



Codergirl – Frontend

Unit 2 - Class 5

December 14, 2020

Agenda

Section 7 and 8

Attribute and Structural Directives

Building our own

Studio on Directives

Structural Directives

- private versus public
- Only one structural directive can be applied to the same element
- ngFor and ngIf
- Exercise

Structural Directives

```
<div class="container">
  <div class="row">
    <div class="col-xs-12">
      <button
        class="btn btn-primary"
        (click)="onlyOdd = !onlyOdd">Only show odd numbers</button>
      <br><br>
      <ul class="list-group">
        <div *ngIf="onlyOdd">
          <li
            class="list-group-item" *ngFor="let odd of oddNumbers">
            {{ odd }}
          </li>
        </div>
        <div *ngIf="!onlyOdd">
          <li
            class="list-group-item" *ngFor="let even of evenNumbers">
            {{ even }}
          </li>
        </div>
      </ul>
    </div>
  </div>
</div>
```

app.component.html

Structural Directives

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  numbers = [1, 2, 3, 4, 5];
  oddNumbers = [1,3,5];
  evenNumbers = [2,4,6];
  onlyOdd = false;
}
```

app.component.ts

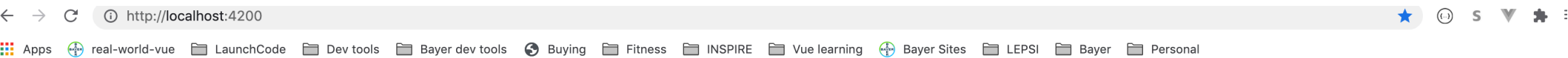
Structural Directives

Only show odd numbers

2

4

6



Only show odd numbers

1

3

5

ngClass – [] means binding to ngClass directive

```
.container {  
  margin-top: 30px;  
}  
  
.odd {  
  color: red;  
}
```

app.component.css ngClass of odd gets assigned if the condition evaluates to true

```
<div class="container">  
  <div class="row">  
    <div class="col-xs-12">  
      <button  
        class="btn btn-primary"  
        (click)="onlyOdd = !onlyOdd">Only show odd numbers</button>  
      <br><br>  
      <ul class="list-group">  
        <div *ngIf="onlyOdd">  
          <li  
            class="list-group-item"  
            [ngClass]="{odd: odd % 2 !== 0}"  
            *ngFor="let odd of oddNumbers">  
              {{ odd }}  
          </li>  
        </div>  
        <div *ngIf="!onlyOdd">  
          <li  
            class="list-group-item"  
            [ngClass]="{odd: even % 2 !== 0}"  
            *ngFor="let even of evenNumbers">  
              {{ even }}  
          </li>  
        </div>  
      </ul>  
    </div>  
  </div>
```

ngClass



ngStyle

```
<div class="container">
  <div class="row">
    <div class="col-xs-12">
      <button
        class="btn btn-primary"
        (click)="onlyOdd = !onlyOdd">Only show odd numbers</button>
      <br><br>
      <ul class="list-group">
        <div *ngIf="onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: odd % 2 !== 0}"
            [ngStyle]="{backgroundColor: odd % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let odd of oddNumbers">
            {{ odd }}
          </li>
        </div>
        <div *ngIf="!onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: even % 2 !== 0}"
            [ngStyle]="{backgroundColor: even % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let even of evenNumbers">
            {{ even }}
          </li>
        </div>
      </ul>
    </div>
  </div>
</div>
```

app-component.html

ngStyle



Only show odd numbers

2

4

6



Only show odd numbers

1

3

5

Building our own directive

Goal: Build a directive which highlights an element when I hover over it.

- Create a new file called basic-highlight.directive.ts
- Add the `@Directive` and import it
- Angular injects the element into the class that the directive sits on
- Easy way to get access to classes – injection
- Angular will try to give the reference to the element using `elementRef`
- Access of `private` makes it a property of class and accessible to class.
- Use the `elementRef` in the `ngOnInit` method.
- Have to inform Angular about the directive (most things setup wise are in the `app.module.ts`)

Making your own directive

