog("바둑이", "진돗개");
const two = new Dog("하나코", "시바견");
const three = new Dog("마르코", "셰퍼드");
```

수고하셨습니다 :)