Apunte Angular

Gustavo García, 2/04/2023.

Para estudiar Angular, ir al repositorio 20-que-es-angular abrir el archivo Qué es Angular.docx, y seguir las instrucciones. Tanto este apunte, como las preguntas que irán en las evaluaciones, están tomados de los materiales que allí se indican. Todos los proyectos del repositorio están probados y funcionan perfectamente en mi PC. Puede ocurrir que en la PC de alguien haya algo que no funciona. Pero los proyectos en sí mismos funcionan con toda seguridad.

# All About Angular Bindings

<https://www.telerik.com/blogs/all-about-angular-bindings>

Los bindings en Angular son un poderoso mecanismo integrado en el marco. Desafortunadamente, pueden ser difíciles de recordar. A veces, cuando busco el enlace correcto, me equivoco u olvido la sintaxis. Pensé que un artículo podría ayudar a aclarar cómo y cuándo aplicar enlaces en sus aplicaciones Angular, para esos momentos de memoria que todos tenemos a veces. A continuación se muestra un gráfico práctico para una referencia rápida (vinculado a cada sección):

| **Binding** | **Example** |
| --- | --- |
| [Property Binding](https://www.telerik.com/blogs/all-about-angular-bindings#property-binding) | <input [placeholder]="placeholderValue" /> |
| [String Interpolation](https://www.telerik.com/blogs/all-about-angular-bindings#string-interpolation-vs-property-binding) | <input placeholder="{{placeholderValue}}" /> |
| Attribute Binding | <td [attr.colspan]="clspn"> |
| [Style Binding](https://www.telerik.com/blogs/all-about-angular-bindings#style-binding) | <input [style.borderStyle]="'dashed'" /> |
| [Class Binding](https://www.telerik.com/blogs/all-about-angular-bindings#class-binding) | <input [class.active]="true" /> |
| [Event Binding](https://www.telerik.com/blogs/all-about-angular-bindings#event-binding) | <input (keyup.enter)="onEnter()" /> |
| [Two-way Binding](https://www.telerik.com/blogs/all-about-angular-bindings#two-way-binding) | <input [(ngModel)]="value" /> |

# Propiedades opcionales

<https://www.typescriptlang.org/docs/handbook/2/objects.html#optional-properties>

Una propiedad de un objeto, que podría estar o no presente se indica con un signo de pregunta (?) inmediatamente después del nombre. Por ejemplo:

miPropiedad?: string;

Significa que miPropiedad podría estar o no presente. Si está presente, deberá ser de tipo (:) string.

**Cómo funciona el operador de signo de interrogación (?) en JavaScript**

<https://www.freecodecamp.org/news/how-the-question-mark-works-in-javascript/>

# Versión vieja

A veces sale un mensaje avisando que la versión global es más nueva que la versión local del proyecto.

Se puede ignorar. Lo que pasa es que Angular mantiene las versiones con las que el proyecto se construyó originalmente, para no arriesgar incompatibilidades.

Pero si uno está seguro de que no va a romper nada, y realmente quiere actualizar su proyecto, en la terminal local del proyecto, escribir:

ng update

Esto va a producir una lista de paquetes que hay que actualizar, y va a proponer un comando. Ese comando, tal como lo propone Angular, no corre, da un mensaje de error avisando que el cache está sucio. Es un bug de Angular, según dicen los foros. Solución:

ng update @angular/cli @angular/core --allow-dirty

Si ahora se ejecuta el comando ng serve ya no va a dar el mensaje de versión outdated.