

LEARN. DO. EARN

ACADGILD

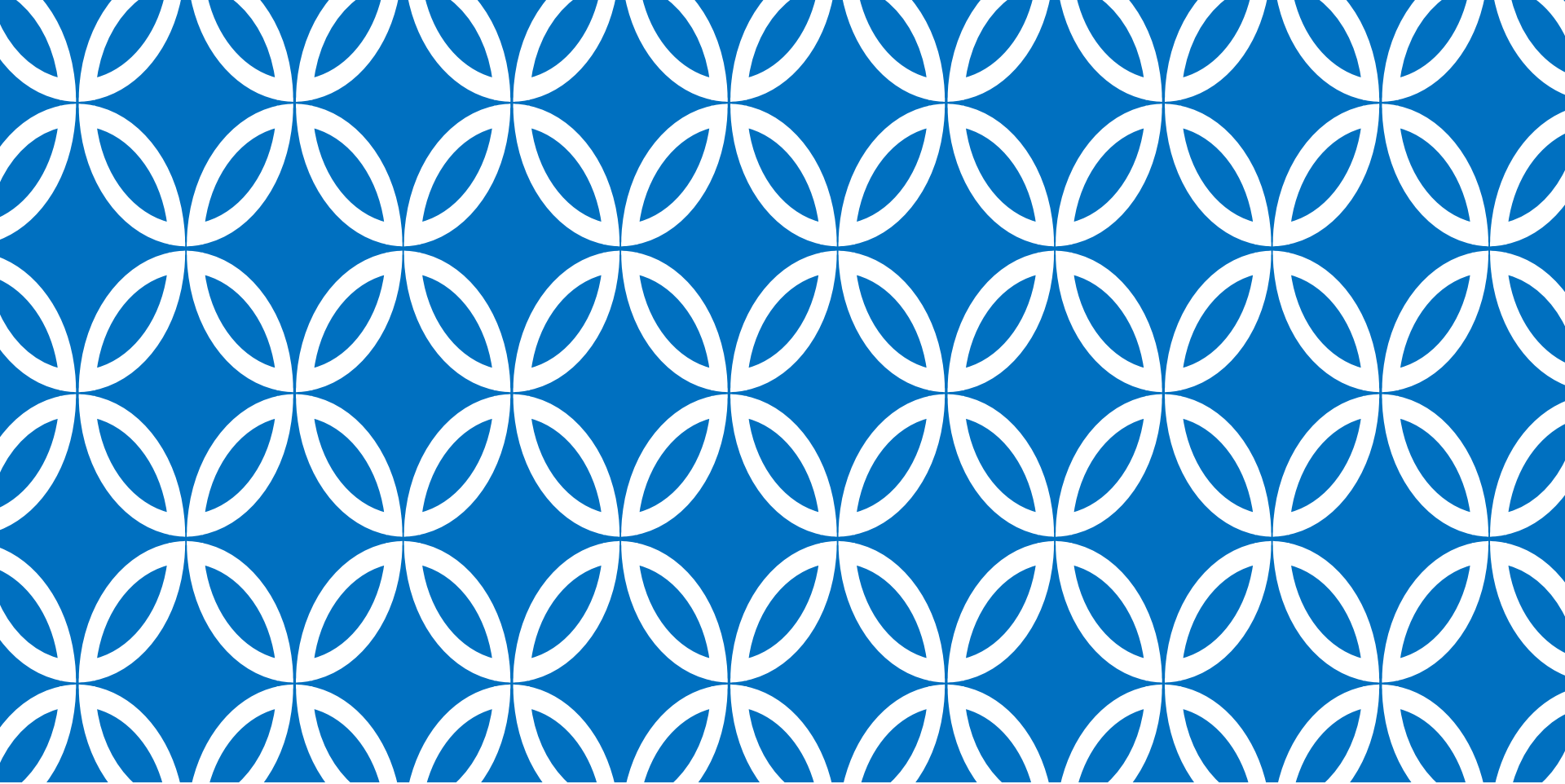


FRONT END DEVELOPMENT (WITH ANGULARJS)



