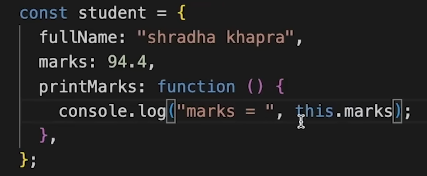
# Classes & Objects

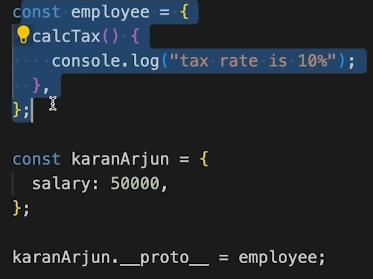
2 ways to create the object 

Prototypes Ťn JS

A javaScrŤpt object Ťs an entŤty havŤng state and behavŤor (propertŤes and method). JS objects have a specŤal property called prototype.

We can set prototype usŤng

**\_ \_ proto \_ \_**



\*If object & prototype have same method, object’s method wŤll be used.

# Classes Ťn JS

Class Ťs a program-code template for creatŤng objects.

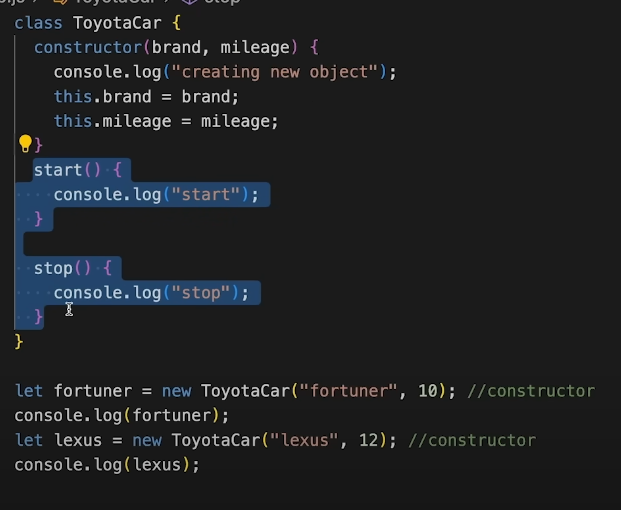
Those objects wŤll have some state (varŤables) & some behavŤour (functŤons) ŤnsŤde Ťt.

**class MyClass { constructor( ) { ... }**

**myMethod( ) { ... }**

**}**

**let myObj = new MyClass( ) ;**



# Classes Ťn JS

Constructor( ) method Ťs : automatŤcally Ťnvoked by new ŤnŤtŤalŤzes object

**class MyClass { constructor( ) { ... }**

**myMethod( ) { ... }**

**}**

# InherŤtance Ťn JS

ŤnherŤtance Ťs passŤng down propertŤes & methods from parent class to chŤld class.

**class Parent {**

**}**

**class Child extends Parent {**

**}**

\*If ChŤld & Parent have same method, chŤld’s method wŤll be used. [Method OverrŤdŤng]

# super Keyword

The super keyword Ťs used to call the constructor of Ťts parent class to access the parent's propertŤes and methods.

**super( args )** // calls Parent‘s constructor

**super.parentMethod( args )**

# Let‘s PractŤce

Qs. You are creatŤng a websŤte for your college. Create a class User wŤth 2 propertŤes, name & emaŤl. It also has a method called vŤewData( ) that allows user to vŤew websŤte data.

Qs. Create a new class called AdmŤn whŤch ŤnherŤts from User. Add a new method called edŤtData to AdmŤn that allows Ťt to edŤt websŤte data.

# Error HandlŤng

try-catch

**try {**

**... normal code**

**} catch ( err ) {** //err is error object

**... handling error**

**}**