

OOPS-3 : PROTOTYPE & PROTOYAL INHERITANCE

Private Properties :

- Ensure that Class data is encapsulated & not directly accessed from outside the class.
- Improves security & data integrity.

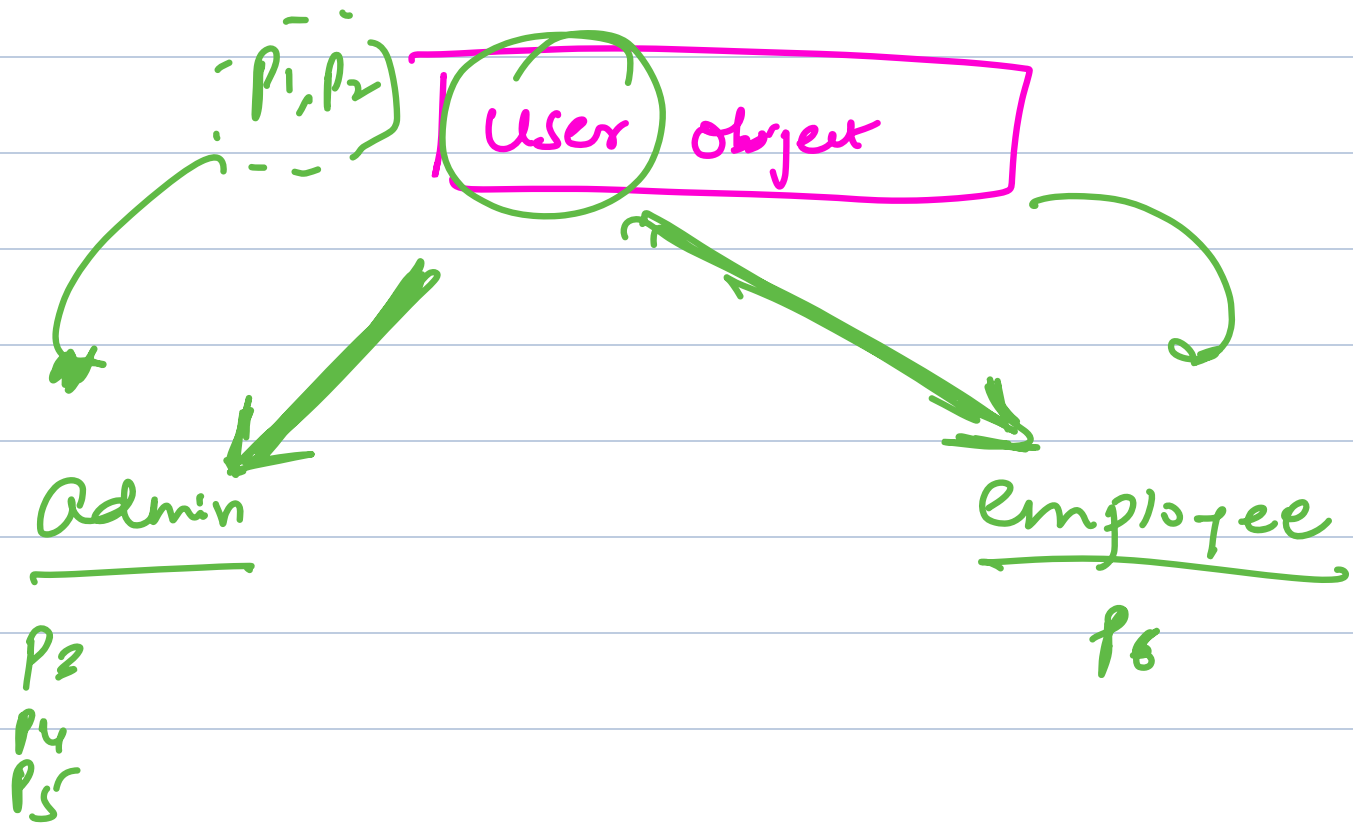
Used as ~~##~~ name

(# prefix to any property makes it private)

AGENDA :

- Prototype Object
- Prototypical Inheritance
- Proto object
- Prototype Chaining
- Object.create()

Prototype \rightarrow first model / of something.
init & design



Arr. join (,) // From where is it
Obj. toString() // coming.

What is prototype

- Prototype is an object that has methods and properties that gets attached to the object.
- Prototype are mechanism by which JS objects **INHERITS** feature from one another.

JS have special hidden property
[[Prototype]].

which is either null or referencing to other object.

This object is called as 'prototype'.

