Every JavaScript object has a prototype. The prototype is also an object.All JavaScript objects inherit their properties and methods from their prototype.

There are two interrelated concepts with prototype in JavaScript:

1.Prototype Property

2. Prototype attribute

First, every JavaScript function has a prototype property (this property is empty by default), and you attach properties and methods on this prototype property when you want to implement inheritance.

The second concept with prototype in JavaScript is the prototype attribute. Think of the prototype attribute as a characteristic of the object; this characteristic tells us the object’s “parent”. In simple terms: An object’s prototype attribute points to the object’s “parent”—the object it inherited its properties from.

Prototype creation

Prototype Attribute of Objects Created with new Object () or Object Literal All objects created with object literals and with the Object constructor inherits from Object.prototype. Therefore, Object.prototype is the prototype attribute (or the prototype object) of all objects created with new Object () or with {}. Object.prototype itself does not inherit any methods or properties from any other object.

Prototype Attribute of Objects Created With a Constructor Function Objects created with the new keyword and any constructor other than the Object () constructor, get their prototype from the constructor function.

These are two important ways the prototype is used in JavaScript,

1. Prototype Property: Prototype-based Inheritance

2. Prototype Attribute: Accessing Properties on Objects

Prototype Property: Prototype-based Inheritance

Prototype is important in JavaScript because JavaScript does not have classical inheritance based on Classes (as most object oriented languages do), and therefore all inheritance in JavaScript is made possible through the prototype property. JavaScript has a prototype-based inheritance mechanism.Inheritance is a programming paradigm where objects (or Classes in some languages) can inherit properties and methods from other objects (or Classes). In JavaScript, you implement inheritance with the prototype property.

Prototype Attribute: Accessing Properties on Objects

Prototype is also important for accessing properties and methods of objects. The prototype attribute (or prototype object) of any object is the “parent” object where the inherited properties were originally defined.

the chain from an object’s prototype to its prototype’s prototype and onwards. And JavaScript uses this prototype chain to look for properties and methods of an object.

If the property does not exist on any of the object’s prototype in its prototype chain, then the property does not exist and *undefined* is returned.