Angular Directive

523419

(Advanced Web Application Development)

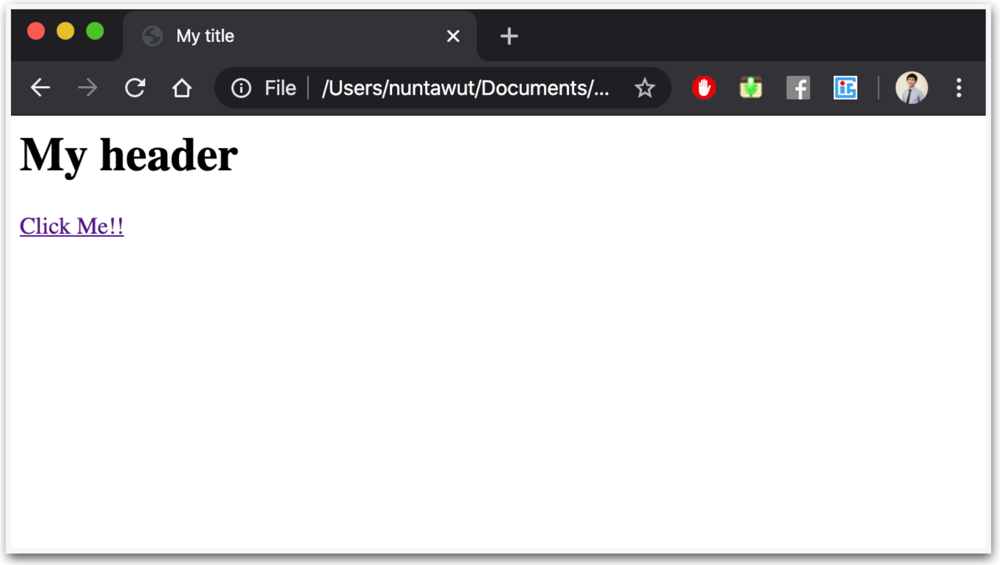
Dr. Nuntawut Kaoungku
Assistant Professor of Computer Engineering

What is the DOM?

- The HTML DOM defines a standard way for accessing and manipulating HTML documents.
- It presents an HTML document as a tree-structure with elements, attributes, and text.
- With JavaScript you can restructure an entire HTML document. You can add, remove, change, or reorder items on a page.
- This access, along with methods and properties to add, move, change, or remove HTML elements, is given through DOM.
- The DOM can be used by JavaScript to read and change HTML, XHTML, and XML documents.

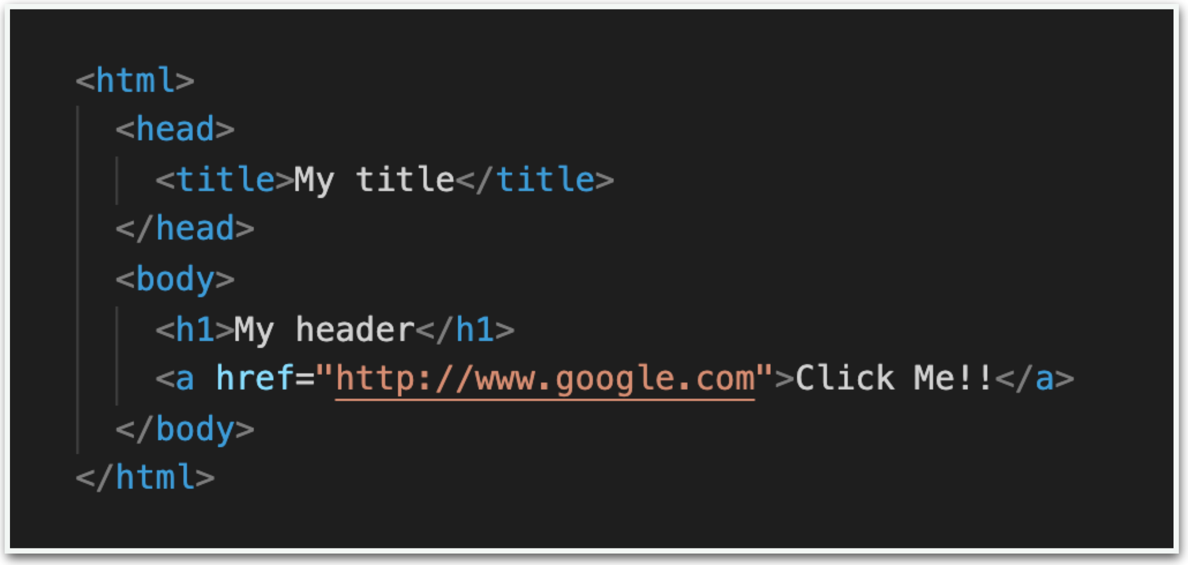
Document Tree

- A simple way to think of the DOM is in terms of the document tree.



My header

Click Me!!

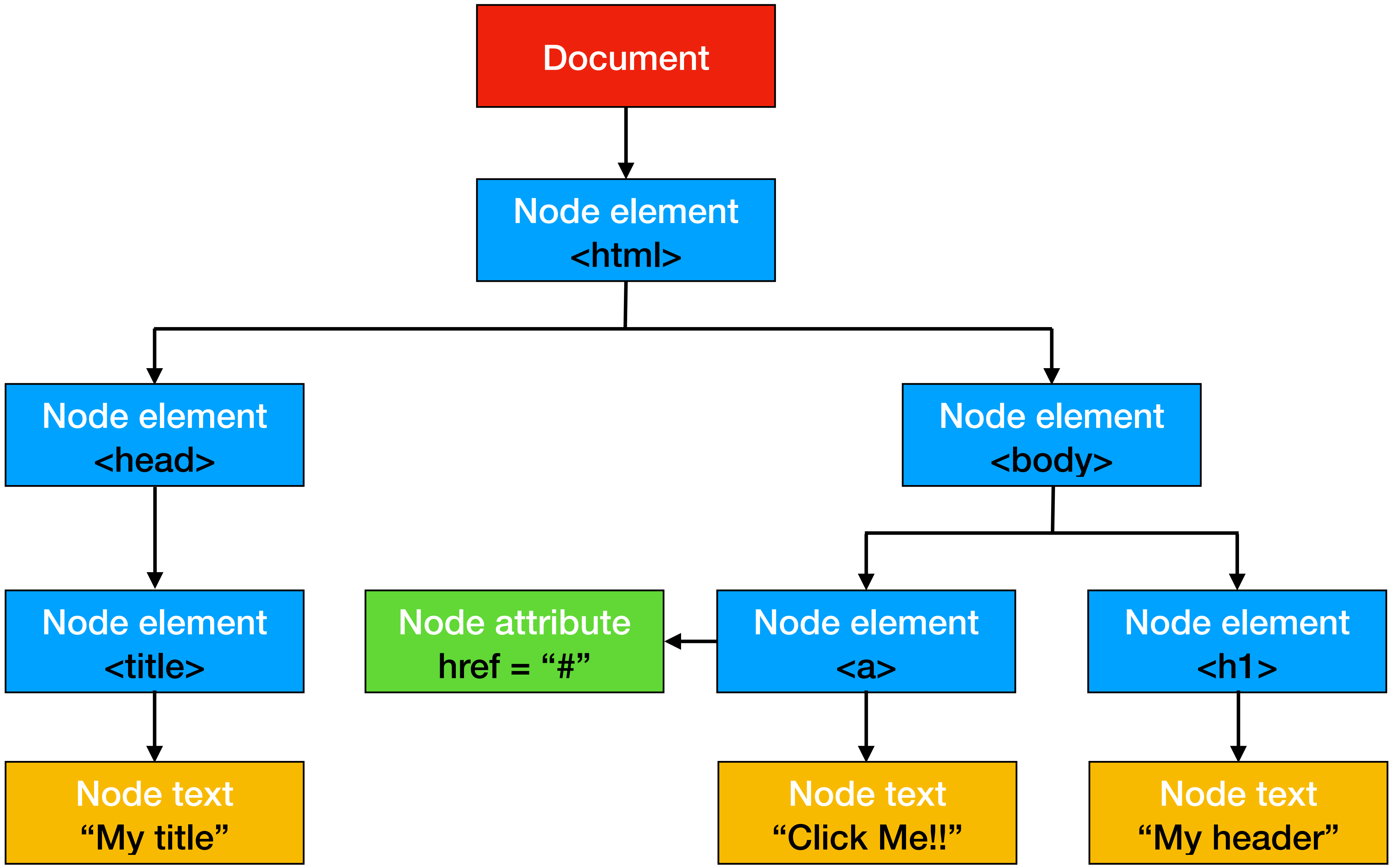


```
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <h1>My header</h1>
    <a href="http://www.google.com">Click Me!!</a>
  </body>
</html>
```

Display

.html

Document Tree (contd.)



