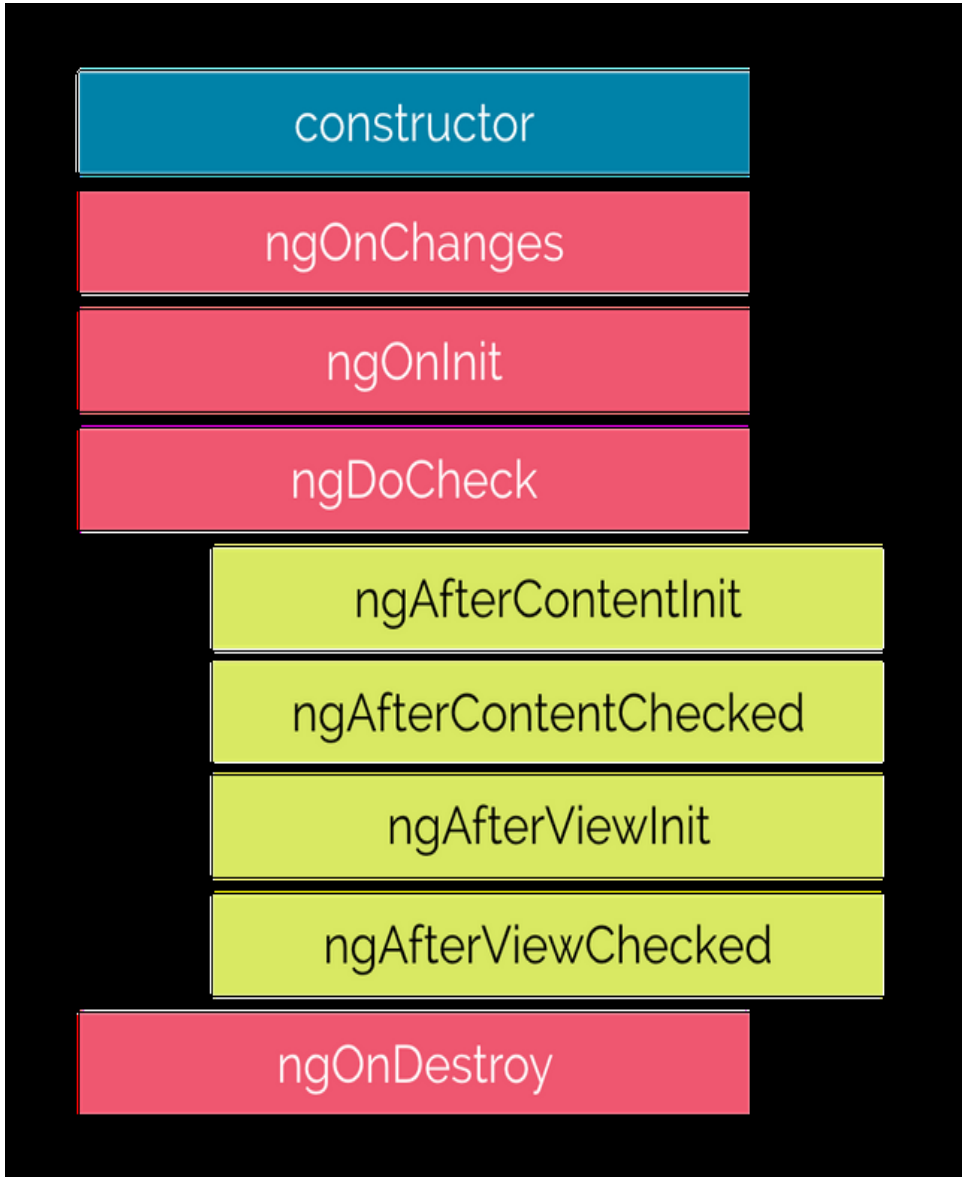
生命週期

生命週期介紹

生命週期是什麼？在 Angular 的每個元件都存在著生命週期（Lifecycle），從建立元件，檢測綁定屬性是否更新，到元件從 DOM 中銷毀，就構成一個元件的生命週期。
而 Angular 提供了八個 Lifecycle Hooks（生命週期鉤子），讓使用者能夠在各個階段對元件進行操作。
生命週期鉤子的執行順序如右圖所示。



Constructor 建構式

- constructor() 會在 class（類別）建立時最先被執行
- 是類別的屬性， Angular 並不能控制 constructor
- 當 constructor 執行時，元件尚未被初始化，因此幾乎不會在這個階段寫程式碼
- 主要用於相依注入（dependency injection），例如服務、函式或值

```
1 | constructor(private service: HeroService) { }
```