```
import { Directive, ElementRef, OnInit } from "@angular/core";

@Directive({
  selector: '[appBasicHighlight]'
})
export class BasicHighlightDirective implements OnInit {

  constructor(private elementRef: ElementRef) {

  }

  ngOnInit() {
    this.elementRef.nativeElement.style.backgroundColor = 'green';
  }
}
```

basic-highlight.directive.ts

The square bracket signify it's an attribute

Setup for your own directive

```
<div class="container">
  <div class="row">
    <div class="col-xs-12">
      <button
        class="btn btn-primary"
        (click)="onlyOdd = !onlyOdd">Only show odd numbers</button>
      <br><br>
      <ul class="list-group">
        <div *ngIf="onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: odd % 2 !== 0}"
            [ngStyle]="{backgroundColor: odd % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let odd of oddNumbers">
            {{ odd }}
          </li>
        </div>
        <div *ngIf="!onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: even % 2 !== 0}"
            [ngStyle]="{backgroundColor: even % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let even of evenNumbers">
            {{ even }}
          </li>
        </div>
      </ul>
      <p appBasicHighlight>Style me with basic directive</p>
    </div>
  </div>
</div>
```

Setup my own directive

← → ↻ http://localhost:4200 ★

Apps real-world-vue LaunchCode Dev tools Bayer dev tools Buying Fitness INSPIRE Vue learning Bayer Sites LEPSI Bayer Personal

Only show odd numbers

2
4
6

Sign me with basic directive

Using Renderer to build directive

Accessing elements directly is not a great idea

Create director using cli

ng g d better-highlight or

ng generate directive better-highlight

You can create a common folder for directives or use better-highlight folder

Access to DOM might not always be possible if app not accessible in browser.
Mostly both work!

Using Renderer to build directive

```
import { OnInit, Directive, Renderer2, ElementRef } from '@angular/core';

@Directive({
  selector: '[appBetterHighlight]'
})
export class BetterHighlightDirective implements OnInit {

  constructor(private elRef: ElementRef, private renderer: Renderer2) { }

  ngOnInit() {
    this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'blue');
  }
}
```

better-highlight.directive.ts

Using Renderer to build directive

```
<div class="container">
  <div class="row">
    <div class="col-xs-12">
      <button
        class="btn btn-primary"
        (click)="onlyOdd = !onlyOdd">Only show odd numbers</button>
      <br><br>
      <ul class="list-group">
        <div *ngIf="onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: odd % 2 !== 0}"
            [ngStyle]="{backgroundColor: odd % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let odd of oddNumbers">
            {{ odd }}
          </li>
        </div>
        <div *ngIf="!onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: even % 2 !== 0}"
            [ngStyle]="{backgroundColor: even % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let even of evenNumbers">
            {{ even }}
          </li>
        </div>
      </ul>
      <p appBasicHighlight>Style me with basic directive</p>
      <p appBetterHighlight>Style me with better directive</p>
    </div>
  </div>
</div>
```

app.component.html

Using Renderer to build directive

← → ↻ ⓘ http://localhost:4200 ★

Apps real-world-vue LaunchCode Dev tools Bayer dev tools Buying Fitness INSPIRE Vue learning Bayer Sites LEPSI Bayer Personal

Only show odd numbers

2
4
6

Style me with basic directive

Style me with better directive

HostListener

```
import { OnInit, Directive, Renderer2, ElementRef, HostListener } from '@angular/core';

@Directive({
  selector: '[appBetterHighlight]'
})
export class BetterHighlightDirective implements OnInit {

  constructor(private elRef: ElementRef, private renderer: Renderer2) { }

  ngOnInit() {
    // this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'blue');
  }

  @HostListener('mouseenter') mouseover(eventData: Event) {
    this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'blue');
  }

