

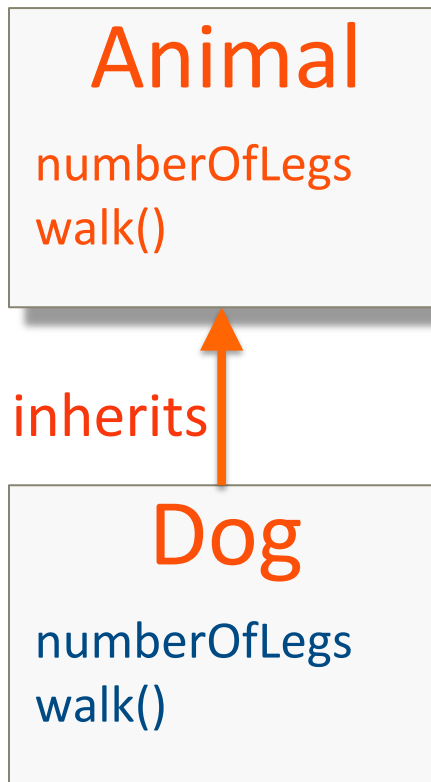
Controller As Syntax



Prototypal Inheritance



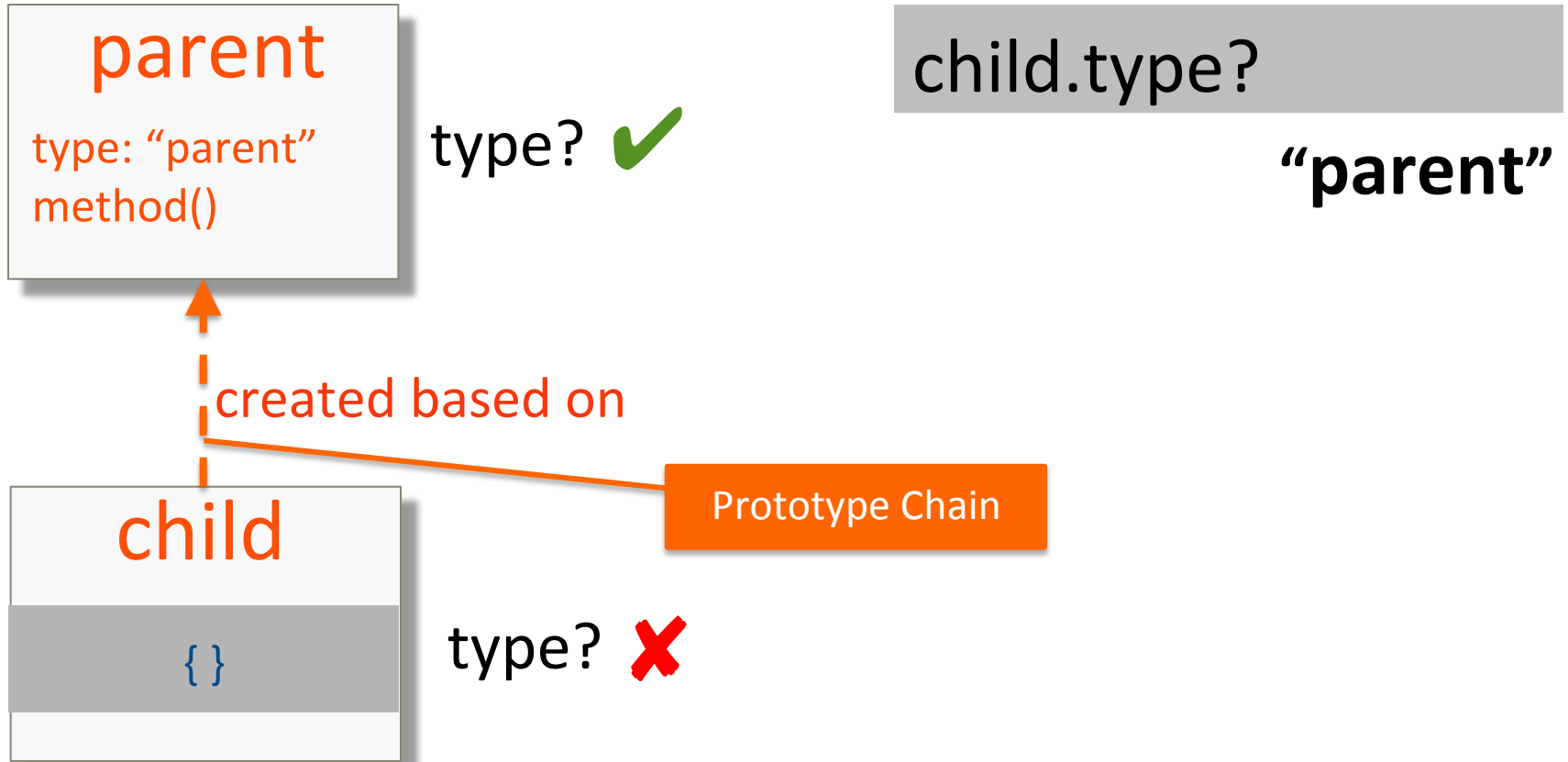
What is Inheritance?