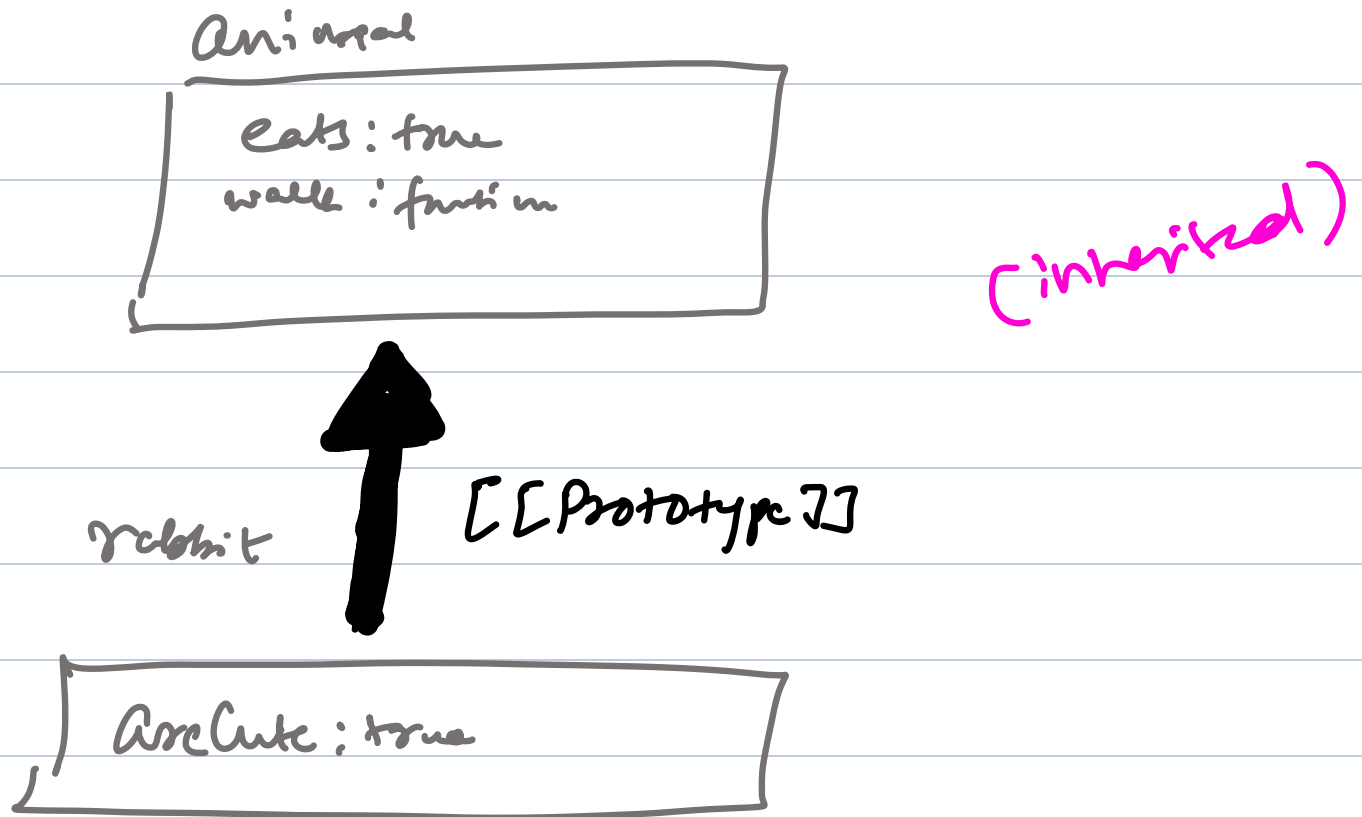
ANALOGY

- 1) Complimentary Dish at restaurant
- 2) Mom's Snacks
- 3) Family is given, friends are Chosen

[[Prototype]] ← internal & hidden;

but we can change it. '__proto__'

eg.



* initially
JS searches for the property in current object,
then it looks/goes to its prototype &
checks, then again its prototype & then
again: - - - - - until 'null' is reached.

Be it array, string, objects,

all have 'Object prototype'.

Traverse an object

- `Object.keys(obj)` ; returns own property only.
- `for...in` loop to traverse objects.
returns own + inherited property.

- `object.hasOwnProperty(key)`
Used to differentiate b/w own & inherited property.

eg. `rabbit.hasOwnProperty(eats Meat)`
// false

`rabbit.hasOwnProperty(Can Jump)` // true.

Adding methods on prototype →

// refer adding-methods.js in folder 27.

```
BeetleUser.prototype.msg = function() {  
    -  
    -  
    -  
}
```

Primitives (String, Boolean, Number)

let s = 'Hello';

```
s.toLowerCase()  
s.toFixed()  
:  
:  
:
```

How?? Primitives also
have prototypes
defined!
String.prototype
Number.prototype
Boolean.prototype

Object.create ()

^{one more} Way to achieve prototypal inheritance.

- Allows us to create new object with a specified prototype object & properties.

(Refer object-create.js) For Code.

~~st~~

function Animal () {

}

Animal.prototype.constructor → ?? Function
Animal.

- prototype only exists on functions designed to act as constructors.

• Can `fn` has property of 'prototype', which includes constructor property.

• `CONSTRUCTOR` property points to function itself'