View Queries

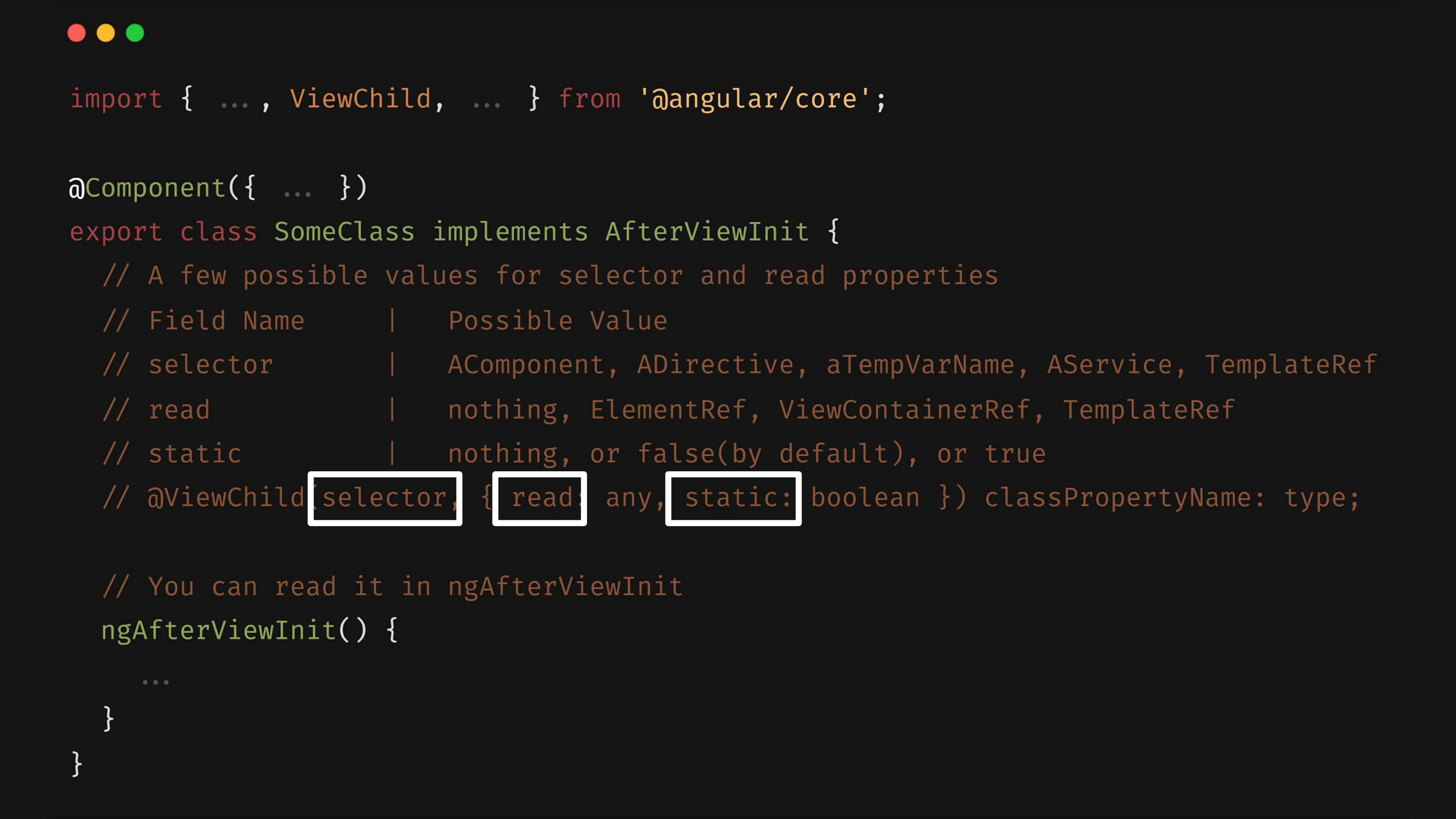
@SiddAjmera

@ViewChild, @ViewChildren, @ContentChild, and @ContentChildren can be used to get element references

@ViewChild

- Used to get element reference from the View.
- Reference can be obtained via Component Type or by Template Variable Name or by simply using an ng-template tag.
- By default gives:
  - Component Instance if used with a Component.
  - ElementRef Instance if used on an HTML Element.
- If read property is used, gives:
  - ElementRef or ViewContainerRef or TemplateRef
- Generally accessed in ngAfterViewInit.
- Read more about @ViewChild here.