Accessing DOM nodes

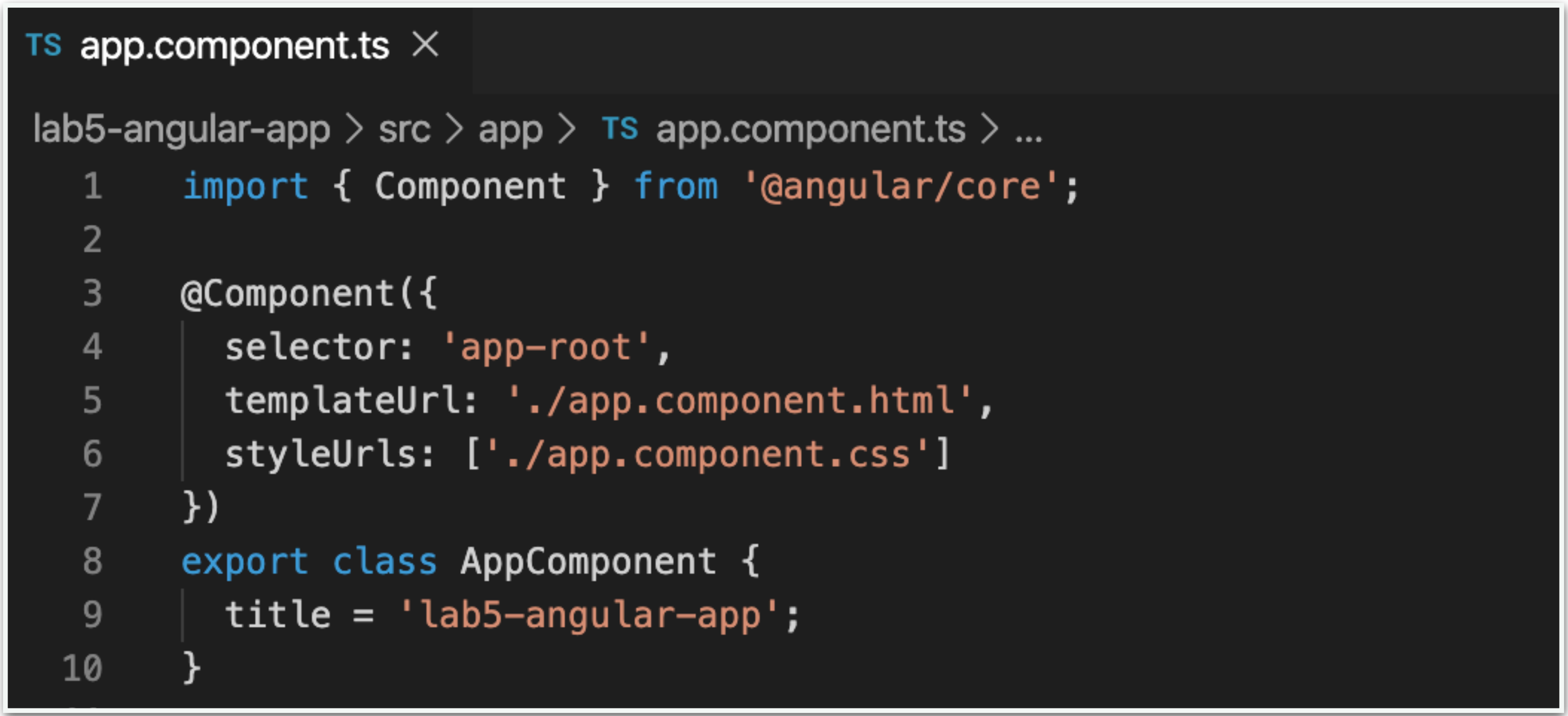
- The getElementById() method returns the element with the specified ID:
    - document.getElementById(“someID”);
- The getElementsByTagName() method returns all elements (as a nodeList) with the specified tag name.
    - document.getElementsByTagName(“a”);

Directive Introduction

- **Components**
  - Directives with a template
- **Structure Directive**
  - Adds and removes DOM elements to change DOM layout
- **Attribute Directive**
  - Changes the appearance or behaviour of an element

Component Directive

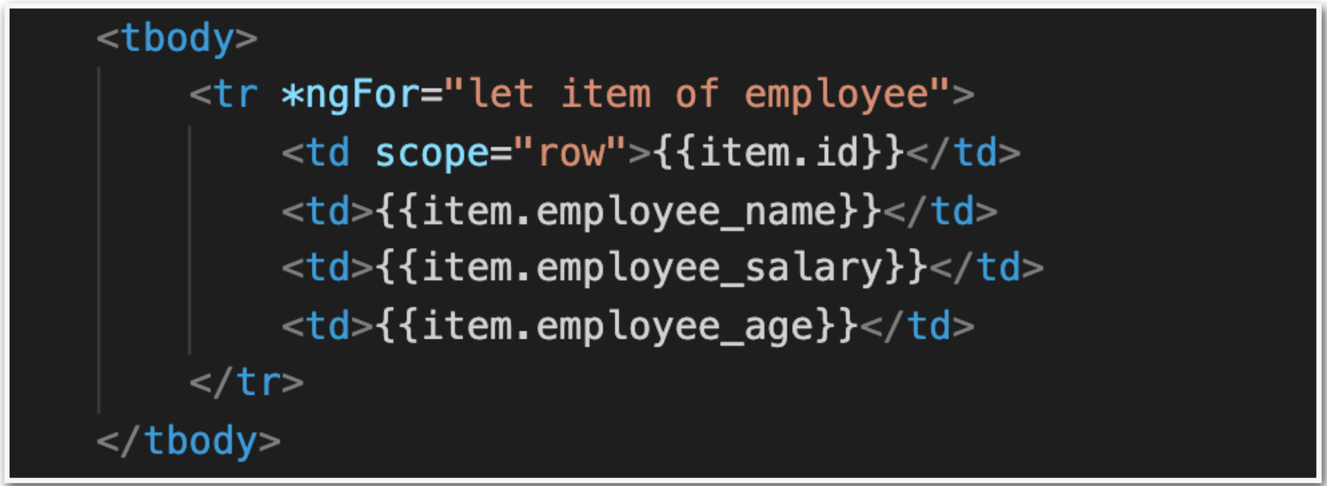
- A component a technically a directive-with-a-template
- @Component decorator is a @Directive decorator extended with template-oriented features



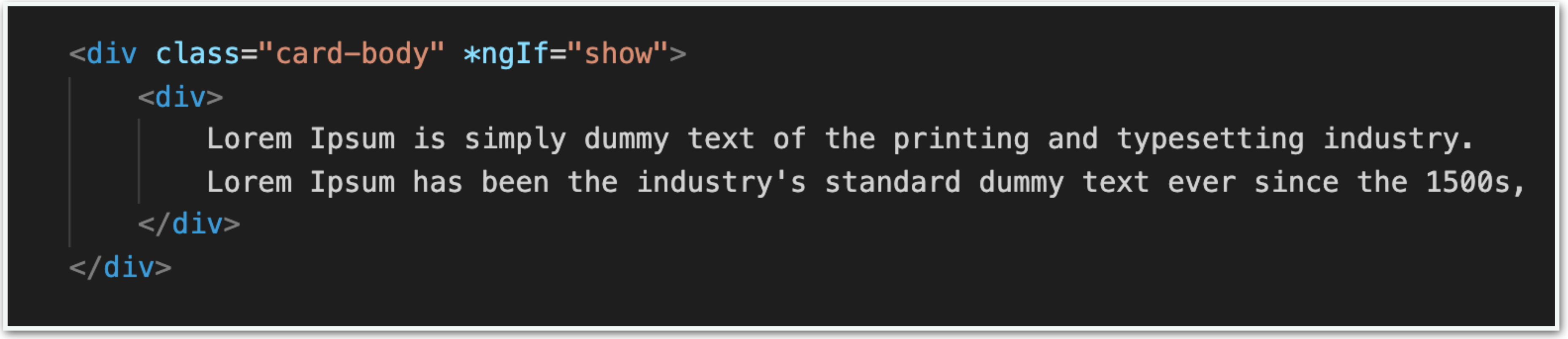
```typescript
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'lab5-angular-app';
}
```

Structural Directives

- Structural directives alter layout by adding, removing, and replacing elements in DOM



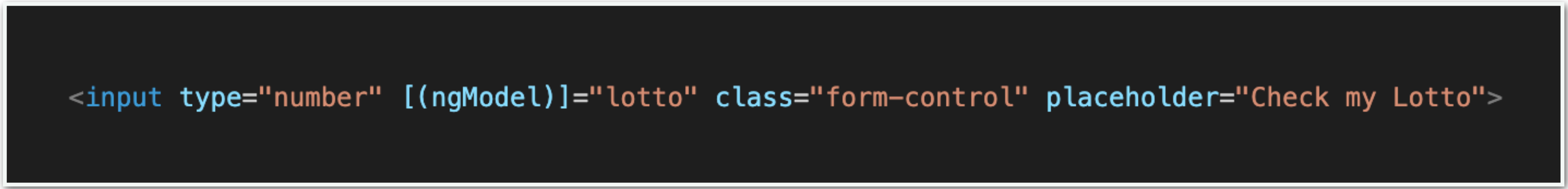
```
<tbody>
    <tr *ngFor="let item of employee">
        <td scope="row">{{item.id}}</td>
        <td>{{item.employee_name}}</td>
        <td>{{item.employee_salary}}</td>
        <td>{{item.employee_age}}</td>
    </tr>
</tbody>
```



```
<div class="card-body" *ngIf="show">
    <div>
        Lorem Ipsum is simply dummy text of the printing and typesetting industry.
        Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    </div>
</div>
```

Attribute Directive

- Attribute directive alter the appearance or behaviour of an existing element
- ngModel modifies the behaviour of an existing element
- Displays value property and responds to changing events



```
<input type="number" [(ngModel)]="lotto" class="form-control" placeholder="Check my Lotto">
```

Built-in Angular Directive

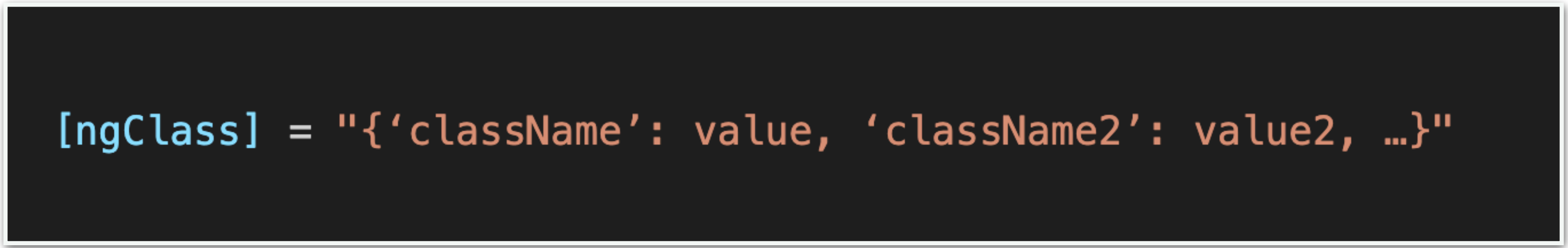
- ***ngClass*** : Adds and removes CSS classes on an HTML element
- ***ngStyle*** : An <u>attribute directive</u> that updates styles for the containing HTML element
- ***ngIf*** : A <u>structural directive</u> that conditionally includes a template based on the value of an expression coerced to Boolean
- ***ngSwitch*** : A <u>structural directive</u> that adds or removes templates (displaying or hiding views) when the next match expression matches the switch expression
- ***ngFor*** : A <u>structural directive</u> that renders a template for each item in a collection. The directive is placed on an element, which becomes the parent of the cloned templates.

ngClass

- ngClass is used for class binding - adding or removing several classes

- Adding an ngClass property binding sets the element’s classes accordingly

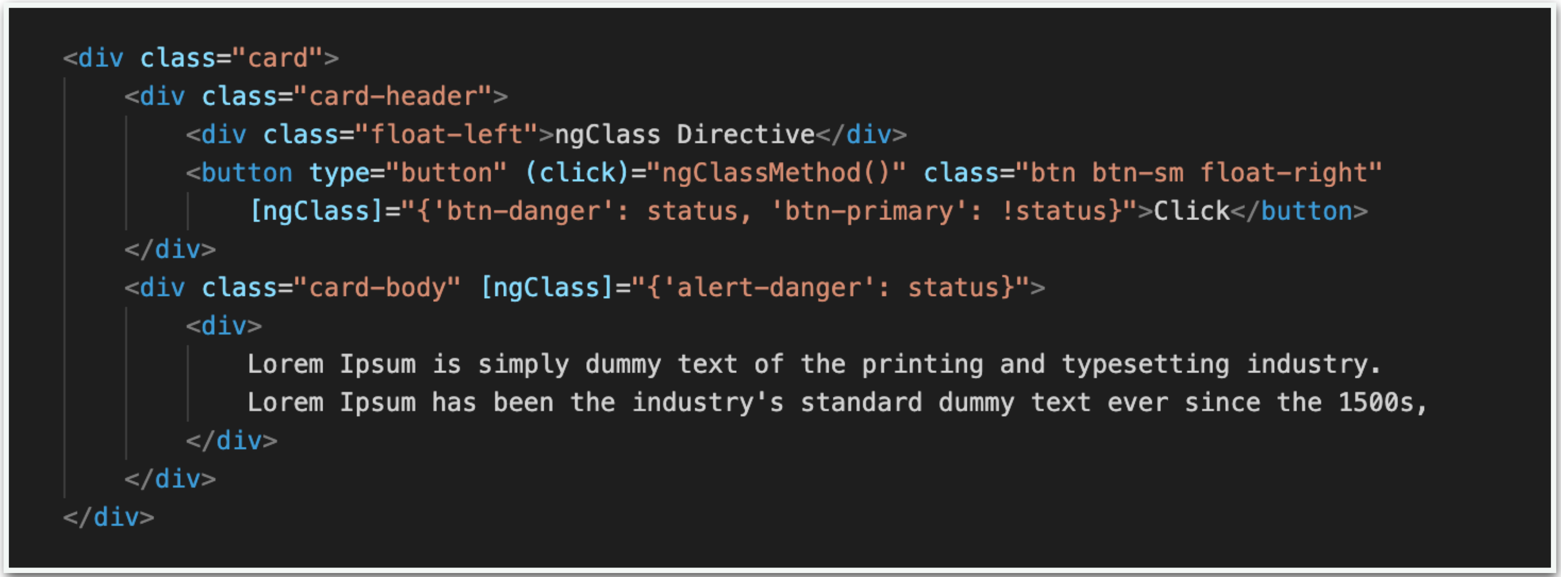
- Pattern :



```
[ngClass] = "{‘className’: value, ‘className2’: value2, …}"
```

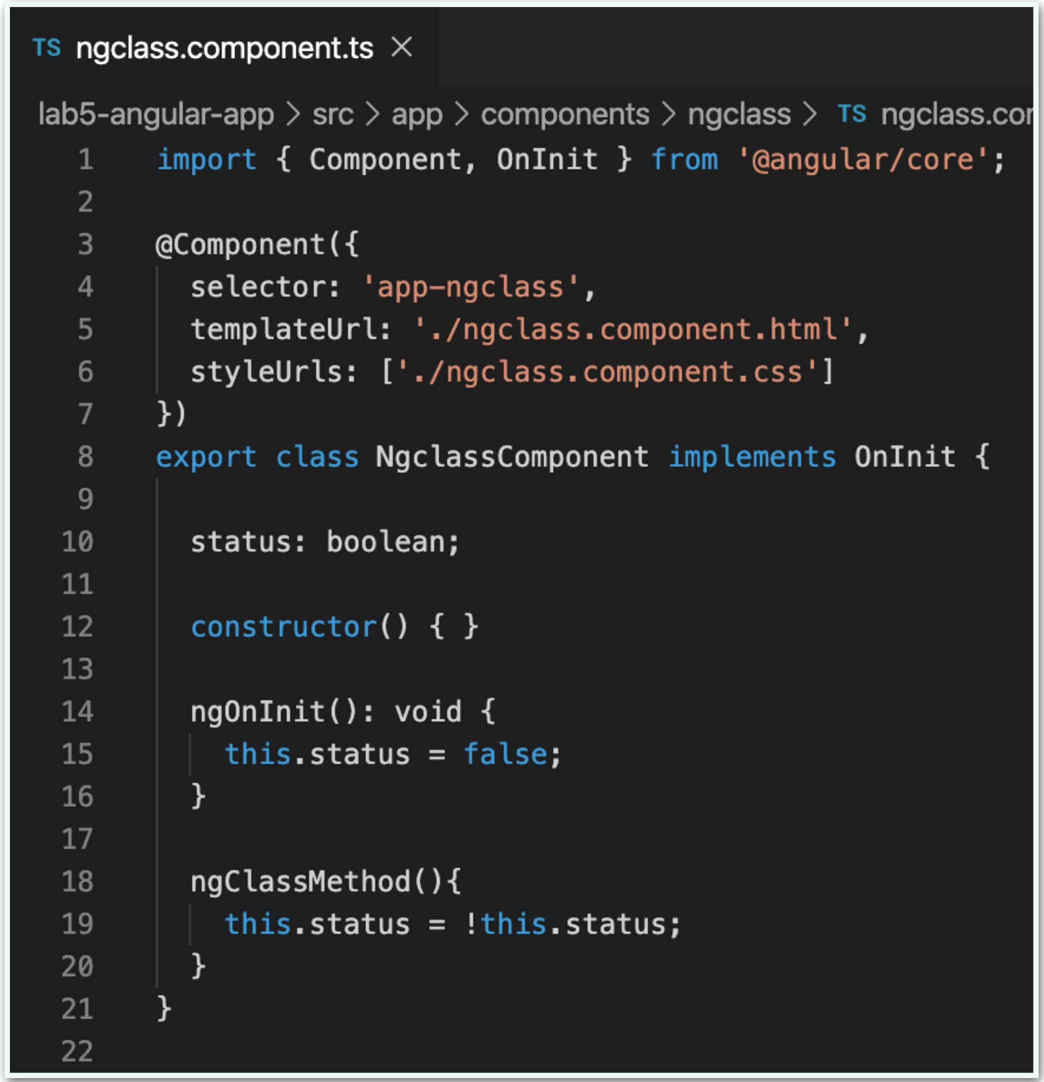
Example: ngClass

Template



```html
<div class="card">
    <div class="card-header">
        <div class="float-left">ngClass Directive</div>
        <button type="button" (click)="ngClassMethod()" class="btn btn-sm float-right"
            [ngClass]="{'btn-danger': status, 'btn-primary': !status}">Click</button>
    </div>
    <div class="card-body" [ngClass]="{'alert-danger': status}">
        <div>
            Lorem Ipsum is simply dummy text of the printing and typesetting industry.
            Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
        </div>
    </div>
</div>
```

Controller

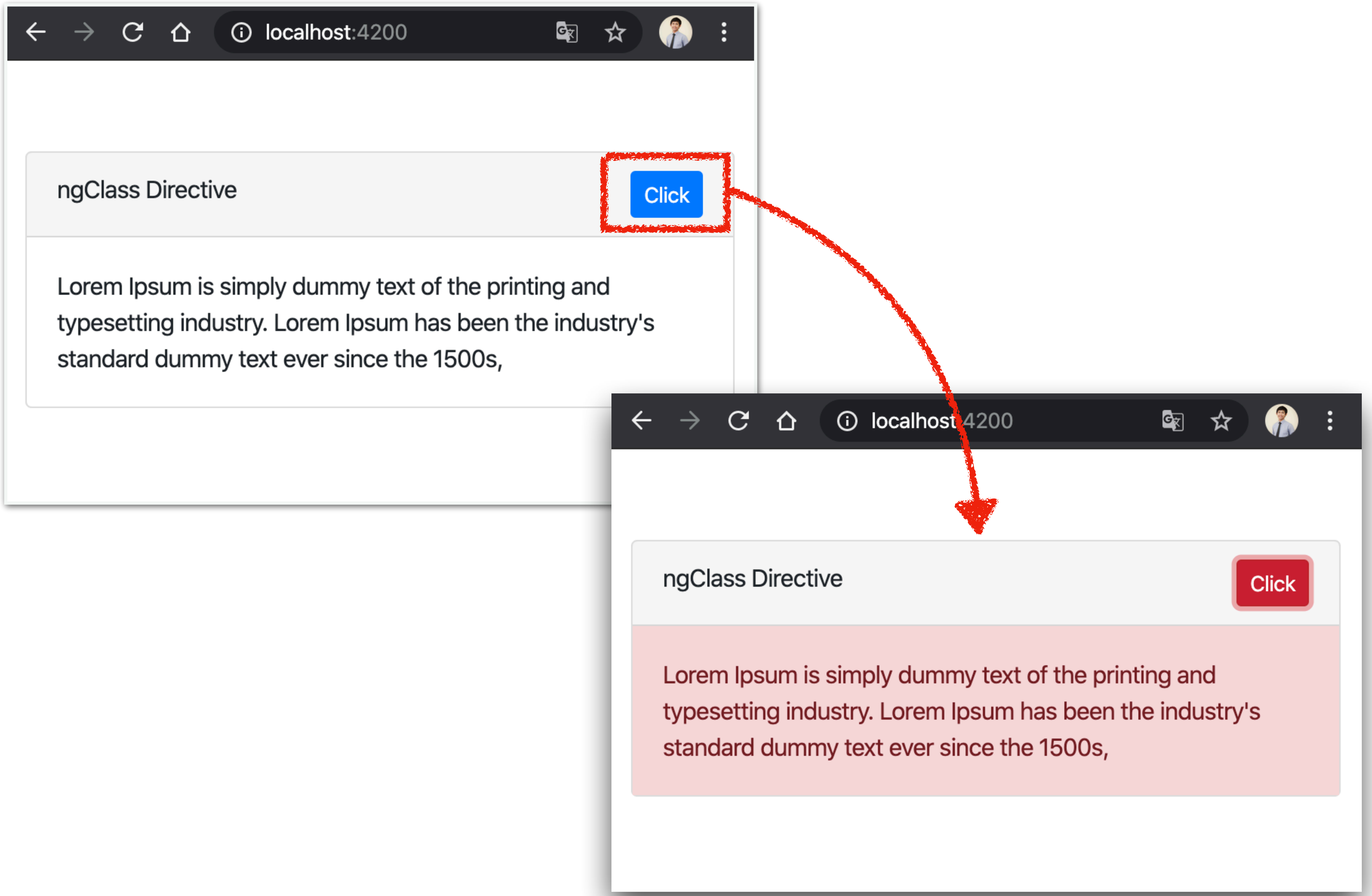


TS ngclass.component.ts ×

lab5-angular-app > src > app > components > ngclass > TS ngclass.co

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

@Component({
  selector: 'app-ngclass',
  templateUrl: './ngclass.component.html',
  styleUrls: ['./ngclass.component.css']
})
export class NgclassComponent implements OnInit {

