EXPRESSIONS IN ANGULARJS

- AngularJS expressions are used to bind data in HTML.
- AngularJS execute the expressions and return the values where the expression is defined.
- AngularJS expressions are same as JavaScript expressions. It can be used Array, Objects, String and Number.
  - 1+2 – Number.
  - Hello+World - String
  - a.name - Objects
  - items[index] - Array
- AngularJS expressions are written inside two curly braces **{{   }}** otherwise we can use **ng-bind** directives.

AngularJS Expression syntax:

{{ Expressions }}   or   ng-bind="expression"

Difference between AngularJs Expressions and Javascript Expressions:

AngularJs Expressions:

- AngularJS expressions does not support the conditionals, loops and exceptions.
- It support the filters.
- It does not use the function declaration, even inside the ng-init directive.
- Bitwise, comma, void, new operator and regular expression does not supported in AngularJS Expression.
- In AngularJS ignore the **undefined** or **null** properties in expression.
- Angular expressions are evaluated belongs to **scope** object

Javascript Expressions:

- Javascript expressions support the conditionals, loops and exceptions.
- It does not support the filters.
- Javascript expressions use the function declaration.
- Operators are used in Javascript expressions.
- Javascript undefined properties creates **ReferenceError** or **TypeError**
- Javascipt expressions are evaluated belongs to global **window.**

Sample coding for AngularJS Expression:

```html
<!DOCTYPE html>
<html>
   <head>
      <title>AngularJS Expressions with example</title>
      <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js">
   </script>
   </head>
   <body>
      <div ng-app=" " ng-init="color='pink';  exp={ firstword:'wiki',
lastword:'techy' }; arr=[1,2,3,4,5]">
         <h3>Expressions Using Numbers:</h3>
         <p>The Addition of two numbers: {{ 100 + 50 }}</p>

         <h3>AngularJS Expression using the css property</h3>
         <input style="background-color:{{color}}" ng-model="color"
value="{{color}}">

         <h3>AngularJS Expression using String</h3>
         <p>The name is <span ng-bind="exp.firstword +' '+
exp.lastword"></span></p>

         <h3>AngularJs Expression using Objects</h3>
         <p>The firstword is {{ exp.firstword }}</p>

         <h3>AngularJs Expression using array</h3>
         <p>The array value is {{ arr[3] }}</p>
      </div>
   </body>
</html>
```

AngularJS Expressions using Numbers:

```
<p>The Addition of two numbers: {{ 100 + 50 }}</p>
```

- The expression 100+50 executed

AngularJS Expressions using the CSS Property:

```
<input style="background-color:{{color}}" ng-model="color" value="{{color}}">
```

- The expression color (AngularJS variable) executed

AngularJS Expressions using String:

```
<p>The name is <span ng-bind="exp.firstword +' '+ exp.lastword"></span></p>
```

The **ng-bind** is used to bind the expression values to the HTML Document.

AngularJS Expressions using Objects:

```
<p>The firstword is {{ exp.firstword }} </p>
```

The expression **exp.firstword** is get the value of the **firstword** from the object

AngularJS Expressions using Array:

```
<p>The array value is {{ arr[3] }}</p>
```

The expression **arr[3]** will get the 4 as a value from an array (like **arr[0]=1, arr[1]=2, arr[2]=3, arr[3]=4, arr[4]=5)**