  @HostListener('mouseleave') mouseleave(eventData: Event) {
    this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'transparent');
  }
}
```

better-highlight.directive.ts

HostListener

← → ↻ http://localhost:4200

Apps real-world-vue LaunchCode Dev tools Bayer dev tools Buying Fitness INSPIRE Vue learning Bayer Sites LEPSI Bayer Personal

Only show odd numbers

2
4
6

Style me with basic directive

Style me with better directive

← → ↻ http://localhost:4200

Apps real-world-vue LaunchCode Dev tools Bayer dev tools Buying Fitness INSPIRE Vue learning Bayer Sites LEPSI Bayer Personal

Only show odd numbers

2
4
6

Style me with basic directive

Style me with better directive

HostBinding

```
import { OnInit, Directive, Renderer2, ElementRef, HostListener, HostBinding } from '@angular/core';

@Directive({
  selector: '[appBetterHighlight]'
})
export class BetterHighlightDirective implements OnInit{

  @HostBinding('style.backgroundColor') backgroundColor: string = 'transparent';

  constructor(private elRef: ElementRef, private renderer: Renderer2) { }

  ngOnInit() {
    // this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'blue');
  }

  @HostListener('mouseenter') mouseover(eventData: Event) {
    // this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'blue');
    this.backgroundColor = 'blue';
  }

  @HostListener('mouseleave') mouseleave(eventData: Event) {
    // this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'transparent');
    this.backgroundColor = 'transparent';
  }
}
```

HostBinding



Only show odd numbers

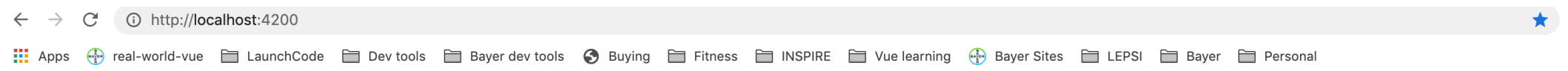
2

4

6

Style me with basic directive

Style me with better directive



Only show odd numbers

2

4

6

Style me with basic directive

Style me with better directive

Binding to Directive Properties

```
import { OnInit, Directive, Renderer2, ElementRef, HostListener, HostBinding, Input } from '@angular/core';

@Directive({
  selector: '[appBetterHighlight]'
})
export class BetterHighlightDirective implements OnInit {

  @Input() defaultColor: string = 'transparent';
  @Input() highlightColor: string = 'blue';

  @HostBinding('style.backgroundColor') backgroundColor: string = this.defaultColor;

  constructor(private elRef: ElementRef, private renderer: Renderer2) { }

  ngOnInit() {
    this.backgroundColor = this.defaultColor;
    // this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'blue');
  }

  @HostListener('mouseenter') mouseover(eventData: Event) {
    // this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'blue');
    this.backgroundColor = this.highlightColor;
  }

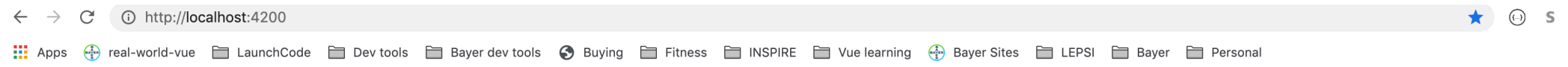
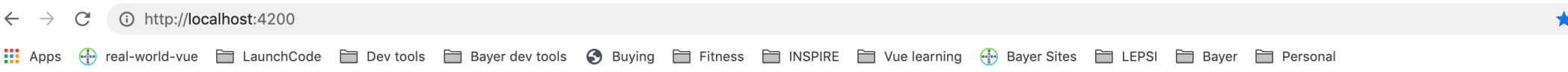
  @HostListener('mouseleave') mouseleave(eventData: Event) {
    // this.renderer.setStyle(this.elRef.nativeElement, 'background-color', 'transparent');
    this.backgroundColor = this.defaultColor;
  }
}
```

```

<div class="container">
  <div class="row">
    <div class="col-xs-12">
      <button
        class="btn btn-primary"
        (click)="onlyOdd = !onlyOdd">Only show odd numbers</button>
      <br><br>
      <ul class="list-group">
        <div *ngIf="onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: odd % 2 !== 0}"
            [ngStyle]="{backgroundColor: odd % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let odd of oddNumbers">
            {{ odd }}
          </li>
        </div>
        <div *ngIf="!onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: even % 2 !== 0}"
            [ngStyle]="{backgroundColor: even % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let even of evenNumbers">
            {{ even }}
          </li>
        </div>
      </ul>
      <p appBasicHighlight>Style me with basic directive</p>
      <p appBetterHighlight [defaultColor]='yellow' [highlightColor]='red'>Style me with better
directive</p>
    </div>
  </div>
</div>

```


Binding to Directive Properties



Binding to Directive Properties Notes

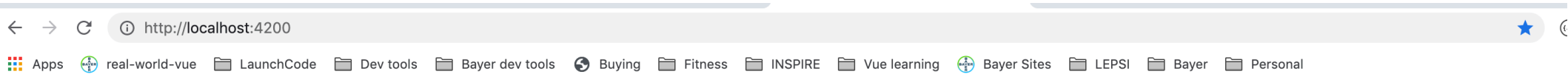
- Angular figures out whether u passing Property of directive or the tag
- Can omit square brackets for property binding and “ if passing string

Story behind *

Angular transforms *

```
<div class="container">
  <div class="row">
    <div class="col-xs-12">
      <button
        class="btn btn-primary"
        (click)="onlyOdd = !onlyOdd">Only show odd numbers</button>
      <br><br>
      <ul class="list-group">
        <div *ngIf="onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: odd % 2 !== 0}"
            [ngStyle]="{backgroundColor: odd % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let odd of oddNumbers">
            {{ odd }}
          </li>
        </div>
        <div *ngIf="!onlyOdd">
          <li
            class="list-group-item"
            [ngClass]="{odd: even % 2 !== 0}"
            [ngStyle]="{backgroundColor: even % 2 !== 0 ? 'yellow' : 'transparent'}"
            *ngFor="let even of evenNumbers">
            {{ even }}
          </li>
        </div>
        <ng-template [ngIf]="!onlyOdd">
          <div>
            <li
              class="list-group-item"
              [ngClass]="{odd: even % 2 !== 0}"
              [ngStyle]="{backgroundColor: even % 2 !== 0 ? 'yellow' : 'transparent'}"
              *ngFor="let even of evenNumbers">
              {{ even }}
            </li>
          </div>
        </ng-template>
      </ul>
      <p appBasicHighlight>Style me with basic directive</p>
      <p appBetterHighlight [defaultColor]='yellow' [highlightColor]='red'>Style me with better
directive</p>
    </div>
  </div>
</div>
```

Story behind *



Only show odd numbers

2

4

6

2

4

6

Style me with basic directive

Style me with better directive

Building an unless directive – ng g d unless

```
import { Directive, Input, TemplateRef, ViewContainerRef } from '@angular/core';

@Directive({
  selector: '[appUnless]'
})
export class UnlessDirective {

  @Input() set appUnless(condition: boolean) {
    if(!condition) {
      this.vcRef.createEmbeddedView(this.templateRef)
    } else {
      this.vcRef.clear();
    }
  }

