**Description**

The ng-repeat directive allows us to iterate over a collection and create a DOM element (with the corresponding eventual subtree) for each item in the collection. Every time the collection changes, the DOM is updated as well. We can iterate over an array or over the properties of an object.

In point 1 of the example we iterate over a people array that we have initialized in the script file and added to the $scope of mainController. Each item in the array is an object with a firstName and a lastName property. For each iteration, the object is assigned to the person variable and a new tr element is created with the corresponding subtree (the td elements).

In point 2 we see how a change in the model is reflected in the DOM tree generated by the ng-repeat directive. We have a checkbox for each row that can be selected/deselected and is bound to the isSelected property of each item in the model array. We also have a checkbox in the table header that allows to select/deselect all the rows by setting the isSelected property of each row accordingly.

In point 3 we generate the items of a list and we also use some special variables that are populated by the ng-repeat directive. Here are the available special variables:

* $index: the zero-based index of the current item in the collection
* $first: true if the current item is the first one in the collection, false otherwise
* $middle: true if the current item is in the middle of the collection (after the first one, but before the last), false otherwise
* $last: true if the current item is the last one in the collection, false otherwise
* $even: true if the current $index is even, false otherwise
* $odd: true if the current $index is odd, false otherwise

Note that in the example we set the number in front of the list item by incrementing by 1 the $index, so the numbers start from one, but the $even and $odd variables consider the $index as is, so the first item is even ($index is 0), the second one is odd ($index is 1) and so on.

In point 4 we see how we can enumerate the properties of an object. We have two nested ng-repeat directives. The first one iterates over the array items, while the second one iterates over the properties of each item getting their name and value. As you can see, the properties are sorted in alphabetical order and this behavior cannot be changed. An interesting test you can do in the example, is observe how the values of the isSelected properties in this list change whenever you select/deselect a row in point 2 because both the points of the example are using the same selectablePeoplearray.

In point 5 we can try to dynamically add and remove items from an array and see that the collection rendering is updated accordingly whenever the monitored model array changes. Here you can see that we must use the $parent scope for the removebutton because the ng-repeat directive creates a new scope for each tr it creates and the removeStringFromArray function is defined on the $scope of mainController.

The ng-repeat directive creates a new scope for each DOM subtree that it creates.