

Keeping Your DOM Clean



David Mann

@MannD | www.HeirloomSoftware.com



Why a Clean DOM Matters



Avoid potential CSS selector problems



Accessibility



Debugging



```
<p>I'd like to say
  <ng-container *ngIf="arriving">
    Hello World
  </ng-container>
  <ng-container *ngIf="!arriving">
    Goodbye World
  </ng-container >
</p>
```

```
<p _ngcontent-c0>
  "I'd like to say "
  "Goodbye World"
</p>
```



Component Attribute Selector

```
@Component({  
  selector: 'app-comp-2',  
  templateUrl: './comp2.component.html',  
  styleUrls: ['./comp2.component.css']  
})
```

```
<div class='myClass'>  
  <app-comp-2></app-comp-2>  
</div>
```

```
▼ <div _ngcontent-c0 class="myClass">  
  <app-comp-2 _ngcontent-c0 _ngghost-c1>  
    Hello from Comp2  
  </app-comp-2>  
</div>
```

Regular “element” selector



Component Attribute Selector

```
@Component({  
  selector: 'app-comp-2',  
  templateUrl: './comp2.component.html',  
  styleUrls: ['./comp2.component.css']  
})
```

```
<div class='myClass'>  
  <app-comp-2></app-comp-2>  
</div>
```

```
▼<div _ngcontent-c0 class="myClass">  
  <app-comp-2 _ngcontent-c0 _ngghost-c1>  
    Hello from Comp2  
  </app-comp-2>  
</div>
```

Regular “element” selector

```
@Component({  
  selector: '[comp2]',  
  templateUrl: './comp2.component.html',  
  styleUrls: ['./comp2.component.css']  
})
```

```
<div class='myClass' comp2 ></div>
```

```
<div _ngcontent-c0 class="myClass" comp2 _ngghost-c1>  
  Hello from Comp2  
</div>
```

Attribute selector



```
▼<app-root _ngghost-c0 ng-version="5.0.3">
  <router-outlet _ngcontent-c0></router-outlet>
  <!--bindings={
    "ng-reflect-ng-if": "false"
  }-->
  ▼<div _ngcontent-c0 style="display: none;">
    ▼<mat-grid-list _ngcontent-c0 class="mat-grid-list" cols="3" guttersize="10px" ng-reflect-
      cols="3" ng-reflect-gutter-size="10px" ng-reflect-row-height="44px" style="height:
      calc(98px);">
      ▼<div>
        ▶<mat-grid-tile _ngcontent-c0 class="widget-zone-wrapper mat-grid-tile" colspan="1"
          rowspan="2" ng-reflect-rowspan="2" ng-reflect-colspan="1" style="left: 0px; width:
          calc(((33.3333% - 6.66667px) * 1) + 0px); top: 0px; height: calc(98px);">...</mat-grid-
          tile>
        ▶<mat-grid-tile _ngcontent-c0 class="widget-zone-wrapper mat-grid-tile" colspan="1"
          rowspan="1" ng-reflect-rowspan="1" ng-reflect-colspan="1" style="left: calc(((33.3333%
          - 6.66667px) + 10px) * 1); width: calc(((33.3333% - 6.66667px) * 1) + 0px); top: 0px;
          height: calc(44px);">...</mat-grid-tile>
        ▶<mat-grid-tile _ngcontent-c0 class="widget-zone-wrapper mat-grid-tile" colspan="1"
          rowspan="2" ng-reflect-rowspan="2" ng-reflect-colspan="1" style="left: calc(((33.3333%
          - 6.66667px) + 10px) * 2); width: calc(((33.3333% - 6.66667px) * 1) + 0px); top: 0px;
          height: calc(98px);">...</mat-grid-tile>
        ▶<mat-grid-tile _ngcontent-c0 class="widget-zone-wrapper mat-grid-tile" colspan="1"
          rowspan="1" ng-reflect-rowspan="1" ng-reflect-colspan="1" style="left: calc(((33.3333%
          - 6.66667px) + 10px) * 1); width: calc(((33.3333% - 6.66667px) * 1) + 0px); top:
          calc(54px); height: calc(44px);">...</mat-grid-tile>
      </div>
    </mat-grid-list>
  </div>
</app-root>
```