Website : <http://www.acadgild.com>
LinkedIn : <https://www.linkedin.com/company/acadgild>
Facebook : <https://www.facebook.com/acadgild>

© copyright ACADGILD



Session 13 – AngularJS Filters





Agenda – AngularJS Filters

1. What Is Angular Filter?
2. Using Filters In JavaScript
3. Passing Arguments To Filters Using Currency Filter
4. Using Number Filter
5. Using Date Filter
6. Using JSON Filter
7. Limit Filter
8. limitTo Filter
9. orderBy Filter
10. Using filter Filter
11. AngularJs Custom Filter





What Is Angular Filter?

- AngularJS Filters can be used to transform data.
- Reusable operation using which you can modify the content that is shown on the page.
- **Examples:** uppercase a value, filter search results, etc.





Using Filters In JavaScript

- Filters can also be used within the JavaScript code.
- The format is **`$filter(<filter name>')(<expression>)`**.





Passing Arguments To Filters Using Currency Filter

- For arguments, we can pass them with a colon after the filter name and for multiple arguments, we append a colon after each argument.
- Lets understand this using a currency filter This filter Formats a number as a currency.

Syntax:

```
{{ currency_expression | currency : symbol : fractionSize }}
```





Using Number Filter

- The **number filter** is used to filter numbers.
- It formats a number as text.
- If input is not a number an empty string is returned.
- If input is an infinite then the Infinity symbol ' ∞ ' is returned.

Syntax:

```
{{ number_expression | number : fractionSize }}
```





Using Date Filter

- The date filter formats date to a string based on the requested format.
- **Template format:** `{{ date_expression | date : format : timezone }}`

Note that data can be either an ISO date object or date string or timestamp.





Using JSON Filter

- The JSON filter allows you to convert a JavaScript object into JSON string.
- It is mostly useful for debugging.
- **Example:** `{{ json_expression | json : spacing }}`
- **Note** that *spacing option* is the number of spaces to use per indentation and defaults to 2.





Limit Filter

- **Limit Filter** limits the number of documents to be executed.

- **Example:**

```
{  
  "filtered" : {  
    "filter" : {  
      "limit" : {"value" : 100}  
    },  
    "query" : {  
      "term" : { "name.first" : "shay" }  
    }  
  }  
}
```





limitTo Filter

- This filter creates a new array or string containing only a specified number of elements as mentioned in the **'limitTo'** filter.
- The elements are taken from either the beginning or the end of the source array, string or number, as specified by the value and sign (positive or negative) of limit.
- If a number is used as input, it is converted to a string.

Syntax:

```
{{ limitTo_expression | limitTo : limit : begin }}
```

or

```
$filter('limitTo')(input, limit, begin)
```





orderBy Filter

- **orderBy** filter orders a specified array by the expression predicate.
- The result is ordered alphabetically for strings and numerically for numbers.
- To order numbers properly ensure that numbers are actually being saved as numbers and not strings.

Syntax:

```
{{ orderBy_expression | orderBy : expression : reverse }}
```





Using filter Filter

- This is a very important filter.
- It selects a subset of items from array and returns it as a new array.

Syntax :

```
{{ filter_expression | filter : expression : comparator }}
```

- Here filter _expression is the expression. Expression parameter represents the reference data to which we have to filter and the comparator is the comparison function or value to decide if the value can be considered a match.





AngularJs Custom Filter

- **Defining a custom Filter**

```
module.filter("filter_name", function(data) {  
    // data is filtered or modified  
    // in a specific manner and returned  
    return data;  
})
```

- **Using the defined filter**

```
<div  
    data-ng-repeat="record in records | filter: filter_name">  
</div>
```





Lets Discuss Assignments