WEB TECHNOLOGIES

JavaScript Objects

Vinay Joshi
Department of
Computer Science and
Engineering

Javascript Objects

Basic Object Oriented Concepts

- An object in JavaScript is a reference data type.
- An object can be compared to any real world entities.
- An object is an unordered list of properties consisting of a name (always a string) and a value. When the value of a property is a function, it is called a method.

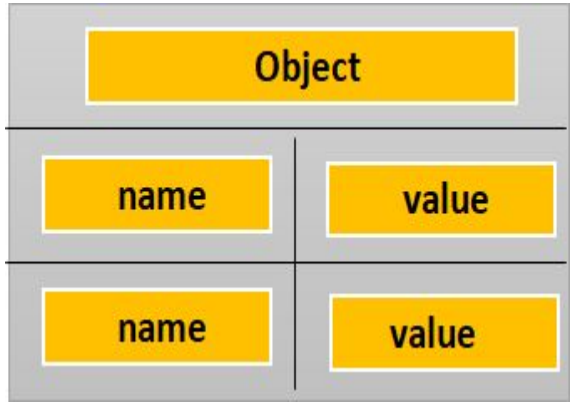

Fig 1:Object Structure

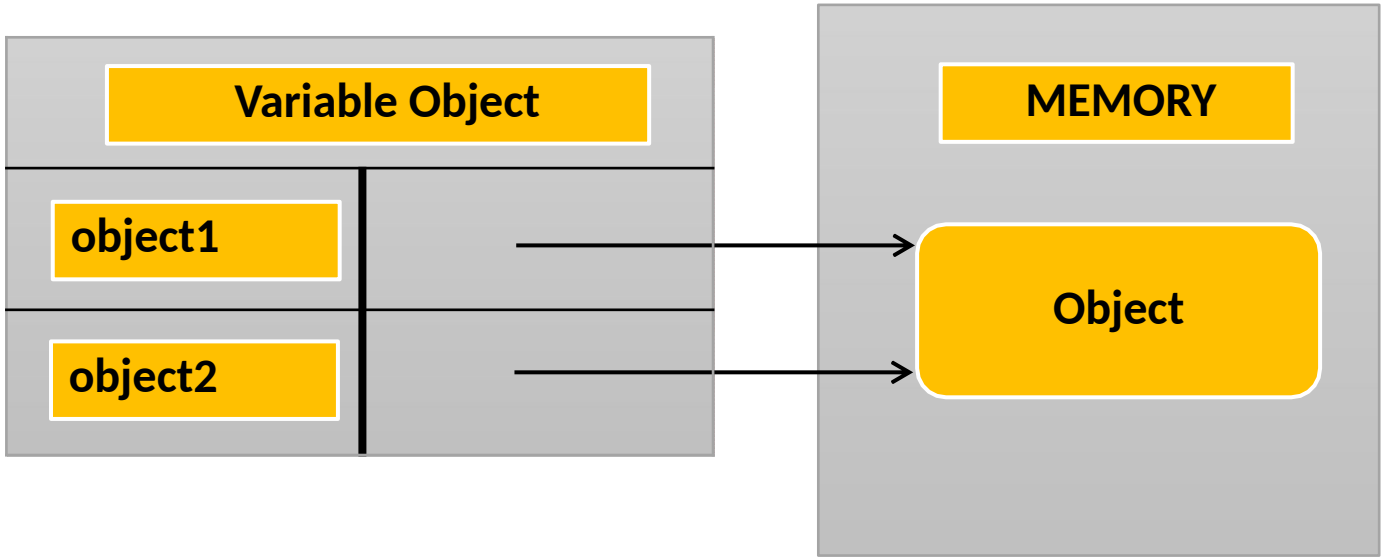

Fig 2:Object Reference

Var object1= new Object(); Var object2 = object1;

Javascript Objects

Object Literals using {key : value}

```
var emp2 = {
    "firstName": "Aruna",
    "lastName": "Srinivasan",
    "showName": function () {
        alert(this.firstName + " " + this.lastName);
    }
};
emp2.showName();
```

- A variant of object literal syntax
- Specifies object members and their values inside the curly brackets as key-value pairs
- A member and its value are delimited using colon (:) character.

Javascript Objects

Creating an object using function Constructor

```
function Employee(fname,lname) {
    this.firstName = fname;
    this.lastName = lname;
    this.showName = function () {
        alert(this.firstName + " " + this.lastName);
    };
}
var emp5 = new Employee("Aruna", "Srinivasan");
emp5.showName();
```

- JavaScript functions are objects themselves. It can be called as a Constructor Function.
- Once the function constructor is created you create object of that function using new keyword.
- The function name uses camel case convention

Javascript Objects

Creating an object using anonymous function

```
var emp6 = new function () {
    this.firstName = "Aruna";
    this.lastName = "Srinivasan";
    this.showName = function () {
        alert(this.firstName + " " + this.lastName);
    };
}
emp6.showName();
```

- Two steps are performed here
  - Firstly, it creates an anonymous constructor function
  - Secondly, it calls new on it to create its object.

Javascript Objects

Prototype property

- Prototype property of an object holds the structure of that object
- It is shared by all object instances created using that constructor
- This can be used to modify/add properties to all instances after they have been created
- Methods should be added to prototype since only one copy of the method is created. If methods are added to constructor, each object instance holds its own copy of the method.

```
Employee.prototype.showName = function(){…}
```

Javascript Objects
Prototype property

```
function Car(make, model, year, owner) {
  this.make = make;
  this.model = model;
  this.year = year;
  this.owner = owner;
}
```

To instantiate the new objects, you then use the following:

```
var car1 = new Car('Eagle', 'Talon TSi', 1993, rand);
var car2 = new Car('Nissan', '300ZX', 1992, ken);
```

Javascript Objects
Prototype property

Note that you can always add a property to a previously defined object. For example, the statement

```
car1.color = 'black';
```

JavaScript `prototype` property allows you to add new properties to object constructors:

```
Car.prototype.color = null;
car1.color = 'black';
```

Javascript Objects
Prototype property

```
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}

Person.prototype.name = function() {
  document.write(this.firstName + " " + this.lastName);
};

var p1 = new Person("John", "Doe", 50, "blue");
p1.name();
```

Javascript Objects

Creating an object using EcmaScript 6 Class Keyword

```
class Employee {
    constructor(fname, lname)
      {  this.firstName = fname;
        this.lastName = lname;   }
        showName()
        {   alert(this.firstName + " " + this.lastName); }
}
var emp7 = new Employee("Aruna", "Srinivasan");
emp7.showName();
emp7.firstName = "Sanvi";
emp7.showName();
```

- Class is defined .
- Then the objects of the class are created.
- It internally uses the constructor/prototype based approach of creating objects

THANK YOU

Vinay Joshi

Department of Computer Science and Engineering

vinayj@pes.edu