Code Explanation for AngularJS Expression:

```html
<!DOCTYPE html>
<html>
    <head>
        <title>AngularJS Expressions with example</title>
    </head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/
        angular.min.js"></script>
    <body>
        <div ng-app="" ng-init="color='pink'; exp={firstword:'wiki',
        lastword:'techy'}; arr=[1,2,3,4,5]">
            <h3>Expressions Using Numbers:</h3>
            <p>The Addition of two numbers:{{ 100 + 50 }}</p>

            <h3>AngularJS Expression using the css property</h3>
            <input style="background-color:{{color}}" ng-model="color"
            value="{{color}}">

            <h3>AngularJS Expression using String</h3>
            <p>The name is <span ng-bind="exp.firstword +''+ exp.lastword">
            </span></p>

            <h3>AngularJs Expression using Objects</h3>
            <p>The firstword is {{ exp.firstword }}</p>

            <h3>AngularJs Expression using array</h3>
            <p>The array value is {{ arr[3] }}</p>
        </div>
    </body>
</html>
```
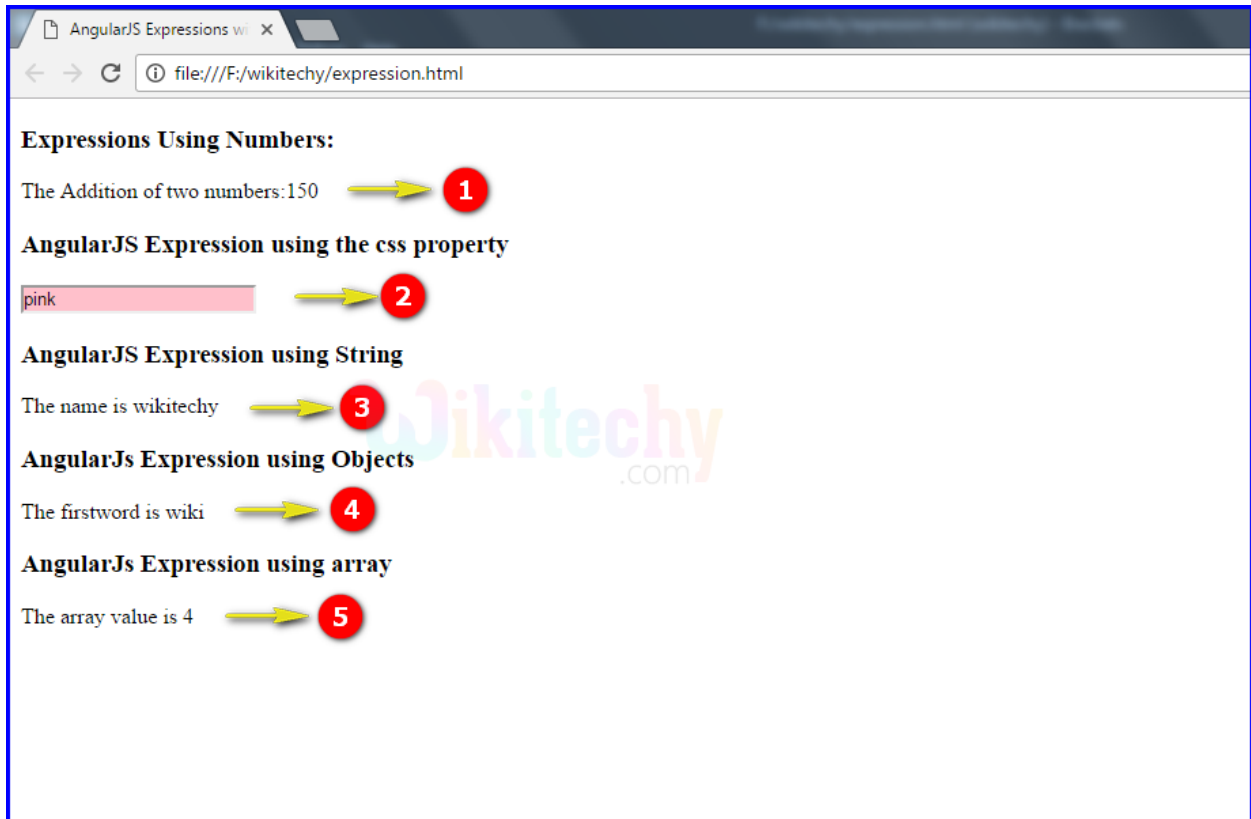
1. The **ng-app** specifies the root element (**<div>**) to define AngularJS application.

2. The **ng**-**init** directive used to initialize the variables (**color**='pink'; **exp**={ **firstword**:'wiki', **lastword**:'techy' }; **arr**=[1,2,3,4,5]).

3. The expression **100+50** executed and the output will be updated in <p> tag.

4. The expression **color (AngularJS variable)** executed and the output will be updated inside the style attribute in <input> tag.

5. The **ng-model** bind an input field value to AngularJS application variable (**"color"**).

6. The **ng-bind** is used to bind the expression (**"exp.firstword +''+ exp.lastword"**) values to the HTML Document.

7. The expression **exp.firstword** is get the value of the **firstword** from the object **(exp)** and that will be updated in the <p> tag.

8. The expression **arr[3]** will get the 4 as a value from an array (like **arr[0]=1, arr[1]=2, arr[2]=3, arr[3]=4, arr[4]=5)** and the output will displayed in the <p> tag.

Sample output:

1. The output displays the addition of two numbers is 150.

2. The output shows the default background color as pink when you change the color name it will be taken as value and will change the background color automatically.

3. The output displays the concatenation of two strings "wiki" and "techy".

4. The output displays the firstword "wiki" by the **exp** object.

5. The output displays the array value 4.