@ViewChild Examples

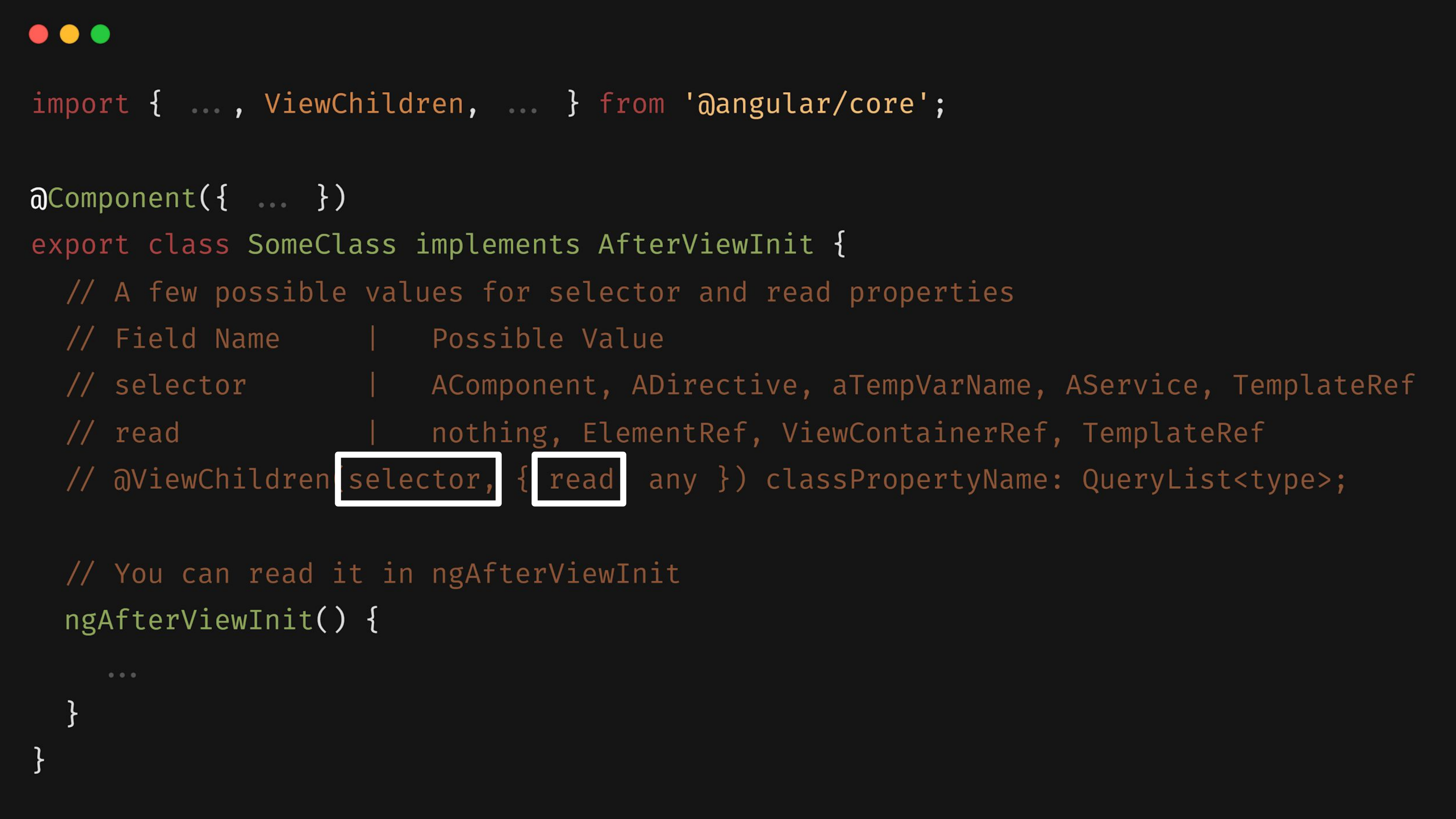


```
import { ..., ViewChild, ... } from '@angular/core';

@Component({ ... })
export class SomeClass implements AfterViewInit {
  // A few possible values for selector and read properties
  // Field Name     |   Possible Value
  // selector       |   AComponent, ADirective, aTempVarName, AService, TemplateRef
  // read           |   nothing, ElementRef, ViewContainerRef, TemplateRef
  // static         |   nothing, or false(by default), or true
  // @ViewChild(selector, { read: any, static: boolean }) classPropertyName: type;

  // You can read it in ngAfterViewInit
  ngAfterViewInit() {
    ...
  }
}
```

