## NgNonBindable

**Learning Objectives**

• Understand when and how to use the NgNonBindable directive.

**Description**

We use ngNonBindable when we want tell Angular not to compile, or bind, a particular section of our page.

The most common example of this is if we wanted to write out some Angular code on the page, for example if we wanted to render out the text {{ name }} on our page, like so:

<

div

>

To render the name variable we use this syntax <pre>{{ name }}</pre>

<

/div

>

Normally Angular will try to find a variable called name on the component and print out the value of the name variable instead of just printing out {{ name }}.

To make angular ignore an element we simply add the ngNonBindable directive to the element, like so:

<

div

>

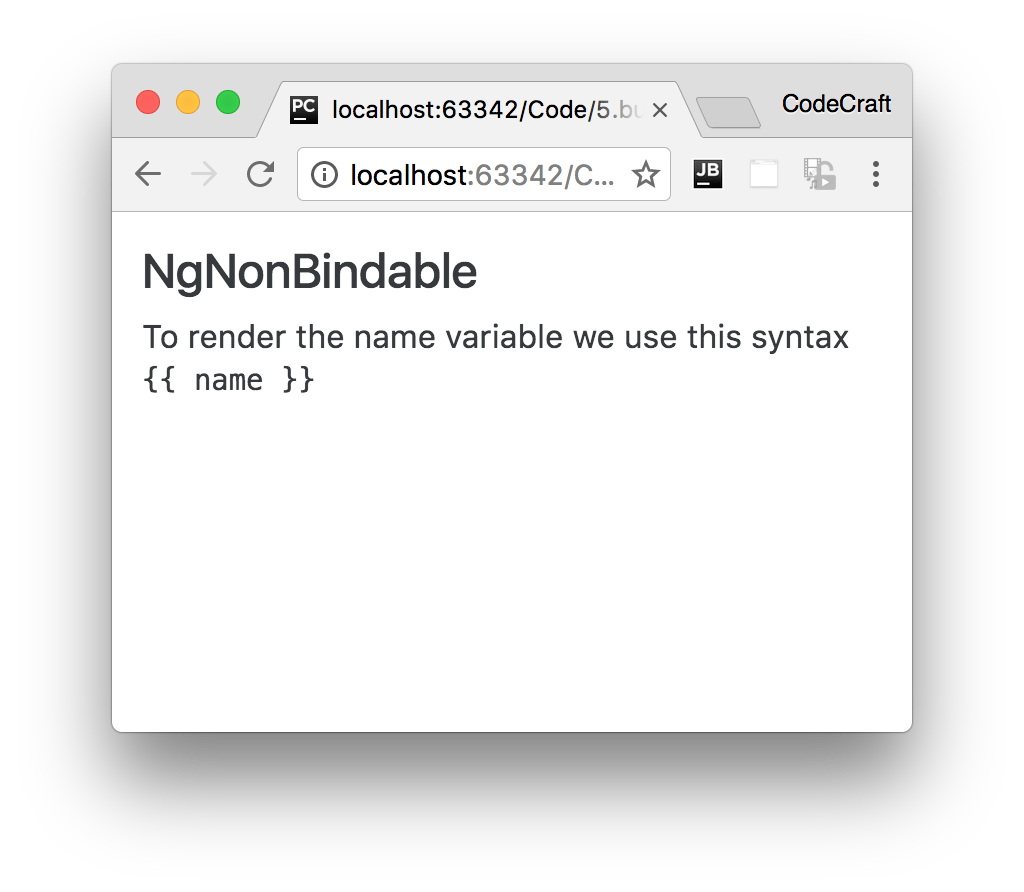
To render the name variable we use this syntax <pre ngNonBindable>{{ name }}</pre>

<

/div

>

If we run this in the browser we would see:



**Summary**

We use NgNonBindable when we want to tell Angular not to perform any binding for an element.

**Listing**

<http://plnkr.co/edit/1u12vtOPFA9K00jvUjlO?p=preview>

*script.ts*

import {NgModule, Component} from '@angular/core';

import {BrowserModule} from '@angular/platform-browser';

import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';

@Component({

selector: 'ngnonbindable-example',

template: `<h4>NgNonBindable</h4>

<

div

>

To render the name variable we use this syntax

<pre ngNonBindable>{{ name }}</pre>

<

/div

>

`

})

class NgNonBindableExampleComponent {

}

@Component({

selector: 'directives-app',

template: `<ngnonbindable-example></ngnonbindable-example>`

})

class DirectivesAppComponent {

}

@NgModule({

imports: [BrowserModule],

declarations: [NgNonBindableExampleComponent, DirectivesAppComponent],

bootstrap: [DirectivesAppComponent],

})

class AppModule {

}

platformBrowserDynamic().bootstrapModule(AppModule);