Topics

Built-in content projection

NgContainer

NgTemplate

- If/Else

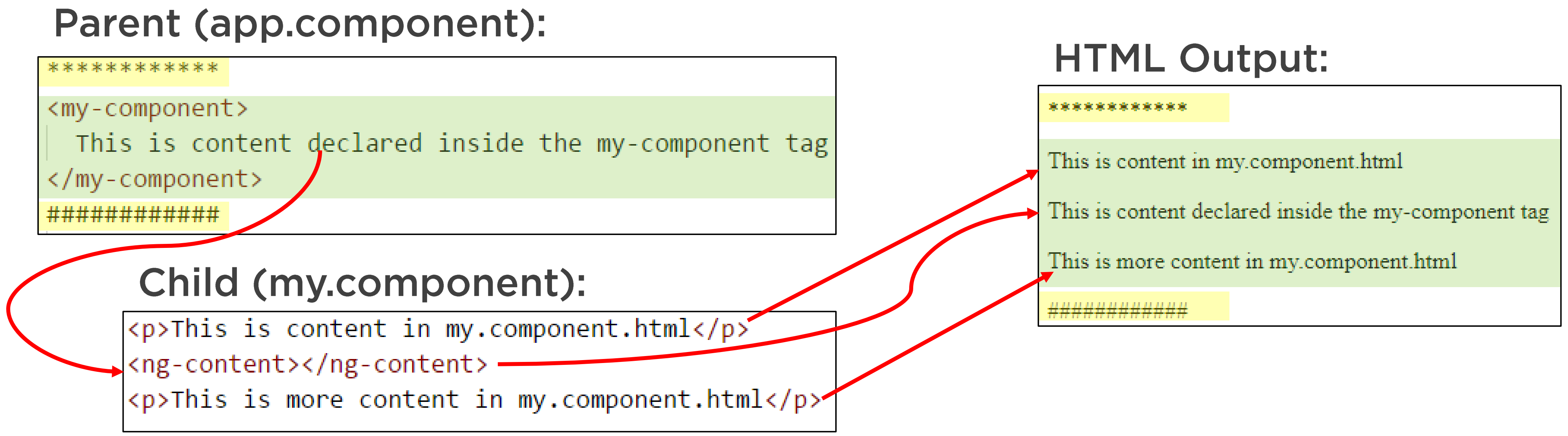
Content Projection with NgTemplateOutlet

- NgTemplateOutletContext

Component Inheritance

Dynamic Component Creation

Built-in Content Projection



Parent (app.component):

```
*************
<my-component>
  This is content declared inside the my-component tag
</my-component>
#############
```

Child (my.component):

```
<p>This is content in my.component.html</p>
<ng-content></ng-content>
<p>This is more content in my.component.html</p>
```

HTML Output:

```
*************
This is content in my.component.html
This is content declared inside the my-component tag
This is more content in my.component.html
#############
```

Named-content Projection

- Attribute
- CSS Selector
- Component

...and???

Use Cases for ng-container

Cleaner DOM

Avoid potential CSS selector problems

Accessibility

```
<ng-template #tmplA>
  Hello from TemplateA
</ng-template>
```

ng-template

<!---->

true false

```
<ng-container *ngIf="arriving;
  then arrivingContent
  else departingContent">
</ng-container>


<!-- Templates @ page-bottom -->

<ng-template #arrivingContent>
  Hello World
</ng-template>
<ng-template #departingContent>
  Goodbye World
</ng-template>
```

◀ *ngIf-then-else

NgTemplateOutlet

```
<ng-template #tmplA>
  Template A
</ng-template>
```

```
<ng-template #tmplB>
  Template B
</ng-template>
```

```
<ng-container  *ngTemplateOutlet='tmplA'>
   <ng-template #tmplA>
     Template A
   </ng-template>
</ng-container>
```

NgTemplateOutlet

```html
<ng-template #tmplA>
  Template A
</ng-template>
```

```html
<ng-template #tmplB>
  Template B
</ng-template>
```

```html
<ng-container *ngTemplateOutlet='tmplB'>

  <ng-template #tmplB>
    Template B
  </ng-template>

</ng-container>
```

ngTemplateOutlet Use-Cases

Alternate UI

Repeated UI Elements

Dynamic UI Placement

NgTemplateOutlet Context

```
<ng-template #tmplB
  let-name let-desc='desc'>
  <p>{{name}} is {{desc}}</p>
</ng-template>
```

JSON Object

Bind to template declarations (*let-*)

Declarative

Out of the Box

Content-only

Component Inheritance

~~Component metadata~~

~~Template~~

~~Styles~~

Component Inheritance

?

Component Inheritance Scenarios

Alternate Presentation

Alternate Logic

Alternate Data

Dynamic Component Creation

Dynamic Components

~~Lazy Loading~~

Component Instantiation

Dynamic Components

*ngComponentOutlet	ComponentFactoryResolver
(mostly) Declarative	Programmatic
Easy	Not Hard
(Pending PR)	Access to component instance