@ViewChildren

- Almost similar to @ViewChild, just for multiple references.
- Everything applicable to @ViewChild is applicable here as well.
- The type of variables declared with @ViewChildren is QueryList<ComponentType>
- Read more about @ViewChildren here.

@ViewChildren Examples

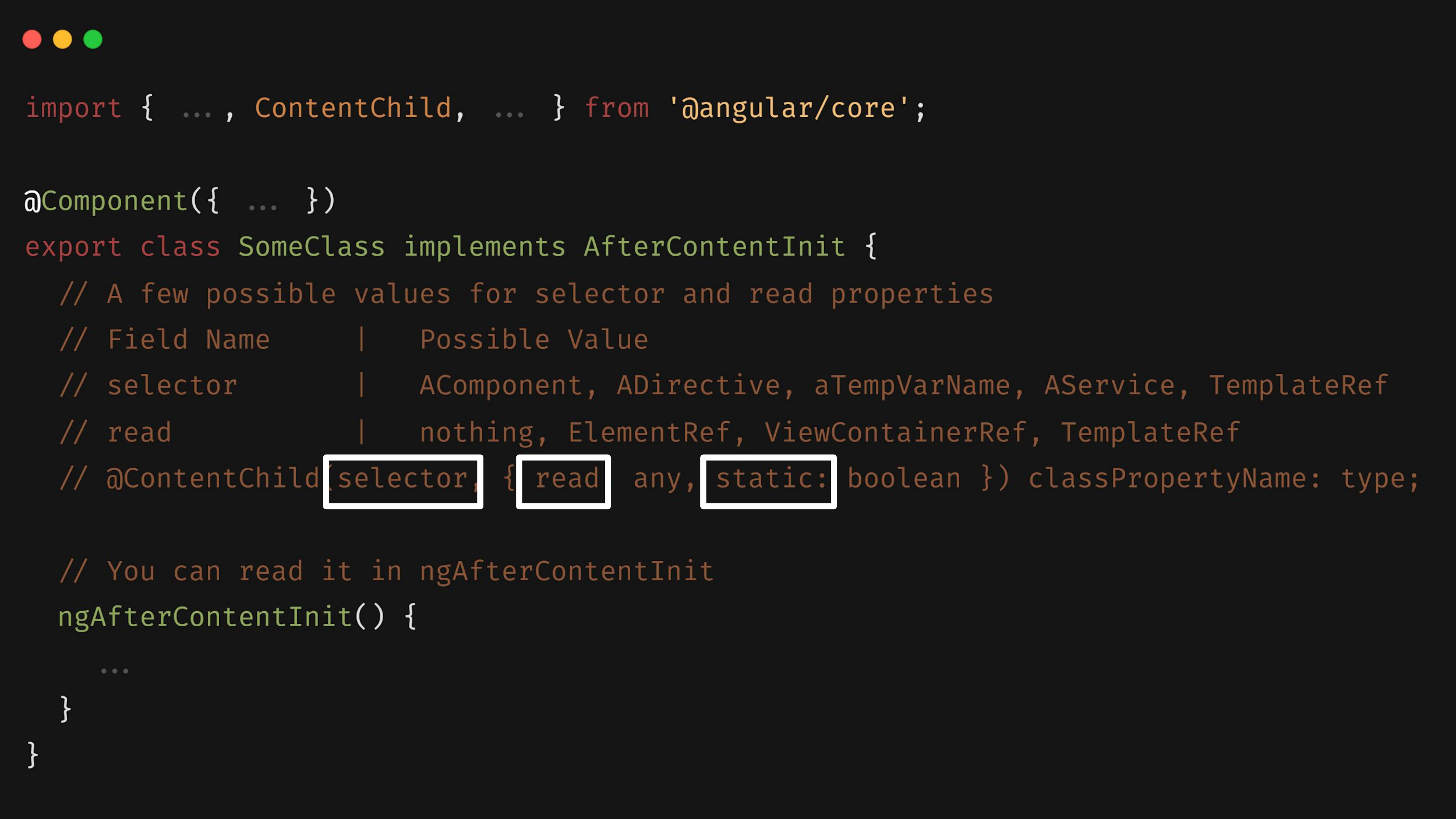


```
import { ..., ViewChildren, ... } from '@angular/core';

@Component({ ... })
export class SomeClass implements AfterViewInit {
  // A few possible values for selector and read properties
  // Field Name     |   Possible Value
  // selector       |   AComponent, ADirective, aTempVarName, AService, TemplateRef
  // read           |   nothing, ElementRef, ViewContainerRef, TemplateRef
  // @ViewChildren(selector, { read any }) classPropertyName: QueryList<type>;

  // You can read it in ngAfterViewInit
  ngAfterViewInit() {
    ...
  }
}
```