  status: boolean;

  constructor() { }

  ngOnInit(): void {
    this.status = false;
  }

  ngClassMethod(){
    this.status = !this.status;
  }
}

```

Example: ngClass


localhost:4200

ngClass Directive

Click

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,

localhost:4200

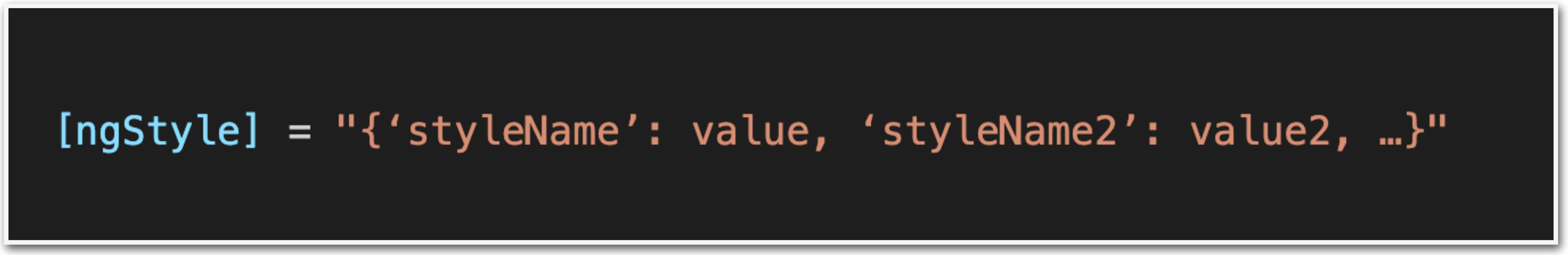
ngClass Directive

Click

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,

ngStyle

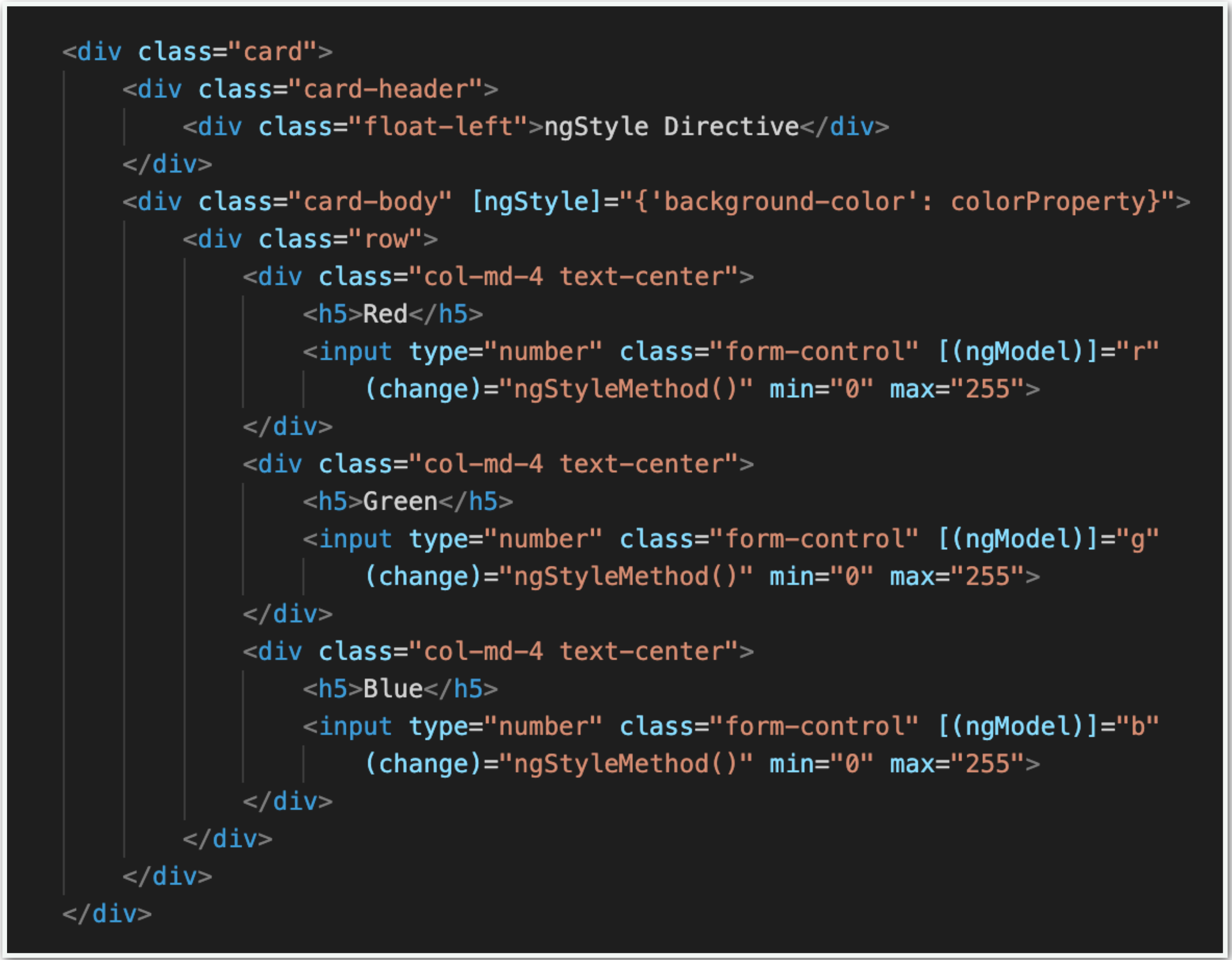
- ngStyle helps in style binding
- ngStyle directive is a better choice while setting many inline styles
- Pattern :



```
[ngStyle] = "{‘styleName’: value, ‘styleName2’: value2, …}"
```

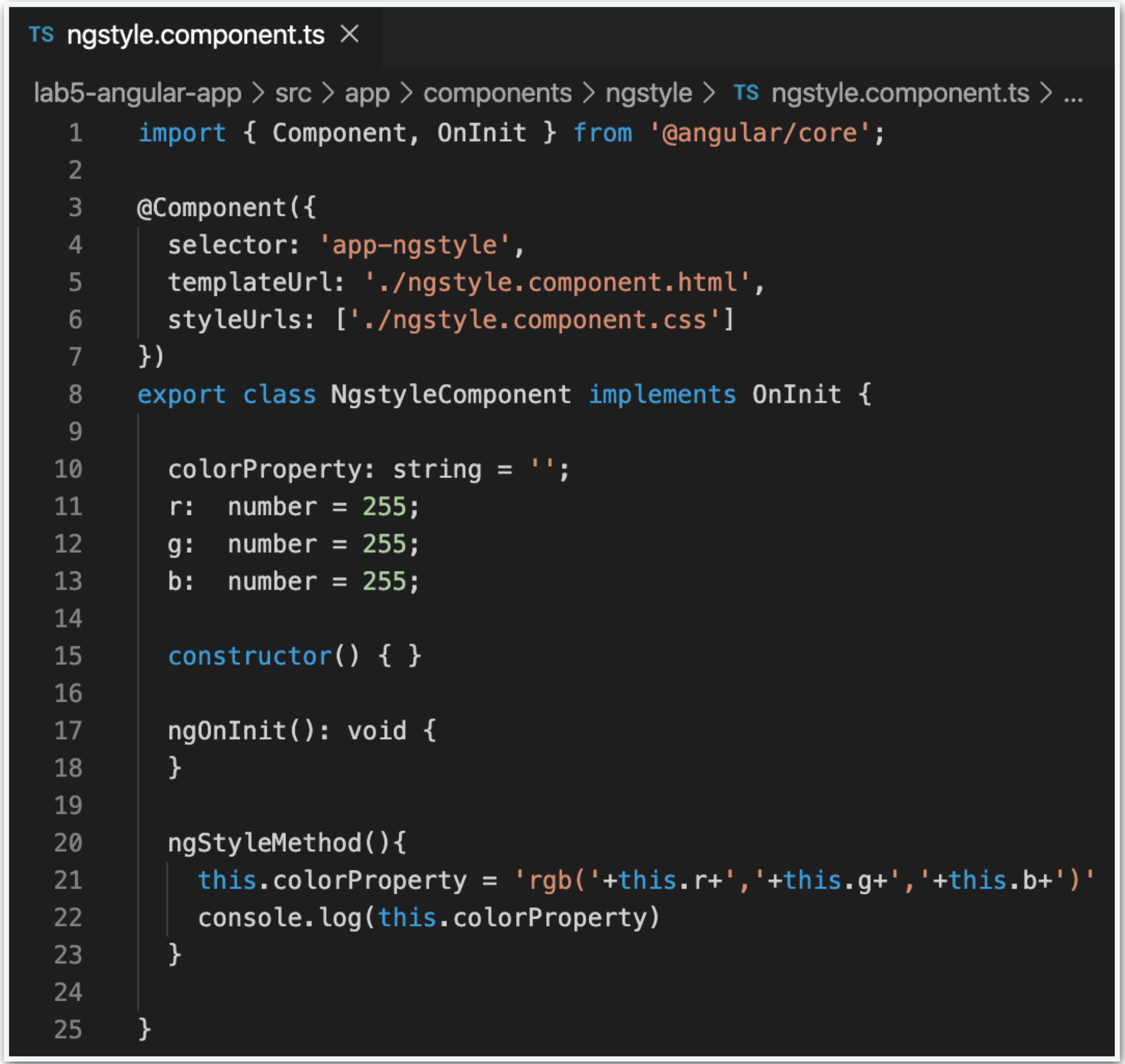
Example: ngStyle

Template



```html
<div class="card">
    <div class="card-header">
        <div class="float-left">ngStyle Directive</div>
    </div>
    <div class="card-body" [ngStyle]="{'background-color': colorProperty}">
        <div class="row">
            <div class="col-md-4 text-center">
                <h5>Red</h5>
                <input type="number" class="form-control" [(ngModel)]="r"
                    (change)="ngStyleMethod()" min="0" max="255">
            </div>
            <div class="col-md-4 text-center">
                <h5>Green</h5>
                <input type="number" class="form-control" [(ngModel)]="g"
                    (change)="ngStyleMethod()" min="0" max="255">
            </div>
            <div class="col-md-4 text-center">
                <h5>Blue</h5>
                <input type="number" class="form-control" [(ngModel)]="b"
                    (change)="ngStyleMethod()" min="0" max="255">
            </div>
        </div>
    </div>
</div>
```

Controller

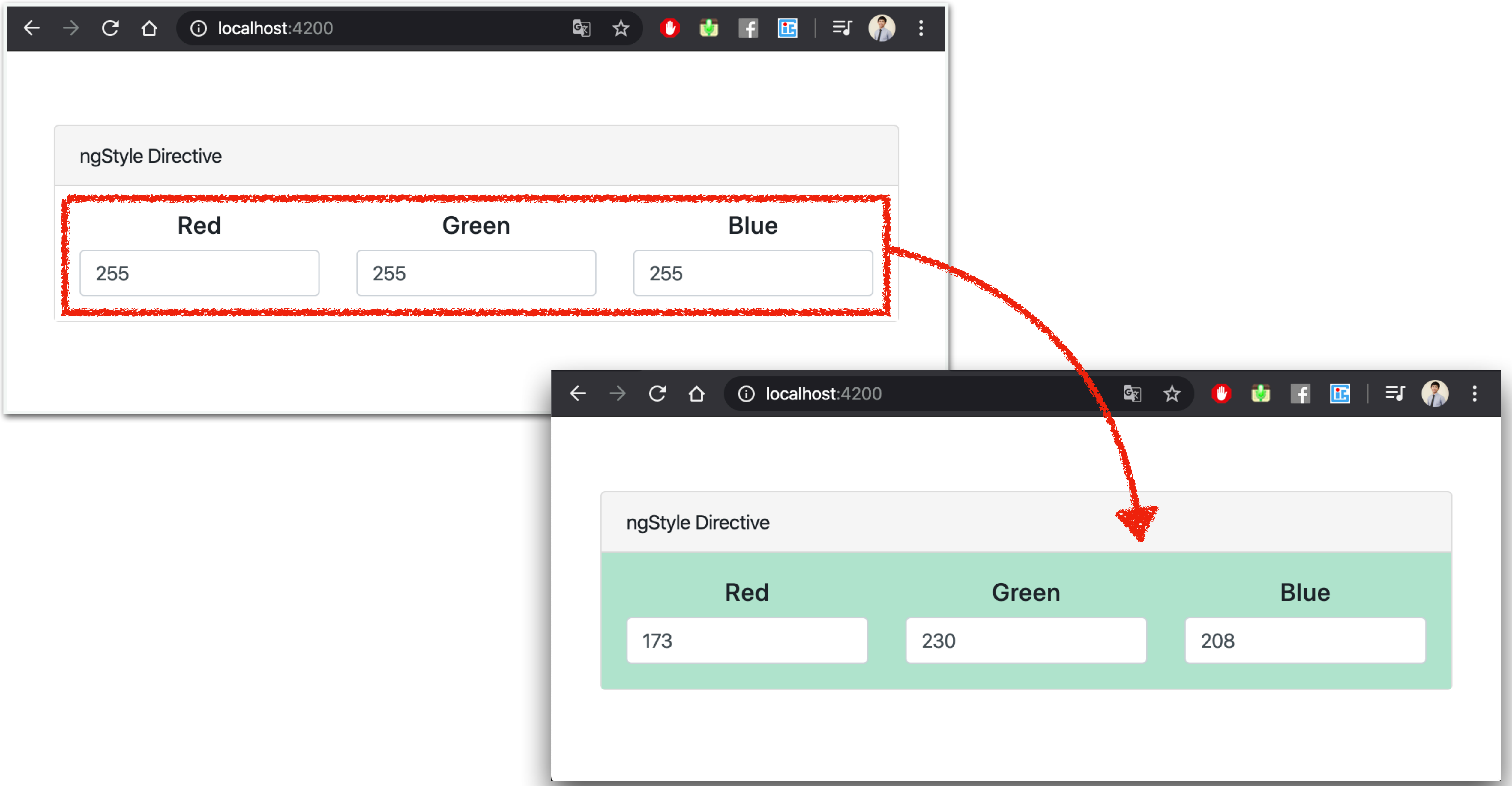


TS ngstyle.component.ts ×

lab5-angular-app > src > app > components > ngstyle > TS ngstyle.component.ts > ...

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

@Component({
  selector: 'app-ngstyle',
  templateUrl: './ngstyle.component.html',
  styleUrls: ['./ngstyle.component.css']
})
export class NgstyleComponent implements OnInit {

  colorProperty: string = '';
  r:  number = 255;
  g:  number = 255;
  b:  number = 255;

  constructor() { }

  ngOnInit(): void {
  }

  ngStyleMethod(){
    this.colorProperty = 'rgb('+this.r+','+this.g+','+this.b+')'
    console.log(this.colorProperty)
  }

}
```

Example: ngStyle



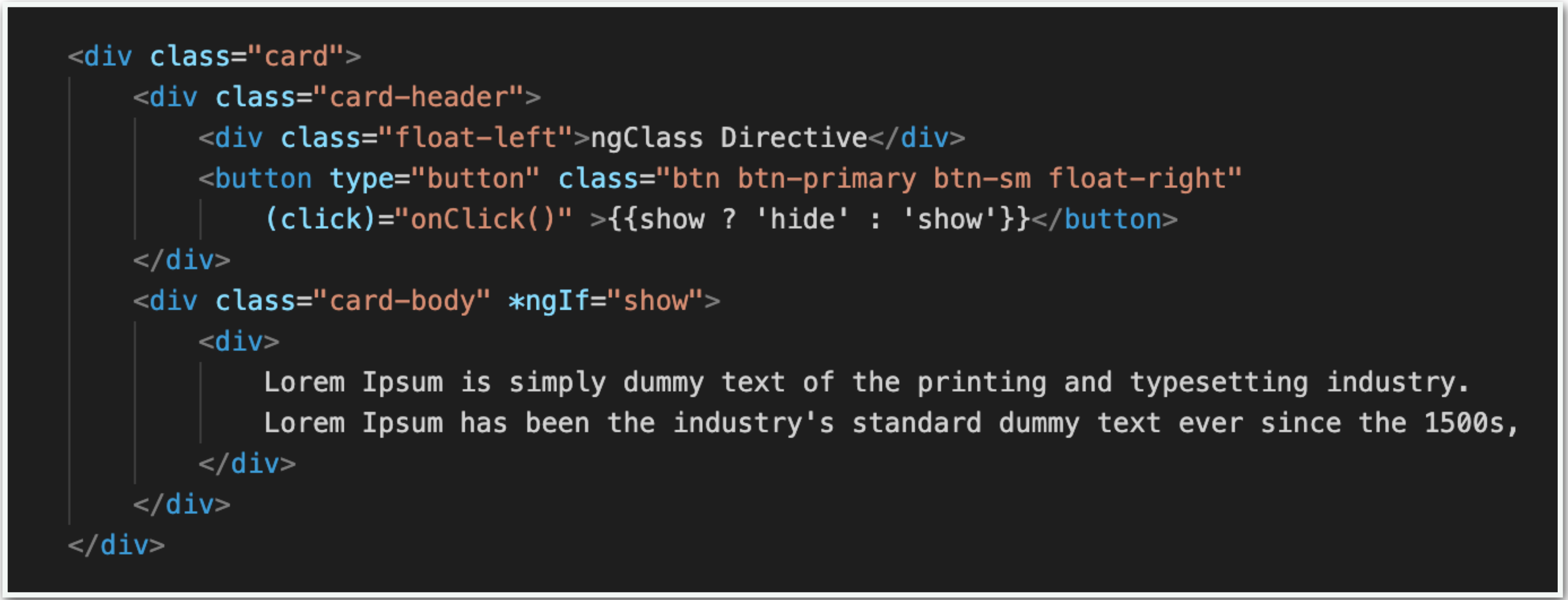
ngIf

- Adds an element subtree to the DOM by binding an ngIf directive
- Binding to false expression removes the element subtree from the DOM
- Pattern:

```
*ngIf="expression"
```

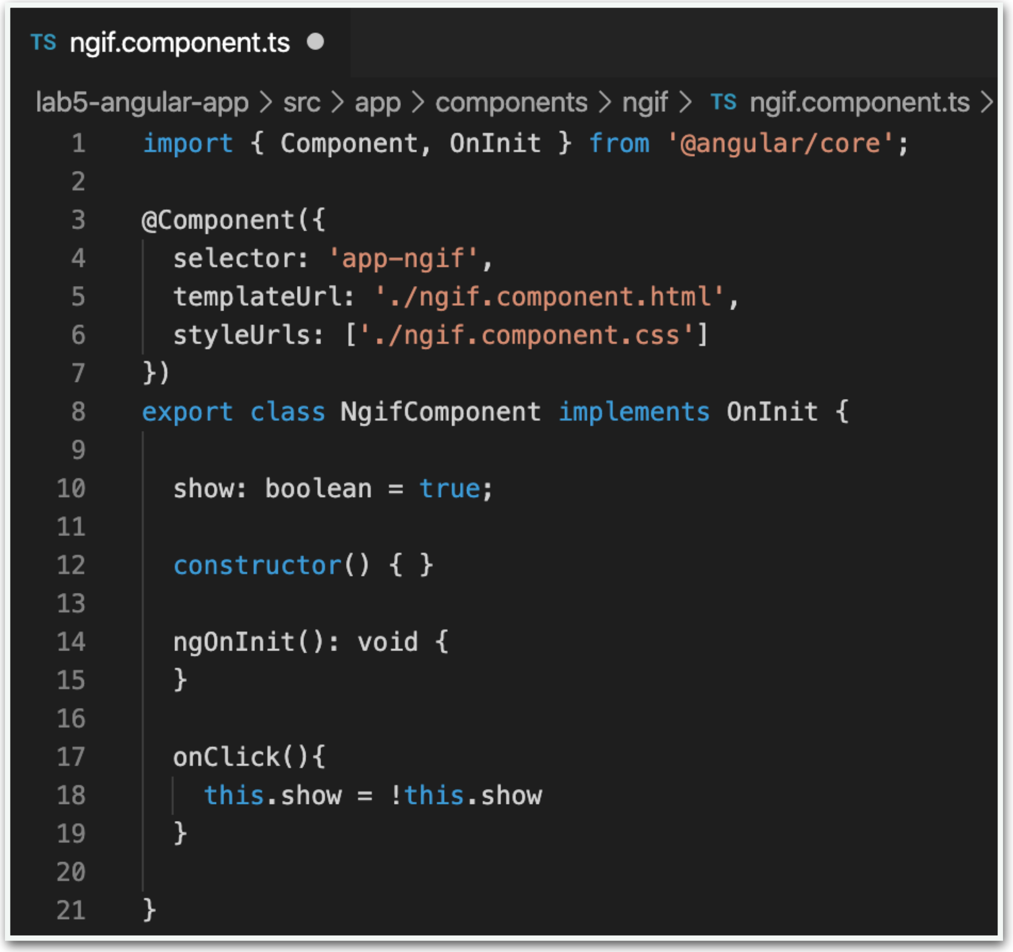
Example: ngIf

Template



```html
<div class="card">
    <div class="card-header">
        <div class="float-left">ngClass Directive</div>
        <button type="button" class="btn btn-primary btn-sm float-right"
            (click)="onClick()" >{{show ? 'hide' : 'show'}}</button>
    </div>
    <div class="card-body" *ngIf="show">
        <div>
            Lorem Ipsum is simply dummy text of the printing and typesetting industry.
            Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
        </div>
    </div>
</div>
```

Controller

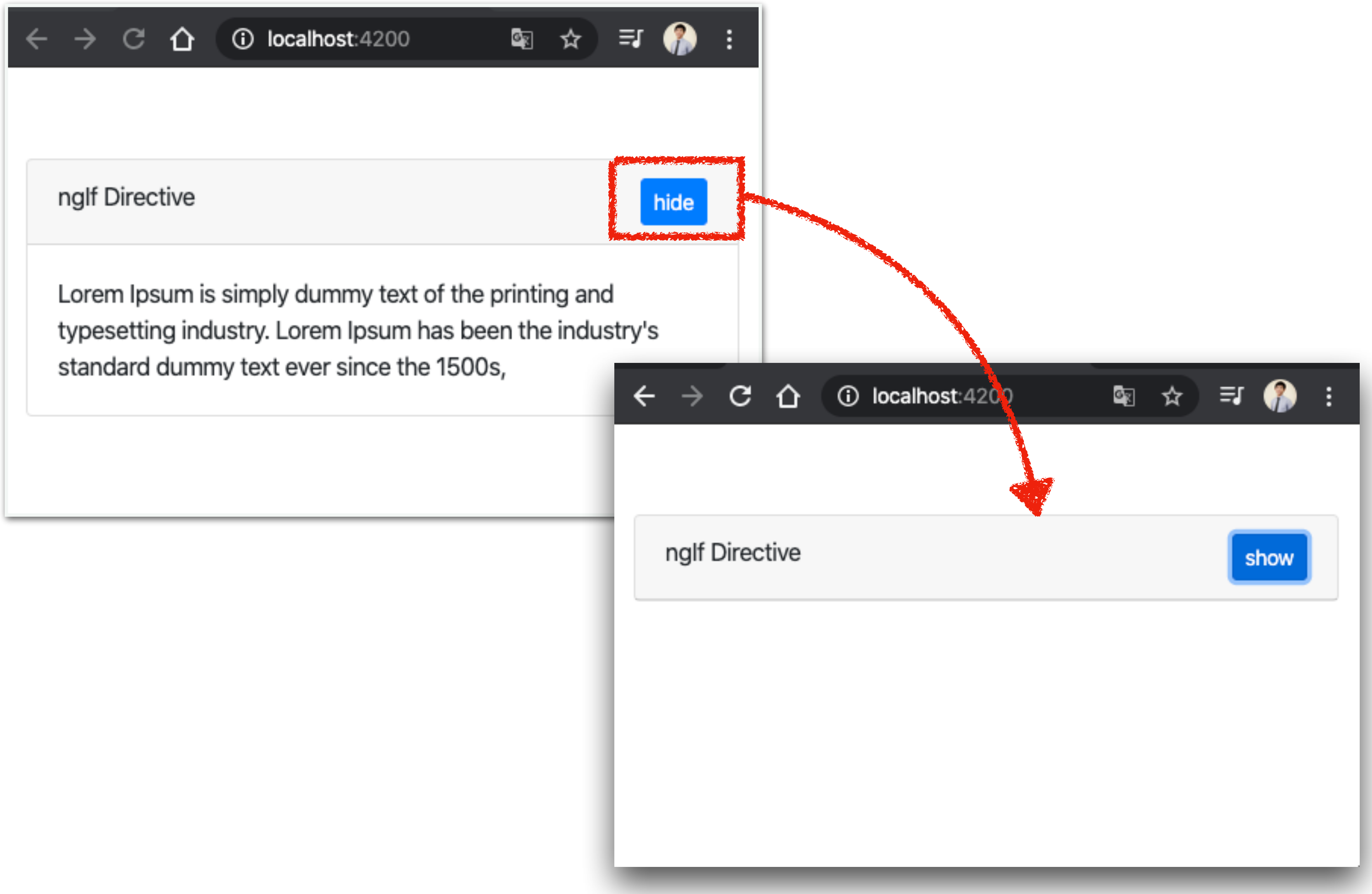


TS ngif.component.ts ●

lab5-angular-app > src > app > components > ngif > TS ngif.component.ts >

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

@Component({
  selector: 'app-ngif',
  templateUrl: './ngif.component.html',
  styleUrls: ['./ngif.component.css']
})
export class NgifComponent implements OnInit {

  show: boolean = true;

  constructor() { }

  ngOnInit(): void {
  }

  onClick(){
    this.show = !this.show
  }

}
```

Example: ngIf



ngIf Directive

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,

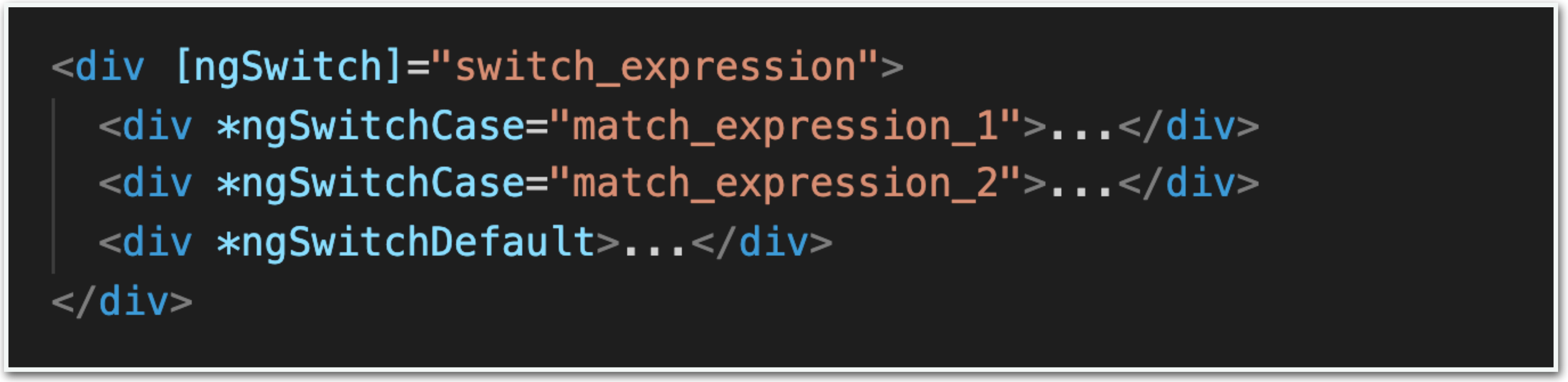
ngIf Directive

ngSwitch

- ngSwitch displays one element from a set of element trees based on conditions
- Return a switch value
- At any particular moment, at most one of these span is in the DOM

ngSwitch (contd.)

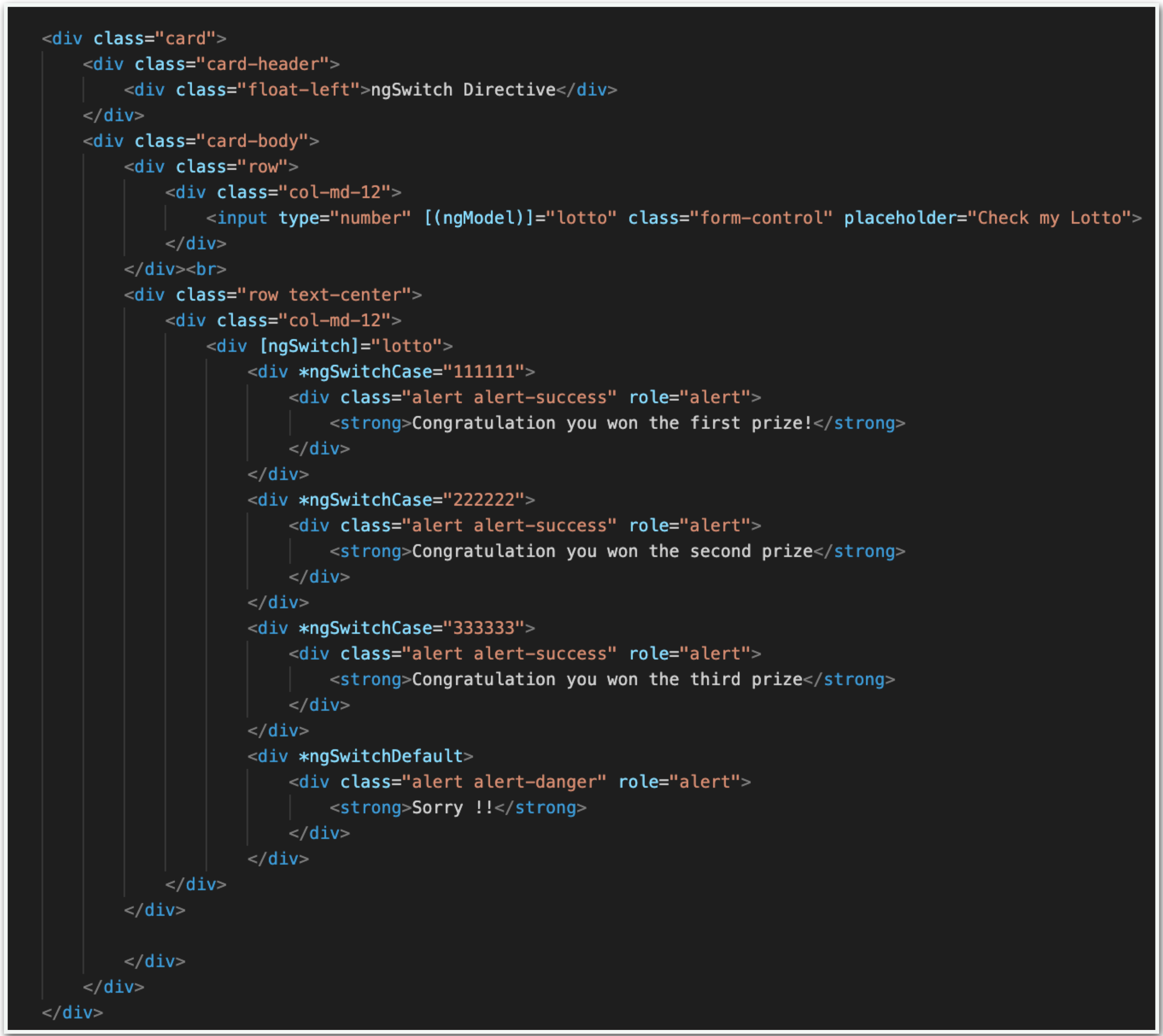
- Pattern:



```
<div [ngSwitch]="switch_expression">
  <div *ngSwitchCase="match_expression_1">...</div>
  <div *ngSwitchCase="match_expression_2">...</div>
  <div *ngSwitchDefault>...</div>
</div>
```

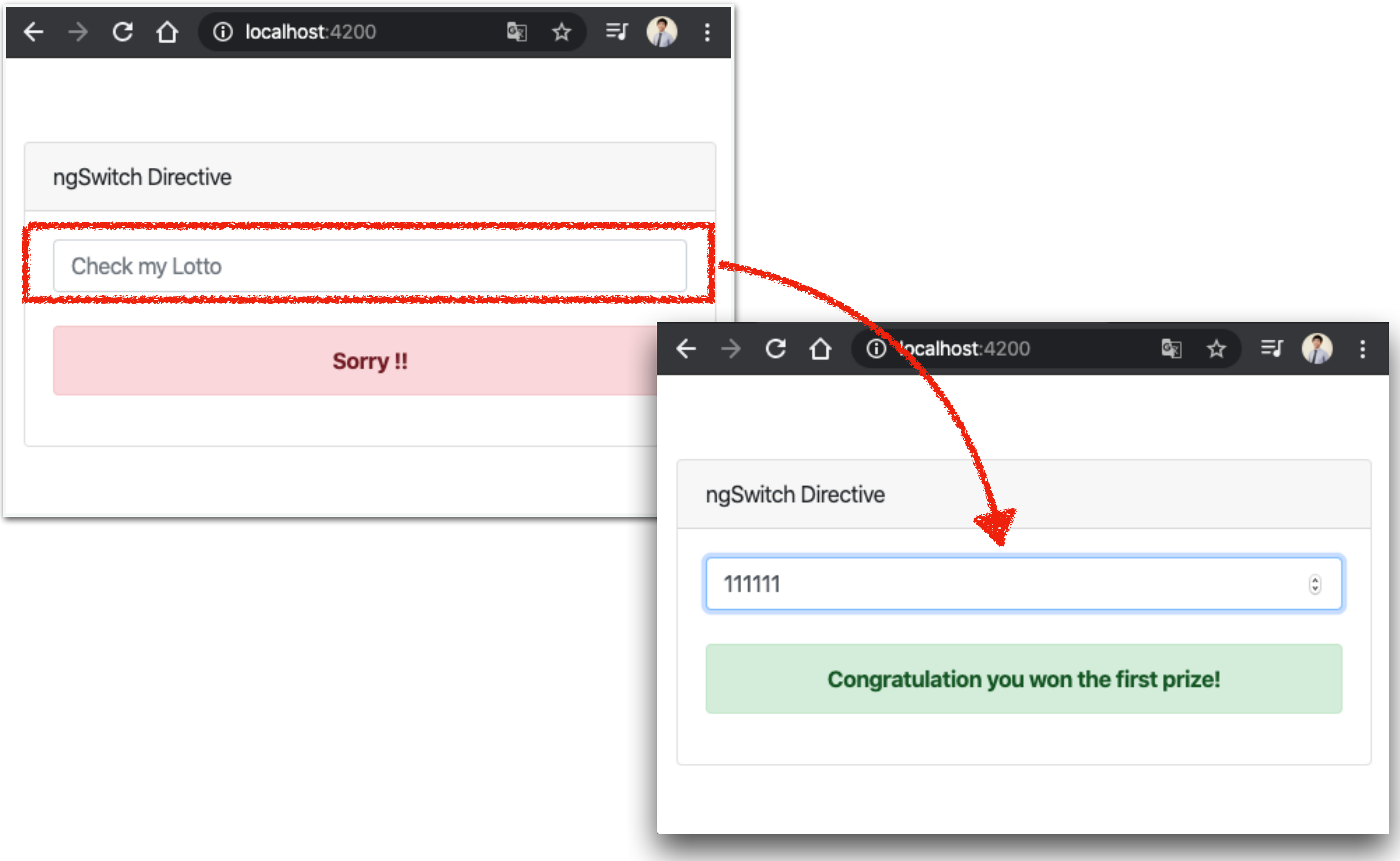
- 3 collaborating directives are at work here
  - *ngSwitch* : bound to an expression that returns the switch value
  - *ngSwitchCase* : bound to an expression returning a match value
  - *ngSwitchDefault* : a maker attribute on the default element

Example: ngSwitch



```html
<div class="card">
    <div class="card-header">
        <div class="float-left">ngSwitch Directive</div>
    </div>
    <div class="card-body">
        <div class="row">
            <div class="col-md-12">
                <input type="number" [(ngModel)]="lotto" class="form-control" placeholder="Check my Lotto">
            </div>
        </div><br>
        <div class="row text-center">
            <div class="col-md-12">
                <div [ngSwitch]="lotto">
                    <div *ngSwitchCase="111111">
                        <div class="alert alert-success" role="alert">
                            <strong>Congratulation you won the first prize!</strong>
                        </div>
                    </div>
                    <div *ngSwitchCase="222222">
                        <div class="alert alert-success" role="alert">
                            <strong>Congratulation you won the second prize</strong>
                        </div>
                    </div>
                    <div *ngSwitchCase="333333">
                        <div class="alert alert-success" role="alert">
                            <strong>Congratulation you won the third prize</strong>
                        </div>
                    </div>
                    <div *ngSwitchDefault>
                        <div class="alert alert-danger" role="alert">
                            <strong>Sorry !!</strong>
                        </div>
                    </div>
            </div>
        </div>

        </div>
    </div>
</div>
```

Example: ngSwitch


localhost:4200

ngSwitch Directive

Check my Lotto

Sorry !!

localhost:4200

ngSwitch Directive

111111

Congratulation you won the first prize!

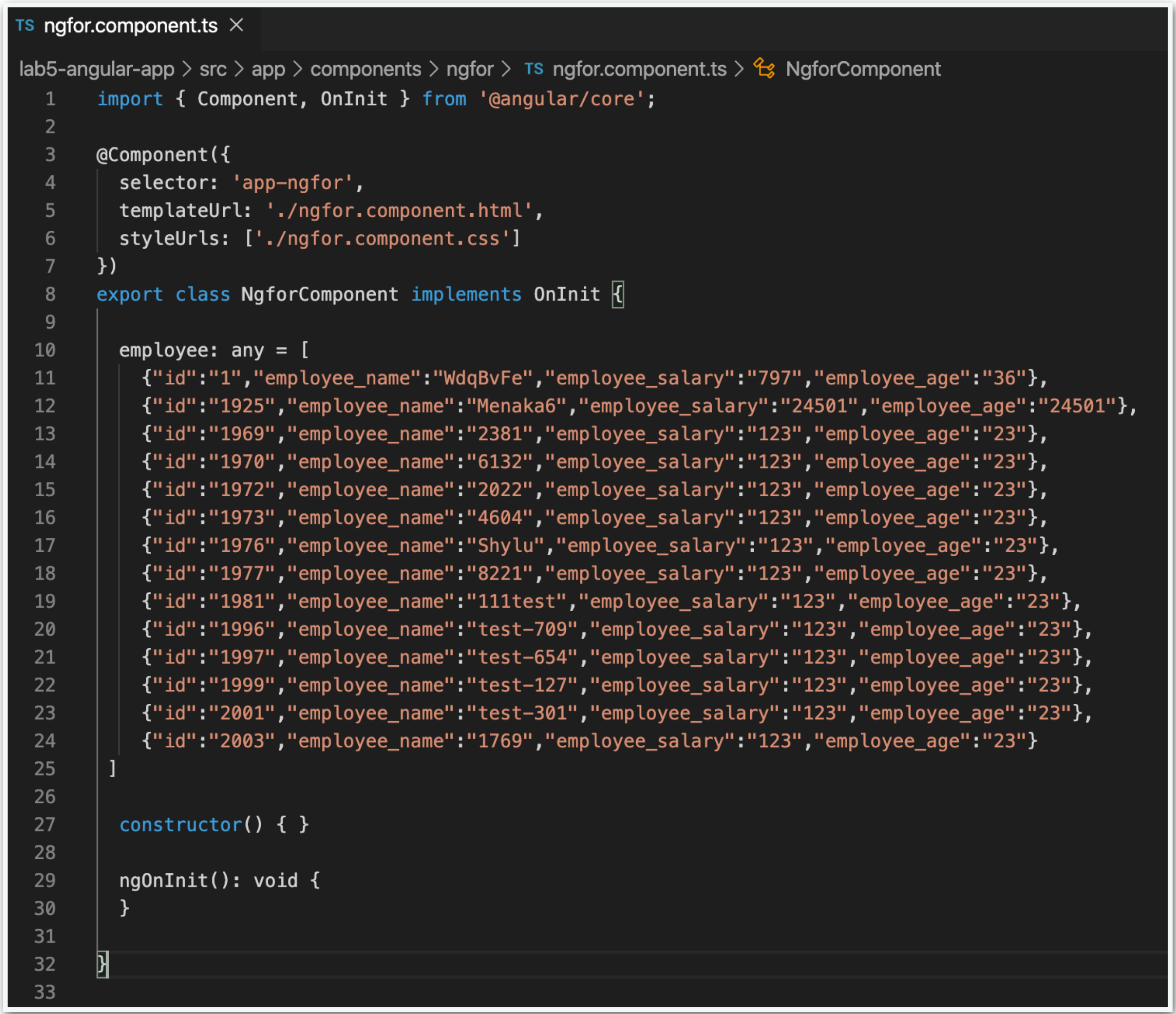
ngFor

- ngFor is a repeater directive - present a list of items
- Pattern :

