# Content Projection

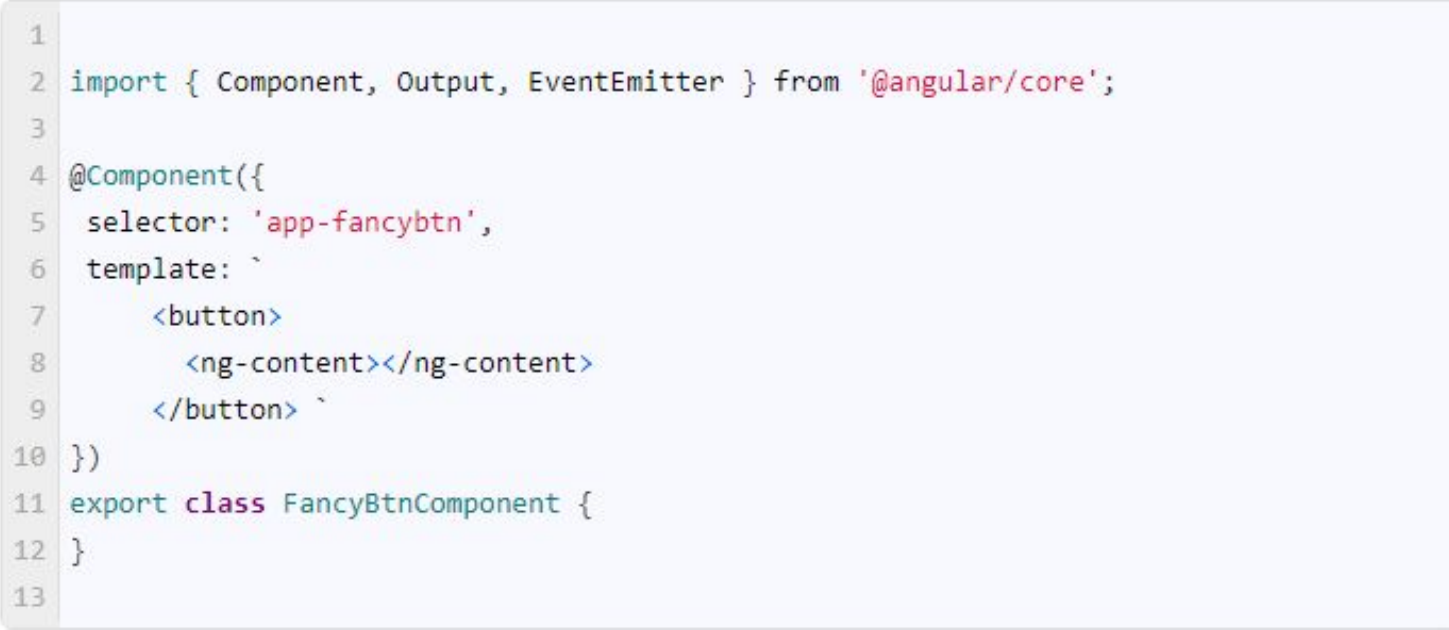
In this guide let us explore how to use `ng-content` to add external content (**content projection**) in the Template. We know how to use **@input** decorator to pass data from the parent component to the child component. But it is only limited to data. We cannot use that technique to pass the content which includes the HTML elements, CSS, etc to the child component. To do that we have to make use of `content projection`.

# Content Projection

- Content projection is a way to pass the HTML content from the parent component to the child component.
- The child component will display the template in a designated spot.
- We use the ng-content element to designate a spot in the template of the child component.

