Angular v2 Template Syntax Summary

*Status: (Draft)*

*Authors:* [*misko@google.com*](mailto:misko@google.com)*,* [*iminar@google.com*](mailto:iminar@google.com)

*This document is published to the web as part of the public* [*Angular Design Docs*](https://drive.google.com/#folders/0BxgtL8yFJbacUnUxc3l5aTZrbVk) *folder*

# Objective

# Background

Stuff one needs to know to understand this doc: motivating examples, previous versions and problems, links to related projects/design docs, etc. You should mention related work if applicable. This is background; do not write about ideas to solve problems here.

# Prior Art

* Html template discussion [document](https://docs.google.com/document/d/1kpuR512G1b0D8egl9245OHaG0cFh0ST0ekhD_g8sxtI/edit#heading=h.xgjl2srtytjt).
* Annotations discussion [document](https://docs.google.com/a/google.com/document/d/1uhs-a41dp2z0NLs-QiXYY-rqLGhgjmTf4iwBad2myzY/edit#heading=h.qbaubqkoiqds).
* Event discussion [document](https://docs.google.com/a/google.com/document/d/1d-yxmxMVqZHHadvxQaC2CCvcAR17JUQ1Qs4RT6Rwp6Y/edit).
* ES6+A Q&A [document](https://docs.google.com/a/google.com/document/d/1cUTD8oVzfpwFqX5tMxHTifKO8uJm5VddwmB0aVQMxpI/edit) and [design document](https://docs.google.com/document/d/1uhs-a41dp2z0NLs-QiXYY-rqLGhgjmTf4iwBad2myzY/edit#heading=h.qbaubqkoiqds).

# Detailed Design

### Property Bindings

This syntax specifies how the element property is bound to an expression. These are based on HTML attributes. The attribute name is escaped to signify property binding and not just an attribute, and the attribute value always represents an angular expression.

<div [property-name]="expression">

In the above syntax the angular expression "expression" is bound to the element's property "propertyName". Note that dash to camelCase property conversion. For an IDE it is safe to assume that the right hand side will always be an angular expression and it can color/code analyze accordingly.

### Parameterized Property Bindings

There are times when the binding need to be further parameterized in a way which is not an expression. In such a case an optional parameter can be associated with the property binding. NOTE: currently only one parameter is allowed, we may extend this in the future.

<div [ng-repeat|person]="people">  
 <span [text]="person.name"></span>  
</div>

In the above example the "ng-repeat" directive is parameterized with "person". This allows the "ng-repeat" to use the "person" as a local variable later in the template.

### Interpolation

We have chosen to support ES6 type interpolation rather than old style '{{}}' . The interpolation can be placed in attributes or in text nodes. NOTE: we may implement '{{}}' for backwards compatibility as a deprecated feature.

<div title="${title}">Hello ${name}</div>

The above syntax is a shorthand for the actual syntax below.

<div [title]="title|stringify" [text|2]="name|stringify">  
 Hello <!--${name}--> </div>

While the IDE does not have to think about the translation, it is important to realize that semantically this is the execution context. Note that [text|2] binding which is placed on element, but it actually means that second (2) child text node should be updated by this binding.

### Events

Events allow the registration of DOM/Angular events to angular expressions.

<button (click)="doSomething()">click me</button>

The event name is encoded in the attribute name as such: (event-name). In this case the event listener only fires if the source of the event is originating from the underlying DOM element (The example above does not have any such children, but if it had children, a click event bubbling from a child would not trigger that handler). If the event listener should allow bubbling, than the event name is prefixed with "^" as such.

<div (^click)="doSomithing">  
 <img src="..."><span>text</span>  
</div>

In the above example the click could originate from div, img, or span.

### References

There are times when an event needs to get a hold of another component to invoke methods on it. In this case the other component needs to have an ID by which we can reference it. The ID is in the form of #name.

<input #login-element type=text>  
<button (click)="loginElement.focus()">

In the above case the #login-element gets camelCased to loginElement which can then be referenced from the button's click handler.

### Example

|  |
| --- |
| <html>  <body>  <zippy #zippy title="Greeting">  Body of the text which is shown conditionally.  <a href (hover)="zippy.close()">hover to close</a>.  </zippy>  <button (click)="zippy.toggle()">toggle</button>  </body>  </html> |

# Caveats

You may need to describe what you did not do or why simpler approaches don't work. Mention other things to watch out for (if any).

# Security Considerations

How you’ll be secure

# Performance Considerations / Test Strategy

How you’ll be fast.

# Work Breakdown

Description of development phases and approximate time estimates.