

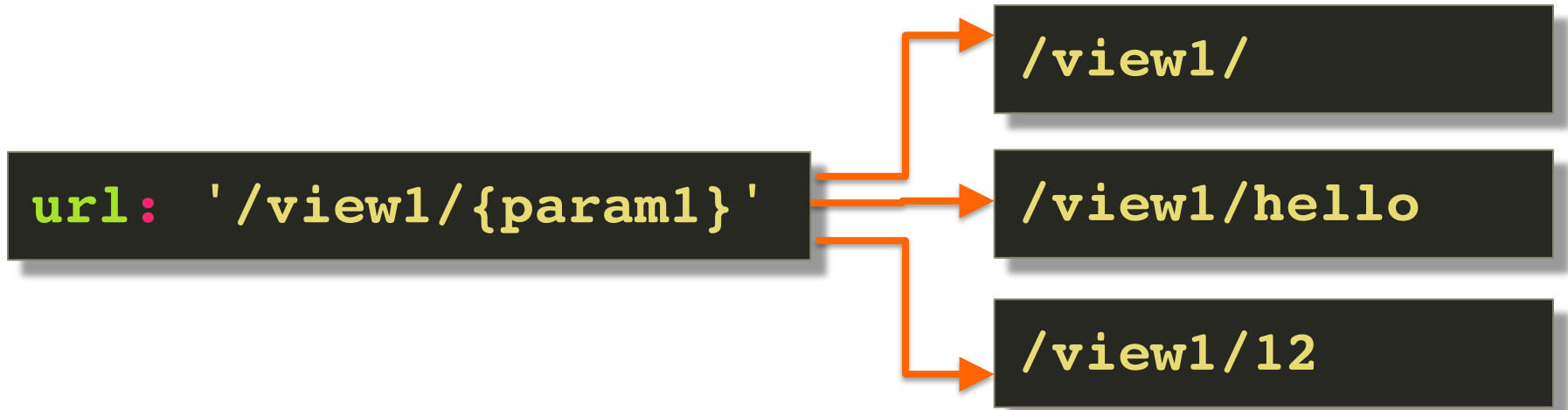
# **Route State With URL Parameters**



## Step 1: Set up URL Property With Param(s)

```
.state('view1', {  
  url: '/view1/{param1}',  
  templateUrl: 'view1.html',  
  controller: 'View1Ctrl as view1',  
  resolve: {  
    myData: ['$stateParams',  
             function ($stateParams) {  
               return getDataBasedOn($stateParams.param1);  
             }]  
  }  
});
```

# URL Matches



## Step 1: Set up URL Property With Param(s)

```
.state('view1', {  
  url: '/view1/{param1}',  
  templateUrl: 'view1.html',  
  controller: 'View1Ctrl as view1',  
  resolve: {  
    myData: ['$stateParams',  
             function ($stateParams) {  
               return getDataBasedOn($stateParams.param1);  
             }]  
  }  
});
```



## Step 2: Inject Resolve Property Into Controller

```
View1Ctrl.$inject = ['myData'];  
function View1Ctrl(myData) {  
    var view1 = this;  
    view1.myData = myData;  
}
```

Injected value



## Step 2: Inject Resolve Property Into Controller

State name

Param name/value pairs

```
<a ui-sref="view1({itemId:someVal})">
```

Link to view with data

```
</a>
```

# Summary

- ✧ State's url property can be declared with parameters
- ✧ Parameters:
  - Wrapped in curly braces `{paramName}`
  - Can have more complex matching rules other than just a string
  - Support regular expression matching
- ✧ Use `$stateParams` service to retrieve parameters
  - `$stateParams.paramName`
- ✧ Construct a URL with `ui-sref` directive:
  - `ui-sref="stateName({paramName: value})"`