display:none

```
<div [style.display]="showGrid? 'block' : 'none'">
```



```
▼<app-root _ngghost-c0 ng-version="5.0.3">
  <router-outlet _ngcontent-c0></router-outlet>
  <!--bindings={
    "ng-reflect-ng-if": "false"
  }-->
  ▼<div _ngcontent-c0 hidden>
    ▼<mat-grid-list _ngcontent-c0 class="mat-grid-list" cols="3" guttersize="10px" ng-reflect-
      cols="3" ng-reflect-gutter-size="10px" ng-reflect-row-height="44px" style="height:
      calc(98px);">
      ▼<div>
        ▶<mat-grid-tile _ngcontent-c0 class="widget-zone-wrapper mat-grid-tile" colspan="1"
          rowspan="2" ng-reflect-rowspan="2" ng-reflect-colspan="1" style="left: 0px; width:
          calc(((33.3333% - 6.66667px) * 1) + 0px); top: 0px; height: calc(98px);">...</mat-grid-
          tile>
        ▶<mat-grid-tile _ngcontent-c0 class="widget-zone-wrapper mat-grid-tile" colspan="1"
          rowspan="1" ng-reflect-rowspan="1" ng-reflect-colspan="1" style="left: calc(((33.3333%
          - 6.66667px) + 10px) * 1); width: calc(((33.3333% - 6.66667px) * 1) + 0px); top: 0px;
          height: calc(44px);">...</mat-grid-tile>
        ▶<mat-grid-tile _ngcontent-c0 class="widget-zone-wrapper mat-grid-tile" colspan="1"
          rowspan="2" ng-reflect-rowspan="2" ng-reflect-colspan="1" style="left: calc(((33.3333%
          - 6.66667px) + 10px) * 2); width: calc(((33.3333% - 6.66667px) * 1) + 0px); top: 0px;
          height: calc(98px);">...</mat-grid-tile>
        ▶<mat-grid-tile _ngcontent-c0 class="widget-zone-wrapper mat-grid-tile" colspan="1"
          rowspan="1" ng-reflect-rowspan="1" ng-reflect-colspan="1" style="left: calc(((33.3333%
          - 6.66667px) + 10px) * 1); width: calc(((33.3333% - 6.66667px) * 1) + 0px); top:
          calc(54px); height: calc(44px);">...</mat-grid-tile>
      </div>
    </mat-grid-list>
  </div>
</app-root>
```



hidden

```
<div [hidden]='!showGrid'>
```



*ngIf

```
▼<app-root _ngghost-c0 ng-version="5.0.3">  
  <router-outlet _ngcontent-c0></router-outlet>  
  <!--bindings={  
    "ng-reflect-ng-if": "false"  
  }-->  
  <!--bindings={}-->  
</app-root>
```



***ngIf**

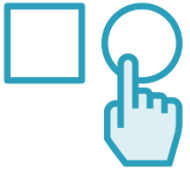
```
<div *ngIf="showGrid">
```

```
  showGrid: boolean = false;
```

```
  toggleGrid() {  
    |   this.showGrid = !this.showGrid;  
  }  
}
```



A Note on Performance



Be selective



As high as possible



Frequency



```
<ng-container [ngSwitch]="switchVar">
  <ng-container *ngSwitchCase="'a'">
    <p>This is case 1.</p>
    <p>It can get pretty long and hard to work with
      in your HTML file</p>
    <p>if there's a lot of "stuff" to show for each
      case</p>
  </ng-container>
  <ng-container *ngSwitchCase="'b'">
    <p>This is case 2.</p>
    <p>While only one of these show up in the UI</p>
    <p>they're all specified in the HTML inside the
      ngSwitch</p>
  </ng-container>
  <ng-container *ngSwitchCase="'c'">
    <p>This is case 3.</p>
    <p>Which can really bloat your HTML file</p>
    <p>especially if you use ngSwitch a lot</p>
  </ng-container>
  <ng-container *ngSwitchDefault>
    <p>Default...</p>
  </ng-container>
</ng-container>
```

◀ ngSwitch *bloat*



Using NgTemplateOutlet

```
<ng-container [ngSwitch]="switchVar">
  <ng-container *ngSwitchCase="'a'">
    <ng-container *ngTemplateOutlet="template1"></ng-container>
  </ng-container>
  <ng-container *ngSwitchCase="'b'">
    <ng-container *ngTemplateOutlet="template2"></ng-container>
  </ng-container>
  <ng-container *ngSwitchCase="'c'">
    <ng-container *ngTemplateOutlet="template3"></ng-container>
  </ng-container>
  <ng-container *ngSwitchDefault>
    <ng-container>Default...</ng-container>
  </ng-container>
</ng-container>
```

Augment ngSwitch

```
<ng-container *ngTemplateOutlet="getTemplate()"></ng-container>
```

Replace ngSwitch



Shadow DOM

```
<span _ngcontent-c1 class="viewType">View</span>
```

```
.viewType[_ngcontent-c1] {  
  text-decoration: ► underline;  
}
```

```
<span class="viewType">View</span>
```

```
.viewType {  
  text-decoration: ► underline;  
}
```

Shadow DOM

- Chrome 61+



Key Takeaways



DOM Hygiene

- NgContainer
- Component Attribute Selectors
- NgIf
- NgSwitch
- Shadow DOM

Next Up



UI Performance

