Proto and Prototype

Proto: Proto is a property of objects using which inheritance in javascript happens.

```
var a = {
    x: 10,
    calculate: function(z) {
        return this.x + this.y + z;
    }
}

var b = {
    y: 20,
    __proto__: a
}

var c = {
    y: 30,
    __proto__: a
}
```

http://dmitrysoshnikov.com/wp-content/uploads/prototype-chain.png

Prototype: Prototype is an object available in constructor functions (functions) which is used for inheritance also known as prototypical inheritance in javascript.

The __proto__ of every instance created from a constructor function would point to the prototype of the constructor function.

```
function Car(model, year) {
  this.model = model;
  this.year = year;
  this.displayInfo = function() {
    return `This is a ${this.year} ${this.model}.`;
  };
}
```

Better

```
function Car(model, year) {
  this.model = model;
  this.year = year;
}
```

```
Car.prototype.displayInfo = function() {
    return `This is a ${this.year} ${this.model}.`;
};
```

Dog

In the constructor function

Numberoflimbs
Color

I want a property of bark that would be same in all the dogs

```
function Dog(color,numberOfLimbs){
     this.color=color;
     this.numberOfLimbs=numberOfLimbs;
}

Dog.prototype.bark=function(){
 console.log("woof woof");
}
var a=new Dog("white",4);
```

**Object.create: It is used to create object from
another object and get all the properties of the**

creating object in the new object using the proto chain.

```
let carPrototype = {
  displayInfo: function() {
    return `This is a ${this.year} ${this.model}.`;
  }
};

let car1 = Object.create(carPrototype);
```

Interview questions

```
let obj = { name: 'Sample Object' };
obj.toString();
```

```
function Cat(name) {
  this.name = name;
}
const fluffy = new Cat("Fluffy");
```