@ContentChild

- Get element reference from the Projected Content in the Component hosting it.
- Reference can be obtained via
  - Component/Directive Type or a Template Variable Name or by TemplateRef, or a Service name.
- By default gives:
  - Component instance if used with a Component.
  - ElementRef if used on an HTML Element.
- If read property is set, can give:
  - ElementRef or ViewContainerRef or TemplateRef
- Generally accessed in ngAfterContentInit.
- Read more about @ContentChild here.

@ContentChild Examples

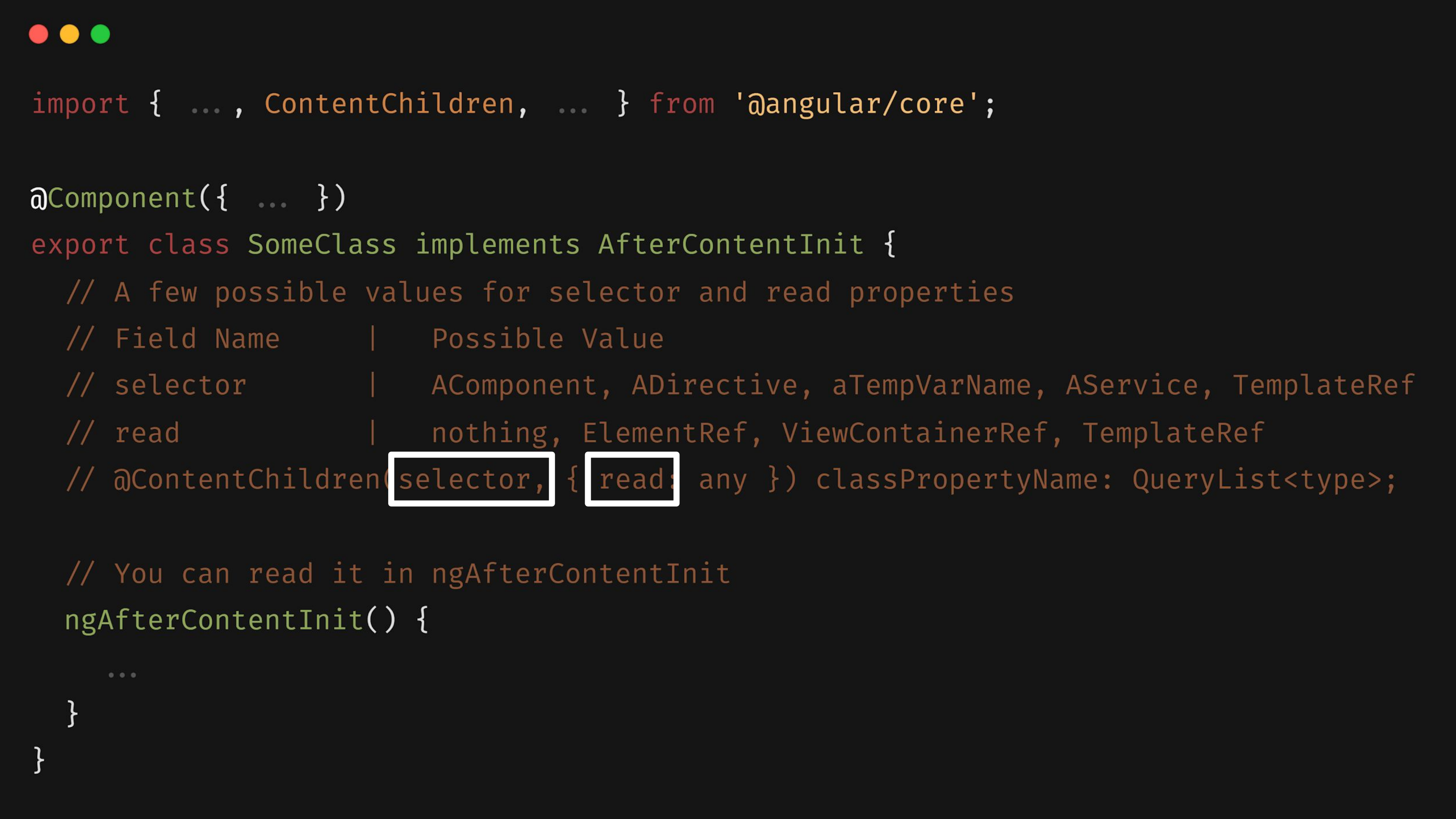


```
import { ..., ContentChild, ... } from '@angular/core';

@Component({ ... })
export class SomeClass implements AfterContentInit {
  // A few possible values for selector and read properties
  // Field Name     |   Possible Value
  // selector       |   AComponent, ADirective, aTempVarName, AService, TemplateRef
  // read           |   nothing, ElementRef, ViewContainerRef, TemplateRef
  // @ContentChild(selector, { read: any, static: boolean }) classPropertyName: type;

  // You can read it in ngAfterContentInit
  ngAfterContentInit() {
    ...
  }
}
```

@ContentChildren

- Almost similar to @ContentChild, just for multiple references.
- Everything applicable to @ContentChild is applicable here as well.
- The type of variables declared with @ContentChildren is QueryList<ComponentType>
- Read more about @ContentChildren here.

@ContentChildren Examples



```
import { ..., ContentChildren, ... } from '@angular/core';

@Component({ ... })
export class SomeClass implements AfterContentInit {
  // A few possible values for selector and read properties
  // Field Name     |   Possible Value
  // selector       |   AComponent, ADirective, aTempVarName, AService, TemplateRef
  // read           |   nothing, ElementRef, ViewContainerRef, TemplateRef
  // @ContentChildren(selector, { read: any }) classPropertyName: QueryList<type>;

  // You can read it in ngAfterContentInit
  ngAfterContentInit() {
    ...
  }
}
```

A way to share Data

- Since a Component can have access to a Child Component’s instance, data can be shared via it.
- An example Sample StackBlitz for reference.

Angular’s Change Detection Mechanism

Next Video

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash