變數

宣告變數

當你的畫面上需要新增一個值而這個值並不是固定的，而是會根據使用者的操作而去修改的。

那你就會需要先在該Html的Ts檔案中新增一個全域變數就可以讓Html去串接Ts中的資料並且展現在畫面中。

宣告完你會發現奇怪為什麼testContent是紅色的？讓我們接著看下去。

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

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet],
  templateUrl: './app.component.html',
  styleUrl: './app.component.scss'
})
export class AppComponent {
  testTitle: string = '我是標題';
  testContent: string;
}
```

宣告變數

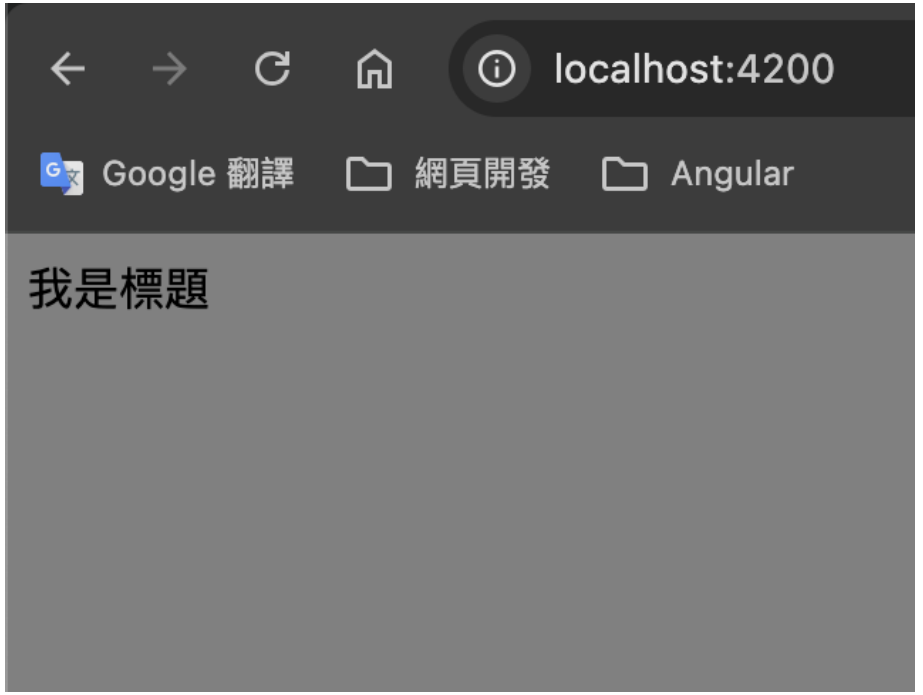
把你的滑鼠移動過去有紅色錯誤的那邊，他就會告訴你錯誤是什麼。
這時候你就可以幫他設定一個初始的值(或者是null或underfind)，這時候有人就會說了，可是真的不需要他有初始值，那就寫成這樣
testContent?: string;
這段的?就是告訴系統，我這個變數可能是有值也有可能是沒有值，系統就會跳過檢查這個變數。

畫面呈現

全域變數宣告完後我們就要讓Html有辦法呈現我們在他的Ts中宣告的變數，來到他的Html畫面並且用這個符號將你的變數名稱包起來{{ 變數名稱 }}，你就會發現畫面呈現出變數的內容了。