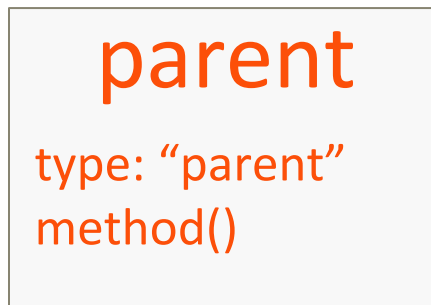
Inheritance is when an object or a class is based on another object or a class (parent), using the same implementation and/or the same values.

Technique for code reuse as well as logical entity structure.

JS Prototypical Inheritance



Prototypical Inheritance



```
child.type = "child";  
child.type?
```

"child"

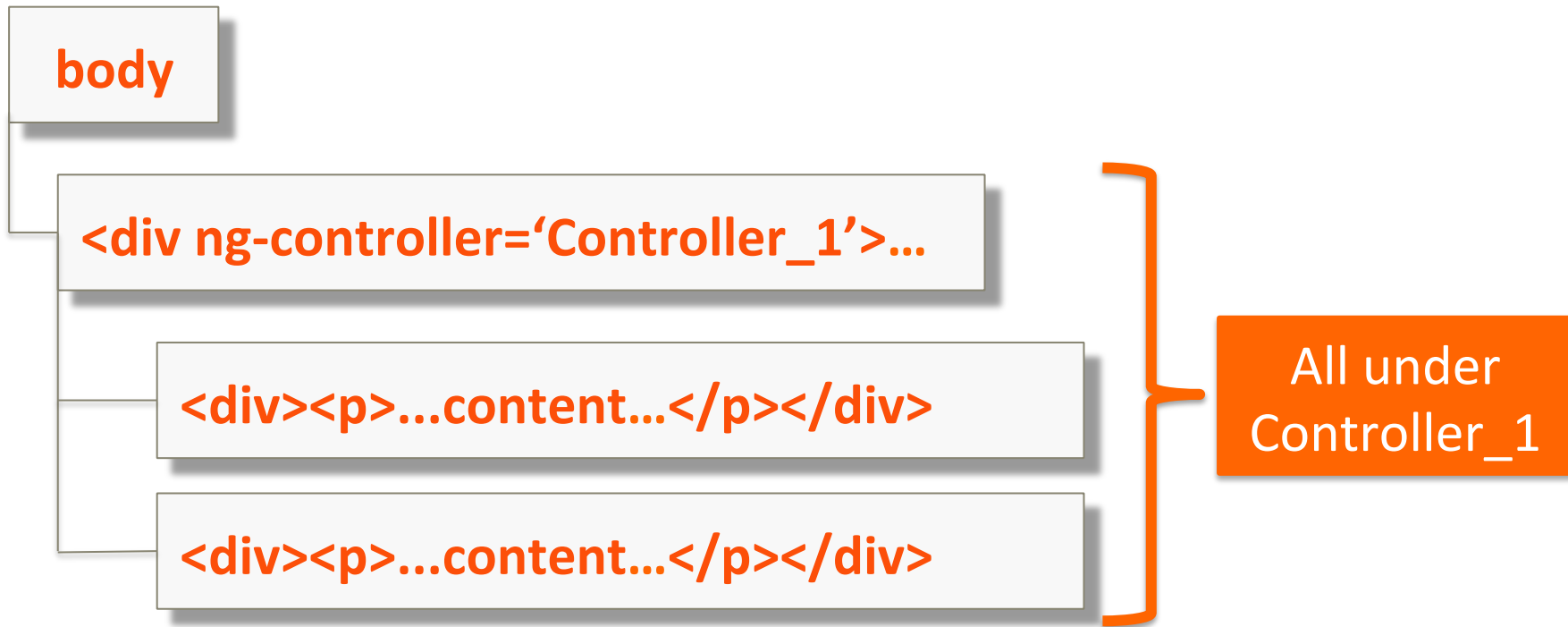
child.type property
masks type inherited
from the prototype

