# ASSIGNMENT8

**AIM:**

Implement any five built in helper functions in angular JS.

**THEORY:**

In AngularJS, helper functions are utility functions provided by the AngularJS framework to facilitate common tasks and operations within applications. These functions are often used for data manipulation, type checking, and other common programming tasks. Here's a brief overview of some of the key helper functions provided by Angular JS:

angular.isArray():This function checks whether the given value is an array or not. It returns true if the value is an array, otherwise false.

angular.isObject():This function checks whether the given value is an object or not. It returns true if the value is an object, otherwise false.

angular.isString(): This function checks whether the given value is a string or not. It returns true if the value is a string, otherwise false.

angular.isNumber():This function checks whether the given value is a number or not. It returns true if the value is a number, otherwise false.

angular.isFunction():This function checks whether the given value is a function or not. It returns true if the value is a function, otherwise false.

These helper functions are commonly used in AngularJS applications to perform type checks and conditional operations based on the types of data being processed. They help developers write cleaner and more concise code by providing convenient ways to handle different types of data. Additionally, these functions can improve code readability and maintain ability by abstracting away low-level type-checking logic.

1. isArray()

Purpose: Checks if a given value is an array.

Usage: angular.isArray(value)

ReturnValue:Returns true if the value is an array, otherwise false.

Example:

```
angular.isArray([1,2,3]);//true

angular.isArray('hello');//false
```

2. isObject()

Purpose: Checks if a given value is an object.

Usage: angular.isObject(value)

Return Value:Returns true if the value is an object, otherwise false.

Example:

```
angular.isObject({});// true

angular.isObject(null);//false
```

3. isString()

Purpose: Checks if a given value is a string.

Usage:  angular.isString(value)

Return Value: Returns true if the value is a string, otherwise false.

Example:

```
angular.isString('hello');//true

angular.isString(123);       //false
```

4. isNumber()

Purpose: Checks if a given value is a number.

Usage: angular.isNumber(value)

Return Value: Returns true if the value is a number, otherwise false.

Example:

```
angular.isNumber(123);       // true

angular.isNumber('123');     //false
```

5. isFunction()

Purpose: Checks if a given value is a function.

Usage: angular.isFunction(value)

Return Value: Returns true if the value is a function, otherwise false.

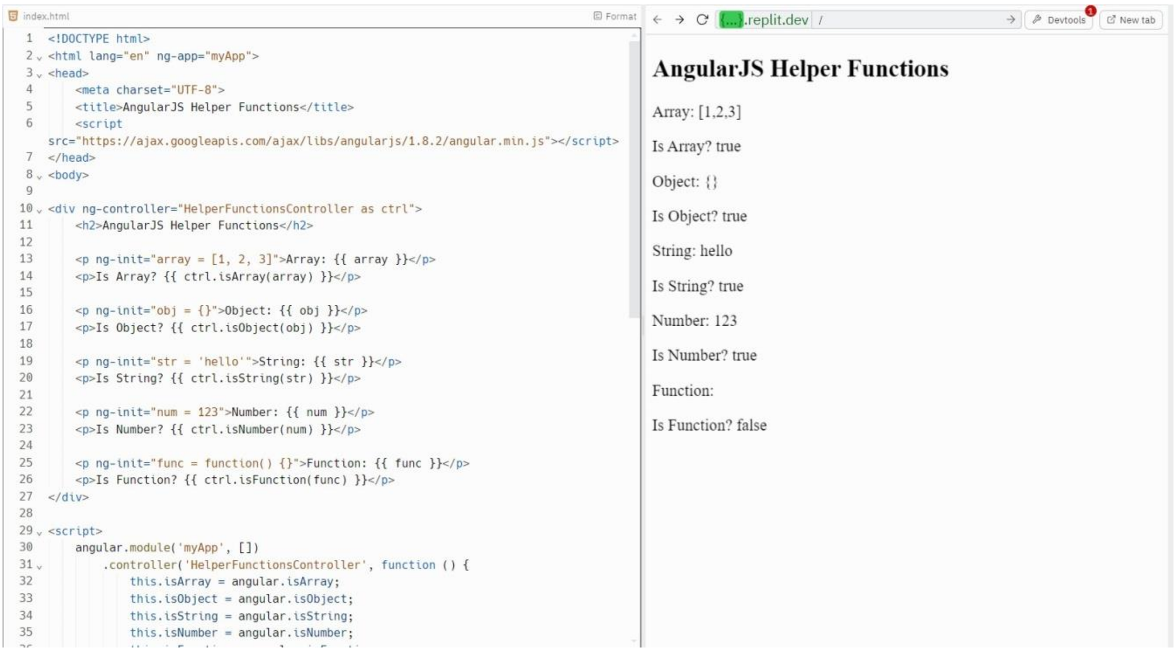
Example:

angular.isFunction(function(){});//true

angular.isFunction('hello');        // false

These helper functions are fundamental in AngularJS development, enabling developers to perform type-checking operations efficiently and effectively. By using these functions, developers can write more robust and maintainable code, leading to better application performance and readability.

PROGRAM:



```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>AngularJS Helper Functions</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="HelperFunctionsController as ctrl">
    <h2>AngularJS Helper Functions</h2>

    <p ng-init="array = [1, 2, 3]">Array: {{ array }}</p>
    <p>Is Array? {{ ctrl.isArray(array) }}</p>

    <p ng-init="obj = {}">Object: {{ obj }}</p>
    <p>Is Object? {{ ctrl.isObject(obj) }}</p>

    <p ng-init="str = 'hello'">String: {{ str }}</p>
    <p>Is String? {{ ctrl.isString(str) }}</p>

    <p ng-init="num = 123">Number: {{ num }}</p>
    <p>Is Number? {{ ctrl.isNumber(num) }}</p>

    <p ng-init="func = function() {}">Function: {{ func }}</p>
    <p>Is Function? {{ ctrl.isFunction(func) }}</p>
</div>

<script>
    angular.module('myApp', [])
        .controller('HelperFunctionsController', function () {
            this.isArray = angular.isArray;
            this.isObject = angular.isObject;
            this.isString = angular.isString;
            this.isNumber = angular.isNumber;
```

## AngularJS Helper Functions

```
Array: [1,2,3]

Is Array? true

Object: {}

Is Object? true

String: hello

Is String? true

Number: 123

Is Number? true

Function:

Is Function? false
```