```
})
export class AppComponent {
  testTitle: string = '我是標題';
  testContent?: string;
}
```

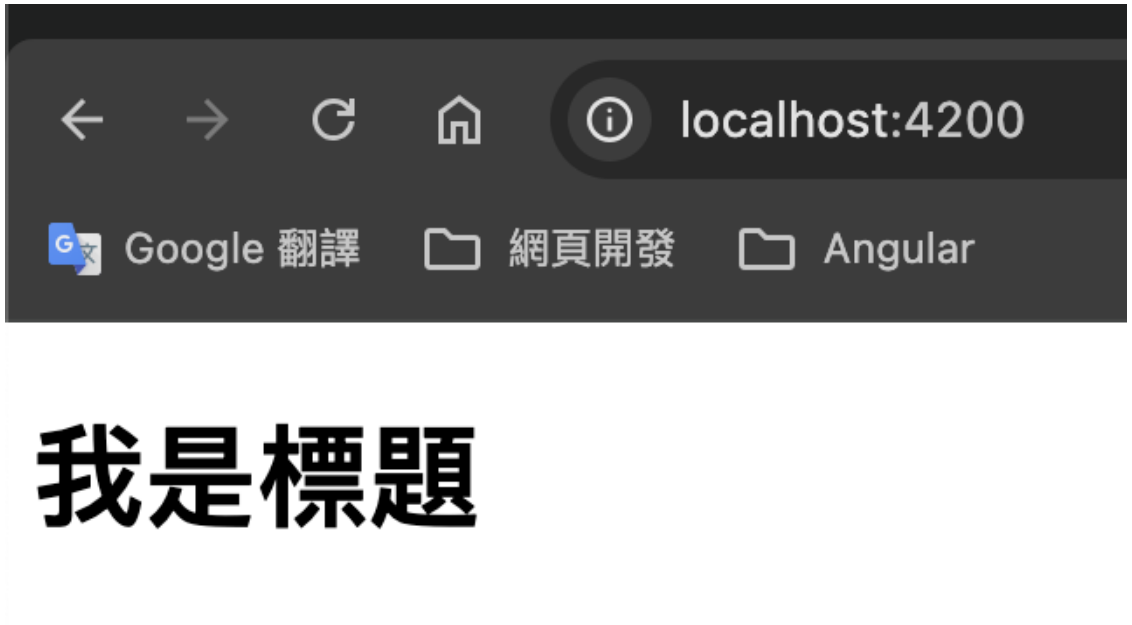
```
demo3 > src > app > <> app.component.html > ...
1  {{ testTitle }}
2
3
```



畫面呈現

當然顯示的變數也可以使用Html的標籤包起來使用，用來取代原本固定的值。

```
demo3 > src > app > app.component.html > h1
1   <h1>{{ testTitle }}</h1>
2
3
```



練習

Html全部都使用變數來去顯示，圖片內容也是。標題<h1>內容<h3>圖片<img>

我是標題

內容是我

全域變數

當你需要一個值要讓整個TS都可以使用到並且值不會消失，那你就需要新增一個全域變數的全域變數，如圖所示我們宣告了一個title，當我們在下面的方法需要取得這個變數的值的時候，我們就要使用this.去呼叫這個變數。

```
export class AppComponent {
  title = 'demo2';

  shwoTitle() {
    alert(this.title);
  }
}
```

區域變數

當你需要一個值只要在某個方法中使用，並且使用完之後不需要他繼續存在記憶體中時，那你就可以宣告一個區域變數，如下圖我們在一個方法中需要顯示一段固定的文字，當這個方法執行玩這個title就會消失，也無法在這個TS中去呼叫他。

區域變數

當你宣告區域變數時會遇到一個問題是我該使用let、var或者const?

- const
  一般使用在識別值(identifier)不會被重新指定值，簡單來說你宣告之後就不能修改它了。
- let
  一般使用在變數(variable)可能會被重新指定值，現在最常使用到的宣告方法。
- var
  最弱的宣告方法，現在大部份不會使用。

let VS var

let 跟 var其實是差不多的，但為什麼現在大家都使用let而不使用var?
主要是因為var不夠嚴謹，寫code的習慣是我要使用這個變數時我需要先宣告這個變數後面才能使用，這樣才可以知道這個變數的類型或者其預設值，但var卻可以寫在隨意的地方，這樣導致對值得理解很容易出錯。

```
shwoTitle() {
  var title = "demo2";
  alert(title);
}
}
```

```
shwoTitle() {
  title = "demo";
  alert(title);
  var title: string;
}
}
```

结束