SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

Object Constructor & Prototyping

Course: 19CAT901 – Web Programming Essentials

Unit IV : Advanced Java Script

I Semester / I MCA

ES6 (ECMA Script 6)

- Major revision to JavaScript, released in 2017. The features are

let keyword	**const** keyword

Arrow functions has a short syntax for writing function
`var x = function(x, y)`
`{   return x * y;  }`
`const x = (x, y) => x * y;`
	for/of statement loops through the values of an iterable objects
	`for (variable of object)`
	`{  // code}`

JavaScript Classes are templates for JavaScript Objects
`class ClassName {`
`  constructor() { ... }`
`}`
	function parameters to have default values like
	`function myFunction(x, y = 10)`

Symbol is a primitive datatype represents a unique "hidden" identifier that no other code can accidentally access	

- □ Everything in JS is an object like string, date, math, array, functions…
- □ Objects are variables too, but can hold many values

```
Var name=“ram”;
var person = {firstName:”Santhosh", lastName:”Menon",
age:41, eyeColor:"black"}
```

- □ A JavaScript object is a collection of **named values,** called **properties**

Object Methods

- Methods are actions that can be performed on objects
- object method is an object property containing a function definition
- Objects can be created by literals or using Object method
- Objects are mutable
- Object can be created using new Object()

Objects are containers for named values, called properties and methods

Object Properties

- ❑ Properties are the values associated with a JavaScript object. It can be changed, deleted or added
- ❑ Property of an object can be accessed by objecname.property or ojectname[property]
- ❑ Use for..in loop to access properties
- ❑ Add properties even after object creation
- ❑ **delete** keyword deletes both the value of the property and the property itself. Example *delete person.lname*
- ❑ *It can be displayed by its name or Object.values(objname)*

Object Accessors

- ❑ Getters and setters allow you to define Object Accessors

```
var person = {
 firstName: “Raj",
 lastName : “Kumar",
 language : “Tamil",
 set lang(lang) {
   this.language = lang;
 }
};
person.lang = “Telugu"; // Set an object property using a setter
document.getElementById("demo").innerHTML =  person.language;     // Display data from the object:
```

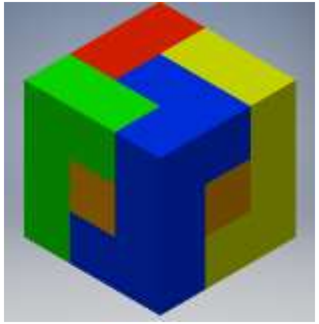
Object constructor

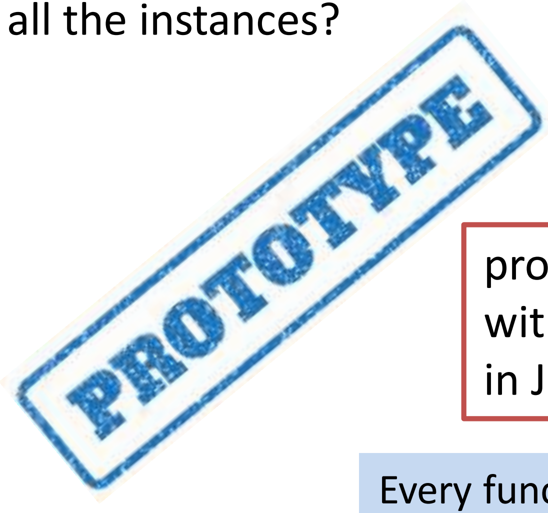
- ❑ Object constructor is merely a regular JavaScript function
- ❑ It is called via the new operator
- ❑ Properties and methods inside function by prefixing the keyword "this“
- ❑ constructors create the blueprints for objects, not the object itself

```
<script>
Function student(regno, name, course) {
   this.name = name;
   this.regno = regno;
   this.course = course;
   this.display = function() {
       alert( this.regno+” – “ +this.name + “-"
+this.course)
   }
}
 s1 = new  student(“20CA001”, “Reghu“, “MCA”);
S1.display();
//-->
</script>
```

What is what?