type? ✓

Scope Inheritance



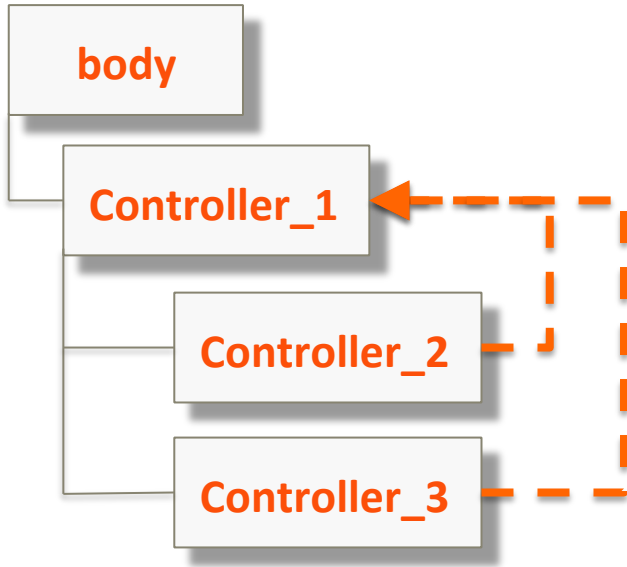
Scope Inheritance



Scope Inheritance



Scope Inheritance



`$scope1.prop = "ctrl-1"`

`$scope2.prop?`

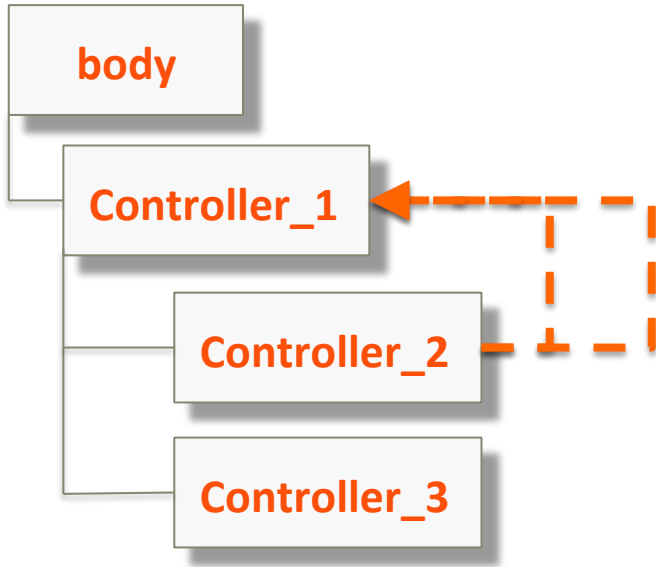
`$scope3.prop?`

"ctrl-1"

`$scope2.prop = "ctrl-2"`

`$scope1.prop?` **"ctrl-1"**

Scope Inheritance



```
$scope1.obj.prop = "ctrl-1"
```

```
$scope2.obj.prop?
```

