Javascriptdə hər şey bir obyektdir. Obyektlərdən eyni kodun təkrarlanmaması və həmçinin team şəklində çalışan zaman işin asanlaşması üçün istifadə olunur. Lakin hər yeni obyekt yaranan zaman belə ona ferqli özəlliklər vermək olar. Deyək ki, bir slider saytda bir neçə yerdə işlənə bilər bunu obyekt halında yazdığımız zaman kod təkrarlanmır sadəcə bu sliderlərin hər birinin özünə məxsus özəllikləri ola bilər, bu zaman əgər fərqli olan özəllik yenilik tələb etməyib yalnız dəyişiklik tələb edərsə həmən slider üçün yalnızca ö dəyişikliklər əlavə olunur. Digər halda əgər bu yenilik tələb edirse o zaman ortaya obyekt yönlü proqramlaşdırmanın miras alma özəlliyi gəlir. Miras almaq oop də nədir? Ilk öncə bu suala cavab axtaraq. Miras mövcud özəllikləri özündə topluyaraq yeni özəlliklərin əlavə olunmasıdır. Deyək ki, bir şəxs ata anadan qan qrupun, dəri rəngi və s. bu kimi kimi özəllikləri miras ala bilir, lakın ad, boy, şəxsi keyfiyyətlər və s. Yalnız özünə məxsus olan özəllikləri isə özü qazanmış olur.

Javascriptdə miras alma özəlliyin prototype ilə edirlər. Js-in özünə tam obyek yönlü proqramlaşdırma demək olmaz. Digər oop yönlü proqramlaşdırma dillərdən fərqlənir. Javasiprtdə bir obyektin daxilində tapılmıyan bir dəyəri elə etmək olar ki digər obyektdə axtaraş Bunun üçün \_\_proto\_\_ dan istifadə olunur. Əgər bir obtyekt \_\_proto\_\_  a sahibdirsə onda əgər axtarılan obyektdə yoxdursa o zaman axtarış \_\_proto\_\_  obyektinin içində davam edir.

Kod ilə buna misal gətirək

var animal = {

eats: true

};

var rabbit = {

jumps: true

};

rabbit.\_\_proto\_\_ = animal;

alert( rabbit.jumps ); // true

alert( rabbit.eats ); // true

1ci alert zamanı ekrana jumps obyekt rabbit çıxardır.

2ci halda isə alert ekrana rabbit.eats çıxartmaq istəyir və onu rabbit obyektinin içində axtarmağa başlayır, lakin tapmir, yenə də axtarışa davam edir yəni rabbit.\_\_proto\_\_, içərisində axtarır o da animal deməkdir.

Hansı obyekt ki onda \_\_proto\_\_ linki göstərilib o prototype adlanır. Yuxarıda olan misaldan bu nəticəyə gəlmək olarki animal rabbitin prototype-dır. Həmçinin demək olar ki, rabbit animaldan miras alıb

Obyekt hansı ki \_proto\_\_ a sahibdir onun da oz \_\_proto\_\_ ola bilər və bu ardıcıllıqla gedir. Lakin axtarış zamanı hamsı bir biri ilə zəncirvari bağlıdır. Bir neçə prototype-ı bir obyektə birləşdirmək olmaz lakin obyektləri zəncirvari yığmaq olar o haldaki bir obyekt digər obyektə \_\_proto\_\_ köməyi ilə yönlənirsə.

Müasir browserlərdə prototype ilə işləmək üçün metodlar mövcuddur

* [Object.getPrototypeOf(obj)](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Object/getPrototypeOf) ( IE8-dən başqa)
* [Object.setPrototypeOf(obj, proto)](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Object/setPrototypeOf) (IE10- dən başqa)
* [Object.create(proto, descriptors)](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Object/create) (IE8- dən başqa)