

AfterContentChecked

ngAfterContentChecked é o quinto ciclo de vida executado pelo componente do Angular. Esse ciclo de vida tem um comportamento parecido com o ciclo de vida *ngDoCheck*, mas, no caso, ele verifica o conteúdo do ng-content. Ele é executado uma vez na criação do componente e também sempre que houver alguma alteração dentro do componente.

Para exemplo, vamos criar um componente principal e um componente secundário que ficará dentro do componente principal.

Componente secundário

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

@Component({
  selector: 'filho',
  templateUrl: 'filho.component.html',
  styleUrls: ['filho.component.css']
})
export class FilhoComponent {

  num: number = 1

  constructor() { }

  increment() {
    this.num++
  }

  decrement() {
    this.num--
  }
}
```

```
<p>Valor: {{ num }}</p>

<button (click)="increment()">+</button>
<button (click)="decrement()">-</button>
```

Componente Principal

```
import { AfterContentChecked, AfterContentInit, Component, ContentChild } from '@angular/core';
import { FilhoComponent } from '../filho/filho.component';

@Component({
  selector: 'main',
  templateUrl: './main.component.html',
  styleUrls: ['./main.component.css']
})
export class MainComponent implements AfterContentInit, AfterContentChecked {

  @ContentChild(FilhoComponent)
  el!: FilhoComponent

  pValue: number = 0
  cValue: number = 0

  constructor() { }

  ngAfterContentInit(): void {
    this.cValue = this.el.num
  }

  ngAfterContentChecked(): void {
    console.log('ngAfterContentChecked')

    if (this.el.num !== this.cValue) {
      this.pValue = this.cValue
      this.cValue = this.el.num
    }

    if (this.pValue > this.cValue) {
      console.log(`${this.pValue} é maior que ${this.cValue}`)
    } else if (this.pValue === this.cValue) {
      console.log(`${this.pValue} é igual a ${this.cValue}`)
    } else {
      console.log(`${this.pValue} é menor que ${this.cValue}`)
    }
  }
}
```

o *ContentChild*, assim como o *ViewChild*, também recupera elementos HTML dentro dele, mas, no caso dele, ele recupera elementos HTML do ng-content. Ele estará funcionando a partir do ciclo de vida ngAfterContentInit.

```
<ng-content></ng-content>
```