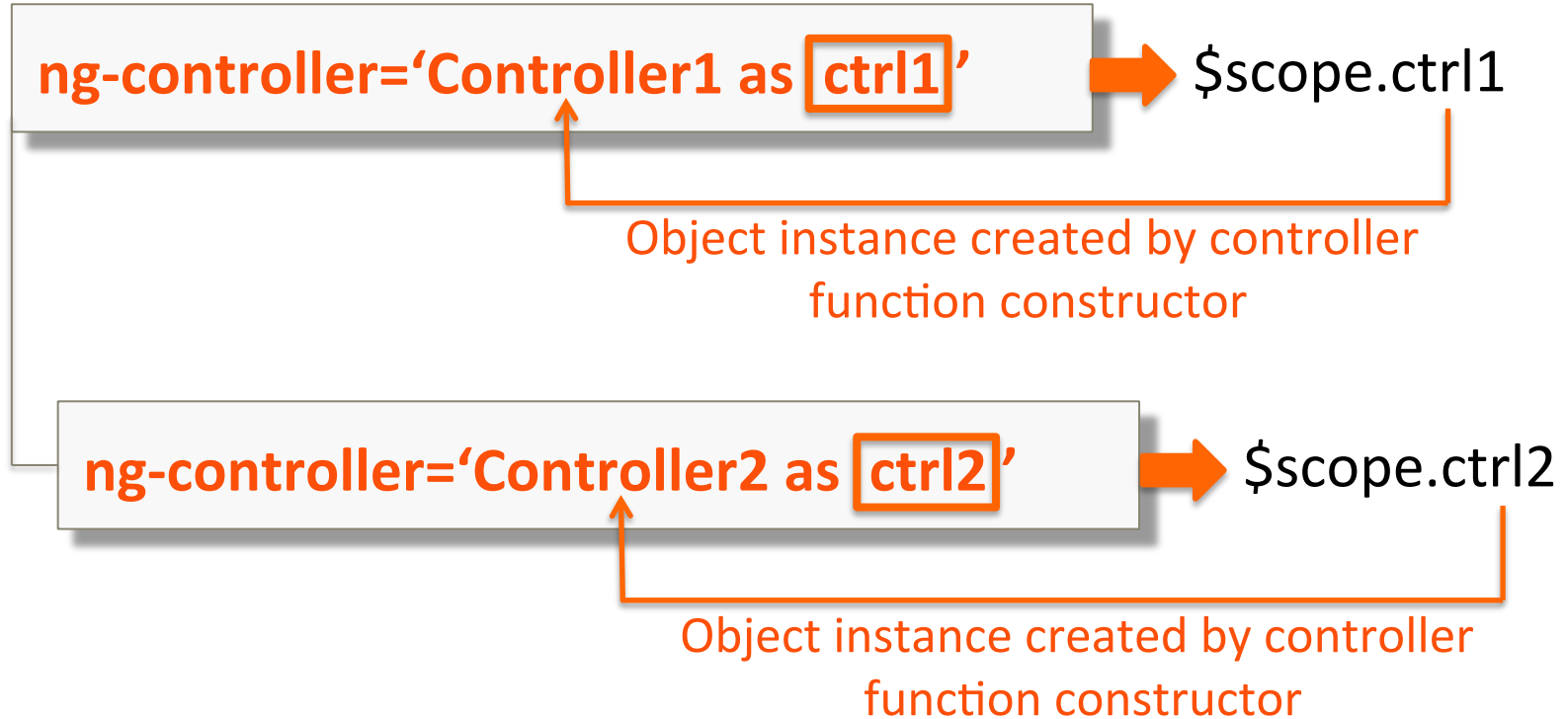
"ctrl-1"

```
$scope2.obj.prop = "ctrl-2"
```

```
$scope1.obj.prop?
```

"ctrl-2"

Controller As Syntax



Controller As Syntax

```
angular.module('...')  
  .controller('Controller1', Controller1);  
  
function Controller1 () {  
  this.myProp = "some value";  
}
```



Controller As – Behind the Scenes (Roughly)

ng-controller='Controller1 as ctrl1 '

ng-controller='Controller2 as ctrl2 '

...

\$scope1.ctrl1 = new Controller1(...)

\$scope2 = Object.create(\$scope1)

\$scope2.ctrl2 = new Controller2(...)

...

Controller As Syntax in HTML

ng-controller='Controller1 as ctrl1 '

ng-controller='Controller2 as ctrl2 '

```
<div>
  {{ ctrl2.myProp }}
  ...
  {{ ctrl1.myProp }}
</div>
```

Summary

- ✧ What we learned as part of this lecture applies to MUCH MORE than just the Controller As syntax
- ✧ Inheritance in general is used for code re-use (and sometimes to express a relationship between entities)
- ✧ Prototypal inheritance in JS is based on object instances rather than classes
 - Property is inherited from parent, looked up through Prototype Chain
 - Property is local if it's declared on the child with the same name as the parent and therefore masking the parent's property



Summary

- ✧ \$scope is based on prototypal inheritance
 - Child controller's \$scope inherits from parent controller's \$scope
- ✧ Controller As syntax is `ControllerName as label`
- ✧ Angular creates property 'label' on the \$scope
 - The label is a reference to 'this', i.e., instance of Controller
 - Works because .controller treats it as a function constructor
- ✧ Attach properties to 'this' inside of Controller, not \$scope
 - Easier syntax in HTML and JS – no masking occurs

