

# Prototype and Objects

- **this** - refers to object
- we can define this keyword in forEach after function
- classes in JS are syntactic sugar of JS constructor functions which follows prototypical inheritance
- constructor functions add all the class properties to object when that object is instantiated
  - all property values are set by constructor function
- constructor function outside class
  - declared using capital letters
  - invoked with **new** keyword
  - will always create a new empty object first
- Every object in JS has prototype object, except Base object
- **Prototypical Inheritance**
  - prototype is a separate object that can be inherited by many constructor methods
  - We can access prototype of object using .prototype property
  - classes use **prototypical** inheritance behind the scenes
  - JS tries to find method and properties on object first, if not able to goes on to searching prototype
- **Class creation flow**
  - when class is instantiated, constructor function is fired
  - construction function creates empty object and add properties to that object
  - construction function do not deal with class methods
  - methods are added to class on **prototype object**
- **Inheritance**
  - ParentConstructor.apply( );
  - Apart from parent properties, we can have our own child properties
  - Object.create({}) - creates new object, taking in object as argument
- TS puts **propertyDescriptor** { **writable** : false } when we use **readonly**