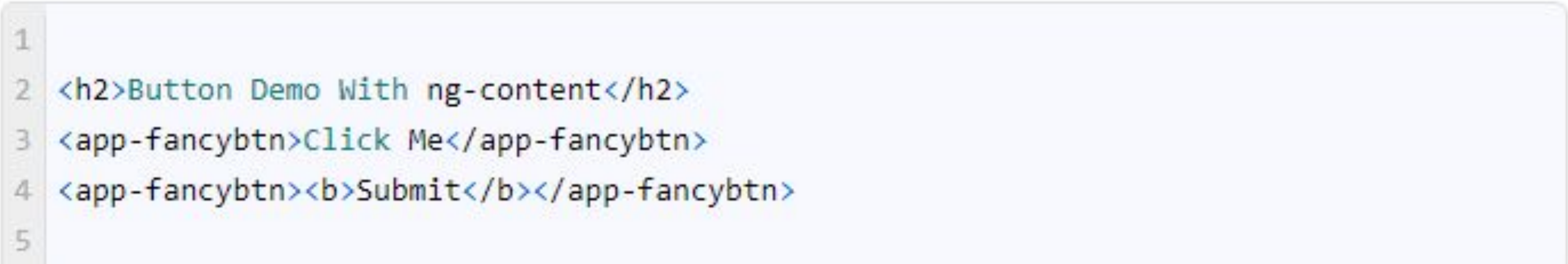
# Example:

FancyBtnComponent.component.ts:

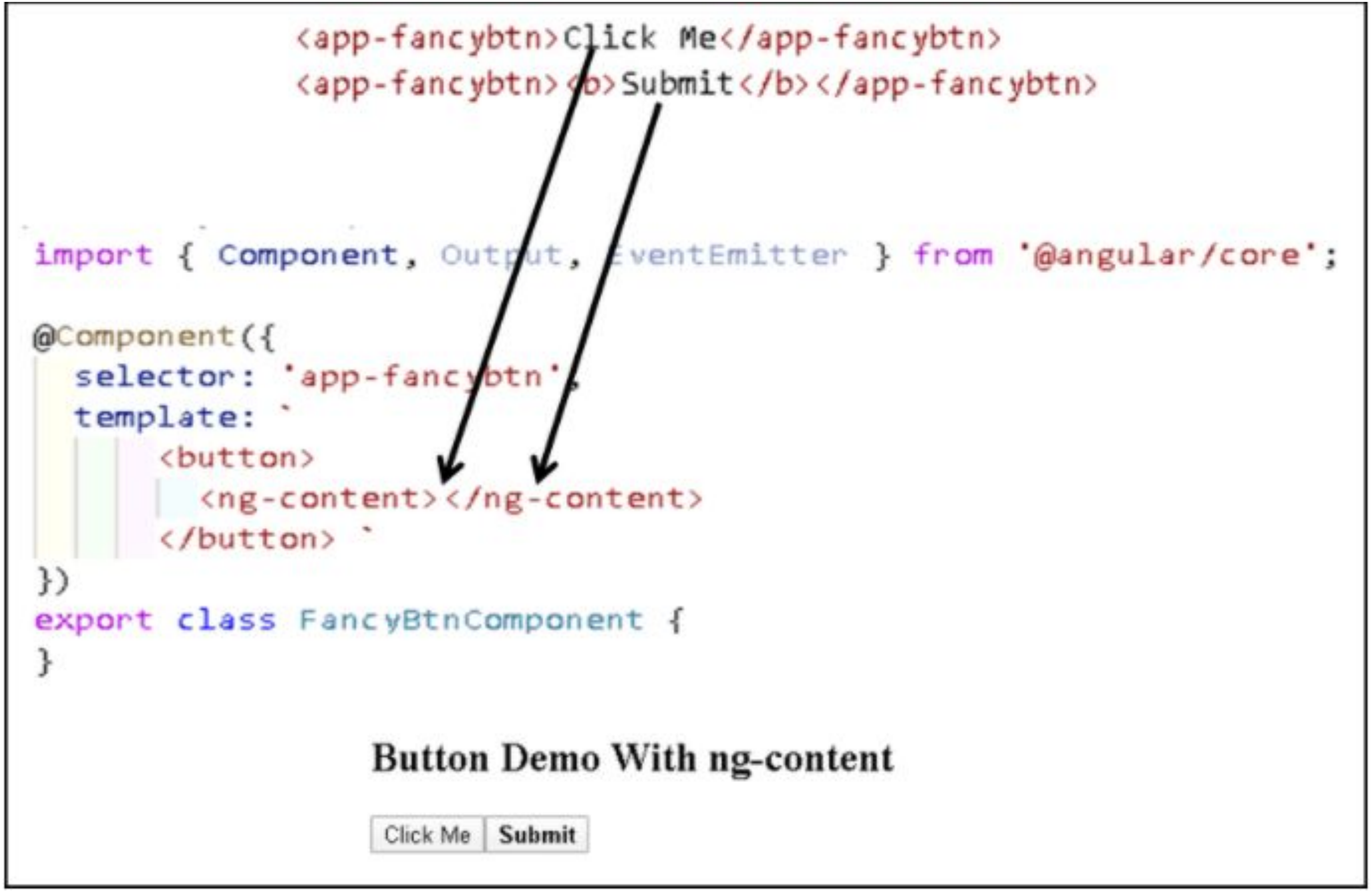


```

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

@Component({
 selector: 'app-fancybtn',
 template: `
     <button>
       <ng-content></ng-content>
     </button> `
})
export class FancyBtnComponent {
}

