When you use Template expressions or interpolation If you reference a name that belongs to more than one of these namespaces, the template variable name takes precedence, followed by a name in the directive's context, and, lastly, the component's member names.

Data-binding works with properties of DOM elements, components, and directives, not HTML attributes. **attributes initialize DOM properties and then they are done. Property values can change; attribute values can't.**

Interpolation is a convenient alternative to property binding in many cases. When rendering data values as strings, there is no technical reason to prefer one form to the other, though readability tends to favor interpolation. However, when setting an element property to a non-string data value, you must use property binding.

[attr.class]="resetClasses" - Reset all [classes](https://angular.io/api/core/DebugElement#classes) at once

[class.item-clearance]="itemClearance" - Add another class

[class.special]="!isSpecial" - This one is not so special.

Style property:

<button [style.color]="isSpecial ? 'red': 'green'">Red</button> <button [style.background-color]="canSave ? 'cyan': 'grey'" >Save</button>

<button [style.font-size.em]="isSpecial ? 3 : 1" >Big</button> <button [style.font-size.%]="!isSpecial ? 150 : 50" >Small</button>

Event binding

<button (click)="onSave($event)">Save</button>

<button on-click="onSave($event)">on-click Save</button>

**Directives**

1)attributive

[ngClass] ={'saveable': this.canSave, 'modified': !this.isUnchanged, 'special': this.isSpecial}

[[ngStyle](https://angular.io/api/common/NgStyle)]="currentStyles"

this.currentStyles = { 'font-style': this.canSave ? 'italic' : 'normal', 'font-weight': !this.isUnchanged ? 'bold' : 'normal', 'font-size': this.isSpecial ? '24px' : '12px' };

[([ngModel](https://angular.io/api/forms/NgModel))]="currentItem.name" – (two way) for forms elements binds to value property

2)structural

\*[ngIf](https://angular.io/api/common/NgIf) ="[isActive](https://angular.io/api/router/RouterLinkActive" \l "isActive)" - When the expression is falsy, [NgIf](https://angular.io/api/common/NgIf) removes the ItemDetailComponent from the DOM.If yoyu use display hidden the element remains in the DOM

\*[ngFor](https://angular.io/api/common/NgForOf)="let [item](https://angular.io/api/core/IterableChangeRecord#item) of items" [[item](https://angular.io/api/core/IterableChangeRecord#item)]="[item](https://angular.io/api/core/IterableChangeRecord#item)"

[OnChanges](https://angular.io/api/core/OnChanges) is specifically designed to work with properties that have the @[Input](https://angular.io/api/core/Input)() decorator.

CHANGE DETECTION

Change Detection means updating the DOM whenever data is changed. Angular provides two strategies for Change Detection. In the onPush strategy, Angular will only run the change detector  when a new reference is passed to @Input() data.