常用生命週期

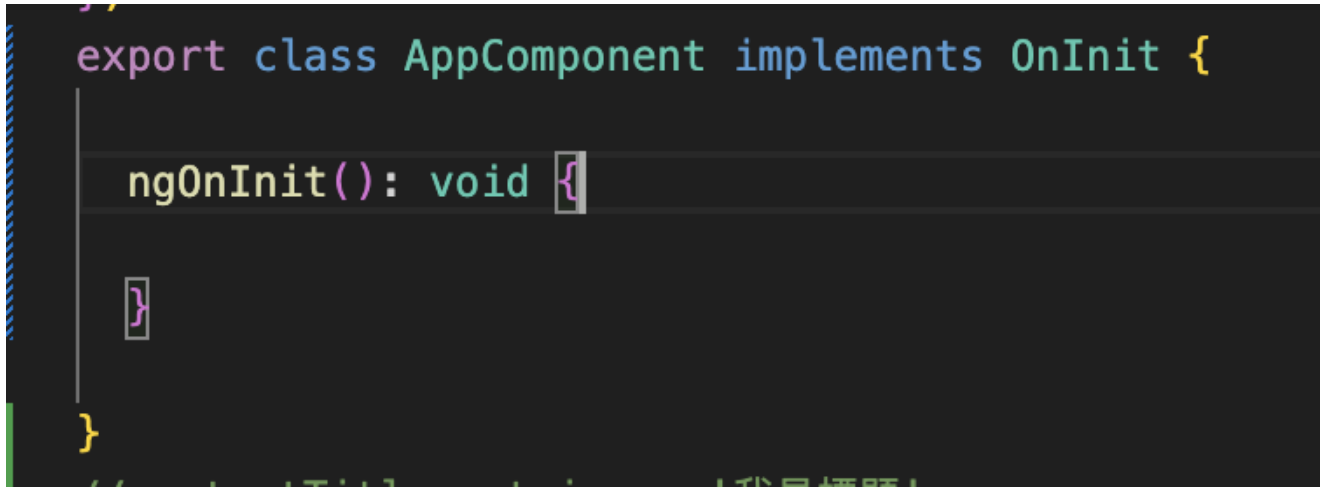
比較常使用到的生命週期是ngOnInit與ngAfterViewInit，
通常ngOnInit都是用來處理初始就要設定的程式碼因為不太會將程式碼寫在constructor中。
而ngAfterViewInit通常是使用在當你需要在一開始就去抓取畫面的屬性時，因為一開始如果畫面還沒loading好是沒辦法抓到屬性的內容的。
其他的生命週期就稍微看一下觸發的時間點用到的機率真的不高。

ngOnChanges

- 當元件 @Input/@Output 綁定的值發生變化時觸發
- 在 ngOnInit 之前執行
- 可能會執行多次

ngOnInit

- 元件初始的時候執行
- 只會執行一次



ngDoCheck

- 第一次ngOnInit之後接著執行，之後都緊接著 onChange。
- 偵測Angular 檢測變更檢測不到的部分，與onChange差異，當input傳入的值是像object的類型，值的參考位置不變onChange就不會被觸發；但docheck會。
- 可能會執行多次

```
})
export class AppComponent implements DoCheck {

  ngDoCheck(): void {

  }

}
```

ngAfterContentInit

- 當父元素的中的參考ng-content內容被初始化的時候(html template 產生時)
- 只會在首次do check發生之後執行
- 只執行一次

ngAfterContentChecked

- 完成 ng-content 的變更檢測調用(每次doCheck之後調用)。
- 可能會執行多次

```
})
export class AppComponent implements AfterContentChecked {

  ngAfterContentChecked(): void {

  }

}
// testTitle: string = '我是標題';
```

ngAfterViewInit

- 當組件的視圖被組裝完成後觸發。
- 只執行一次

```
export class AppComponent implements AfterViewInit {

  ngAfterViewInit(): void {

  }

}
```

ngAfterViewChecked

- 當組件的視圖更新後觸發。
- 可能會執行多次

```
export class AppComponent implements AfterViewChecked {

  ngAfterViewChecked(): void {

  }

}
```

ngOnDestory

- 組件被銷毀時觸發。
- 只執行一次

引用多個生命週期

你也可以一次使用多個生命週期的方法，寫法如下圖。

```
export class AppComponent implements OnDestroy,OnChanges {

  ngOnDestroy(): void {

  }

  ngOnChanges(): void {

  }

}
```

结束