- JavaScript is a dynamic language which permits you to attach new properties to an object at any time
- Suppose if we want to add new properties at later stage to a function which will be shared across all the instances?





prototype is an object that is associated with every functions and objects by default in JavaScript

Every function includes prototype object by default

Add methods using Prototype(Inheritance)

- ❑ Prototype is a type of inheritance in JavaScript
- ❑ We would like an object to inherit a method after it has been defined
- ❑ Prototype is "attaching" a method to an object after it's been defined, in which all object instances then instantly share
- ❑ Use the keyword "prototype" immediately following the object's name to utilize this functionality

```
Objectname.prototype.functionname=function(name)
    { …}
```

Implementation

```
function Student()
{
   this.name = 'John';
   this.gender = 'Male';
}
var stud1 = new Student();
stud1.age = 15;
alert(stud1.age);  // 15
 var stud2 = new Student();
alert(stud2.age);    //undefined
```

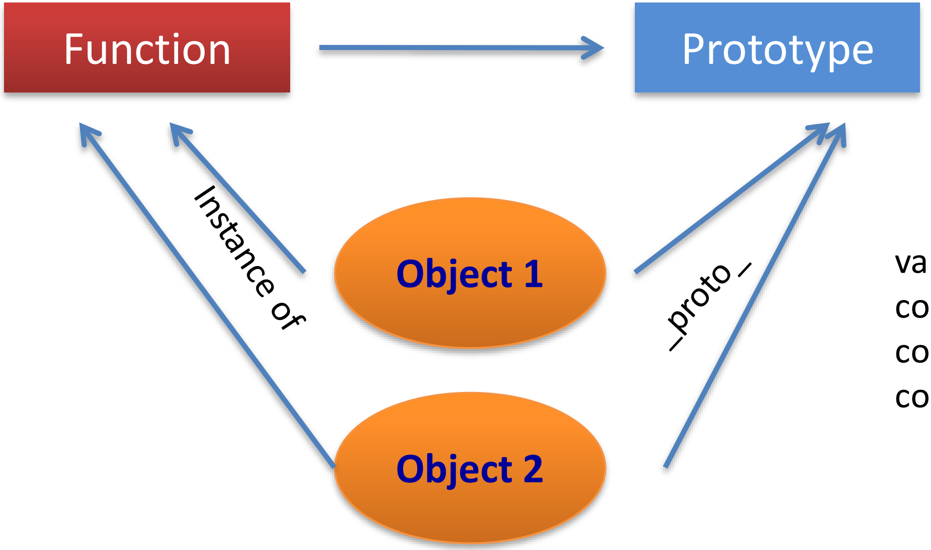
Program without prototype

PROTOTYPE

```
function Student()
{
    this.name = 'John';
   this.gender = 'Male';
}
Student.prototype.age  = 15;
var stud1 = new Student();
alert(stud1.age);         //  15
 var stud2 = new Student();
alert(stud2.age);    // 15
```

Implementation

Every object which is created with the new keyword includes **__proto__** property that points to prototype object of a function that created this object