```

app.component.html :



```html

<h2>Button Demo With ng-content</h2>
<app-fancybtn>Click Me</app-fancybtn>
<app-fancybtn><b>Submit</b></app-fancybtn>

```



```
<app-fancybtn>Click Me</app-fancybtn>
<app-fancybtn><b>Submit</b></app-fancybtn>
```

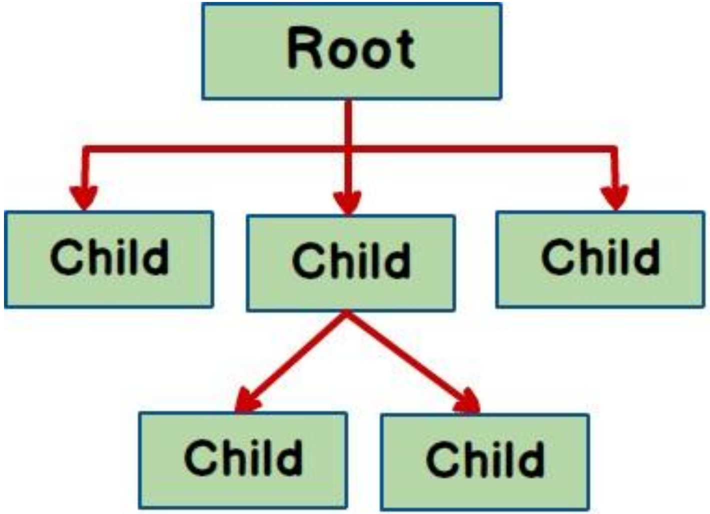
```
import { Component, Output, EventEmitter } from '@angular/core';

@Component({
  selector: 'app-fancybtn',
  template: `
      <button>
        <ng-content></ng-content>
      </button> `
})
export class FancyBtnComponent {
}
```

Button Demo With ng-content

# View child ,Children

# ViewChild DECORATOR

Property decorator that configures a view query. The change detector looks for the first element or the directive matching the selector in the view DOM. If the view DOM changes, and a new child matches the selector, the property is updated.



## Metadata Properties:

- selector - The directive type or the name used for querying.
- read - Used to read a different token from the queried elements.
- static - True to resolve query results before change detection runs, false to resolve after change detection. Defaults to false.

The angular ElementRef class is one way to directly access to the DOM for manipulation. Angular official documentation on ElementRef is *a wrapper around a native element inside of a View.*

# ViewChildren DECORATOR

Use to get the QueryList of elements or directives from the view DOM. Any time a child element is added, removed, or moved, the query list will be updated, and the changes observable of the query list will emit a new value.

View queries are set before the `ngAfterViewInit` callback is called.

Metadata Properties:

- selector - The directive type or the name used for querying.
- read - Used to read a different token from the queried elements.

# ContentChild

Use to get the first element or the directive matching the selector from the content DOM. If the content DOM changes, and a new child matches the selector, the property will be updated.

Content queries are set before the `ngAfterContentInit` callback is called.

# ContentChildren

Use to get the QueryList of elements or directives from the content DOM. Any time a child element is added, removed, or moved, the query list will be updated, and the changes observable of the query list will emit a new value.

Content queries are set before the `ngAfterContentInit` callback is called.