  constructor(private templateRef: TemplateRef<any>, private vcRef: ViewContainerRef) { }
}
```


Building an unless directive

← → ↻ http://localhost:4200 ★

Apps real-world-vue LaunchCode Dev tools Bayer dev tools Buying Fitness INSPIRE Vue learning Bayer Sites LEPSI Bayer Personal

Only show odd numbers

2

4

6

Style me with basic directive

Style me with better directive

Only show odd numbers

1

3

5

Style me with basic directive

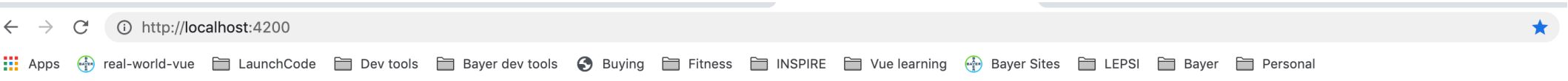
Style me with better directive

ngSwitch

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  numbers = [1, 2, 3, 4, 5];
  oddNumbers = [1,3,5];
  evenNumbers = [2,4,6];
  onlyOdd = false;
  value = 10;
}
```

ngSwitch



Only show odd numbers

2

4

6

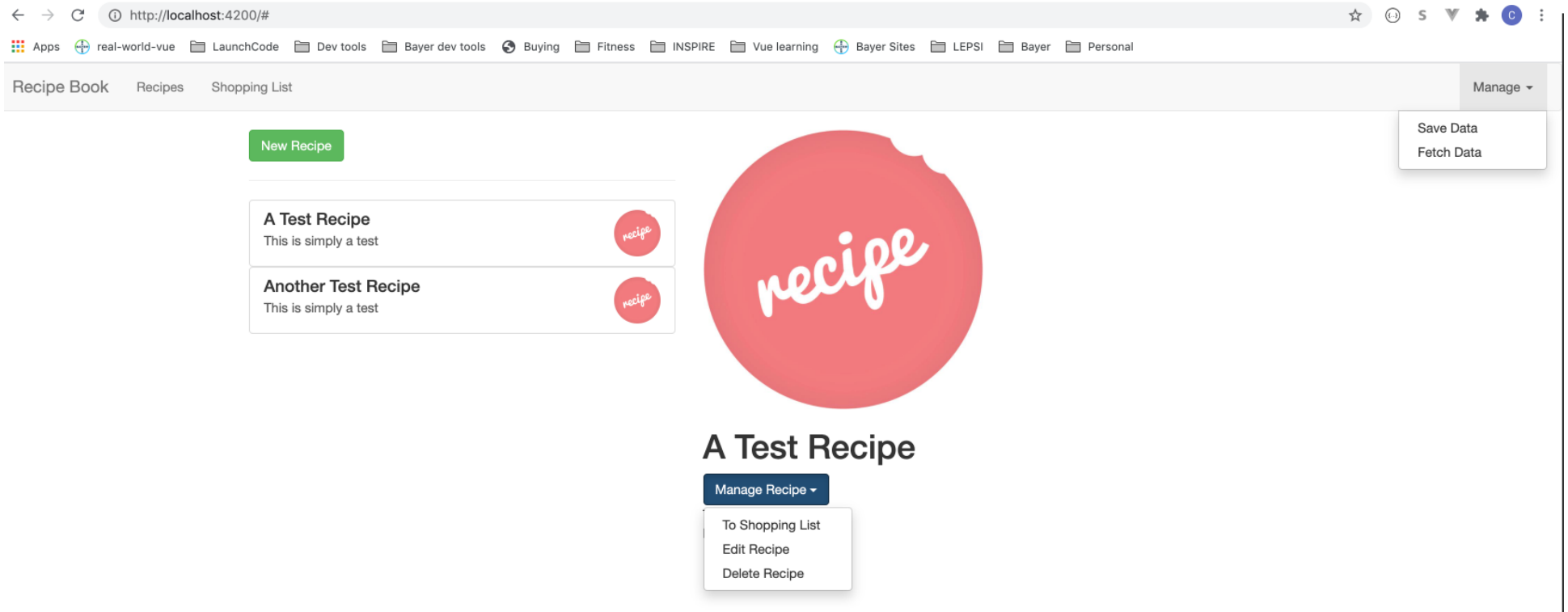
Style me with basic directive

Style me with better directive

Value is 10

Questions?

Studio



Create dropdown directive