```
*ngFor="let item of list"
```

Example: ngFor

Controller

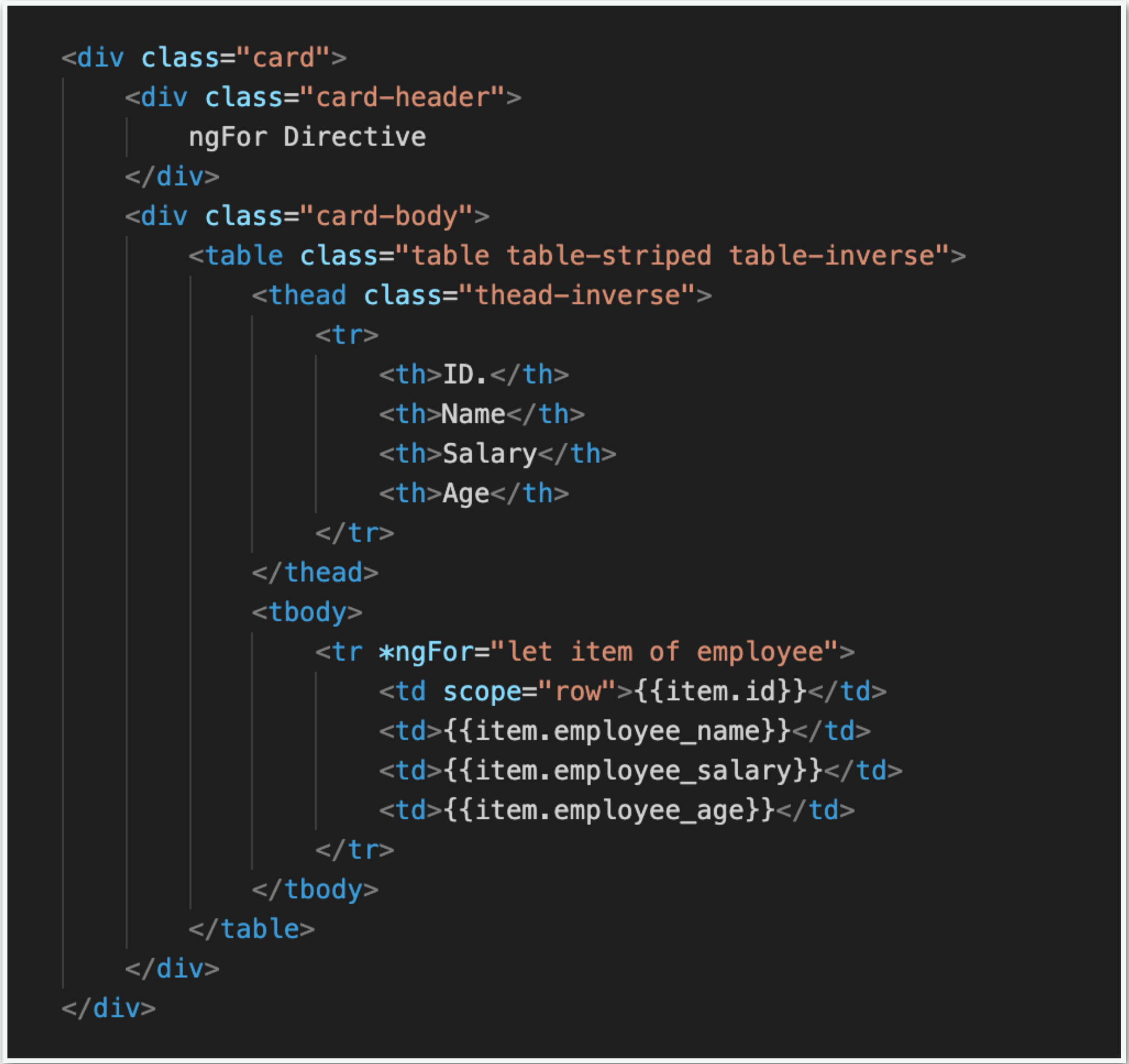


TS ngfor.component.ts ×

lab5-angular-app > src > app > components > ngfor > TS ngfor.component.ts > NgforComponent

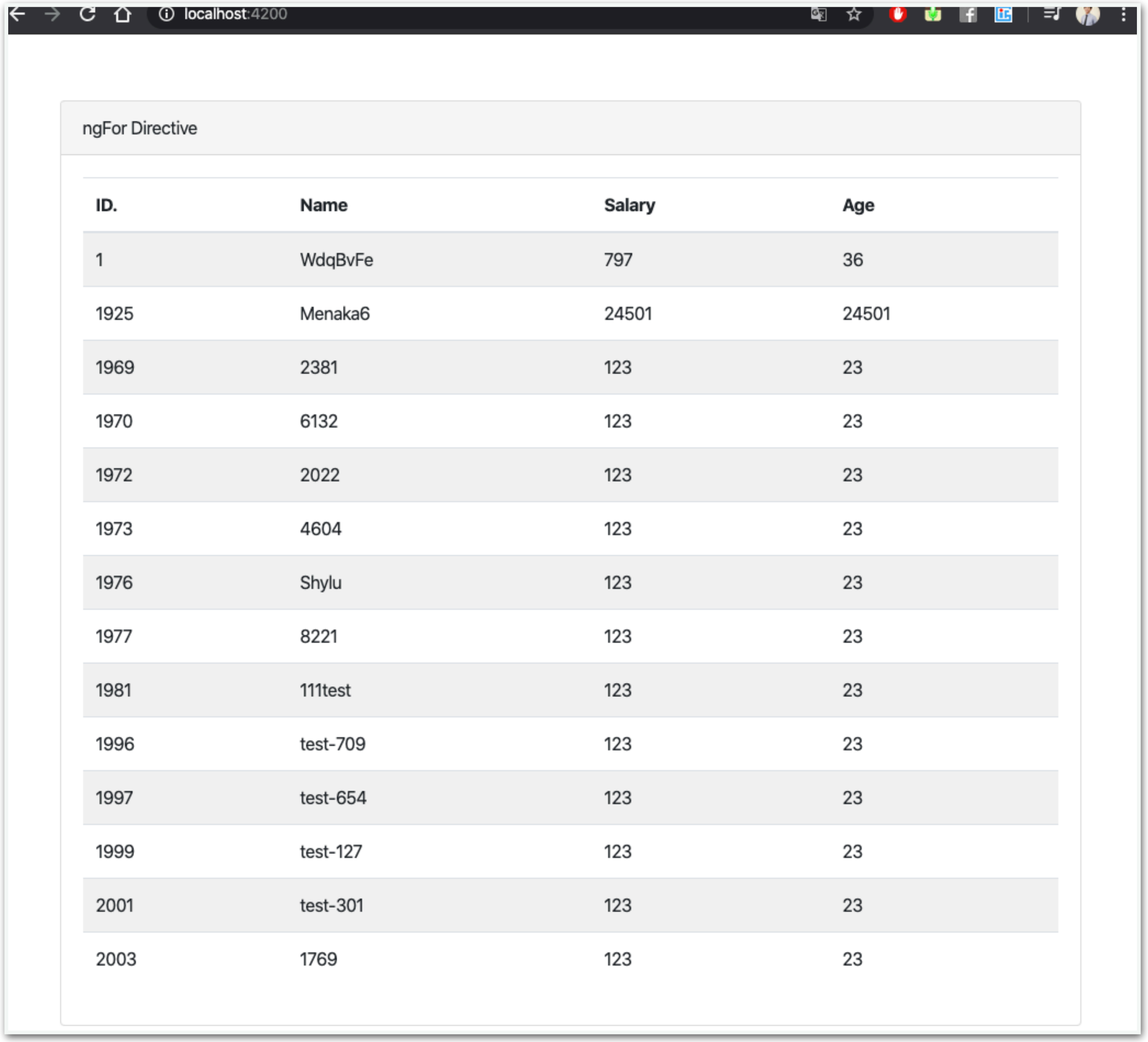
```typescript
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-ngfor',
  templateUrl: './ngfor.component.html',
  styleUrls: ['./ngfor.component.css']
})
export class NgforComponent implements OnInit {

  employee: any = [
    {"id":"1","employee_name":"WdqBvFe","employee_salary":"797","employee_age":"36"},
    {"id":"1925","employee_name":"Menaka6","employee_salary":"24501","employee_age":"24501"},
    {"id":"1969","employee_name":"2381","employee_salary":"123","employee_age":"23"},
    {"id":"1970","employee_name":"6132","employee_salary":"123","employee_age":"23"},
    {"id":"1972","employee_name":"2022","employee_salary":"123","employee_age":"23"},
    {"id":"1973","employee_name":"4604","employee_salary":"123","employee_age":"23"},
    {"id":"1976","employee_name":"Shylu","employee_salary":"123","employee_age":"23"},
    {"id":"1977","employee_name":"8221","employee_salary":"123","employee_age":"23"},
    {"id":"1981","employee_name":"111test","employee_salary":"123","employee_age":"23"},
    {"id":"1996","employee_name":"test-709","employee_salary":"123","employee_age":"23"},
    {"id":"1997","employee_name":"test-654","employee_salary":"123","employee_age":"23"},
    {"id":"1999","employee_name":"test-127","employee_salary":"123","employee_age":"23"},
    {"id":"2001","employee_name":"test-301","employee_salary":"123","employee_age":"23"},
    {"id":"2003","employee_name":"1769","employee_salary":"123","employee_age":"23"}
 ]

  constructor() { }

  ngOnInit(): void {
  }

}
```

Template



```html
<div class="card">
    <div class="card-header">
        ngFor Directive
    </div>
    <div class="card-body">
        <table class="table table-striped table-inverse">
            <thead class="thead-inverse">
                <tr>
                    <th>ID.</th>
                    <th>Name</th>
                    <th>Salary</th>
                    <th>Age</th>
                </tr>
            </thead>
            <tbody>
                <tr *ngFor="let item of employee">
                    <td scope="row">{{item.id}}</td>
                    <td>{{item.employee_name}}</td>
                    <td>{{item.employee_salary}}</td>
                    <td>{{item.employee_age}}</td>
                </tr>
            </tbody>
        </table>
    </div>
</div>
```

Example: ngFor



ngFor Directive

ID.	Name	Salary	Age
1	WdqBvFe	797	36
1925	Menaka6	24501	24501
1969	2381	123	23
1970	6132	123	23
1972	2022	123	23
1973	4604	123	23
1976	Shylu	123	23
1977	8221	123	23
1981	111test	123	23
1996	test-709	123	23
1997	test-654	123	23
1999	test-127	123	23
2001	test-301	123	23
2003	1769	123	23