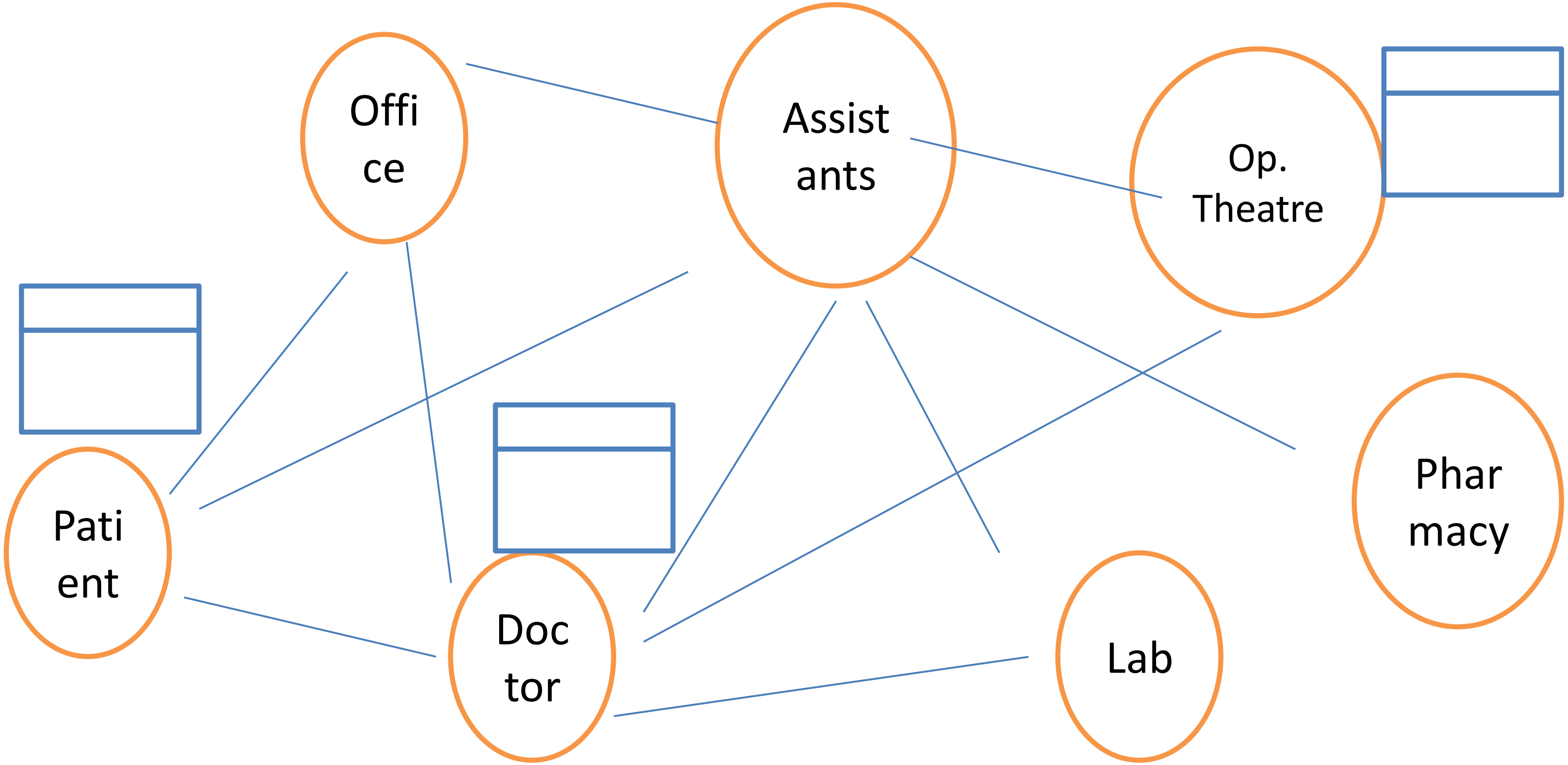


```
var stud1 = new Student();
console.log(student.prototype);
console.log(stud1._protot_);
console.log(typeof Student.prototype);
```

References

- ❑ Thomas A. Powell, “HTML & CSS: The Complete Reference”, Fifth Edition, 2010
- ❑ https://www.w3schools.com/js/js_object_constructors.asp
- ❑ https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/constructor
- ❑ https://css-tricks.com/understanding-javascript-constructors/

Problem Space


Offi
ce
Assist
ants
Op.
Theatre
Pati
ent
Doc
tor
Lab
Phar
macy