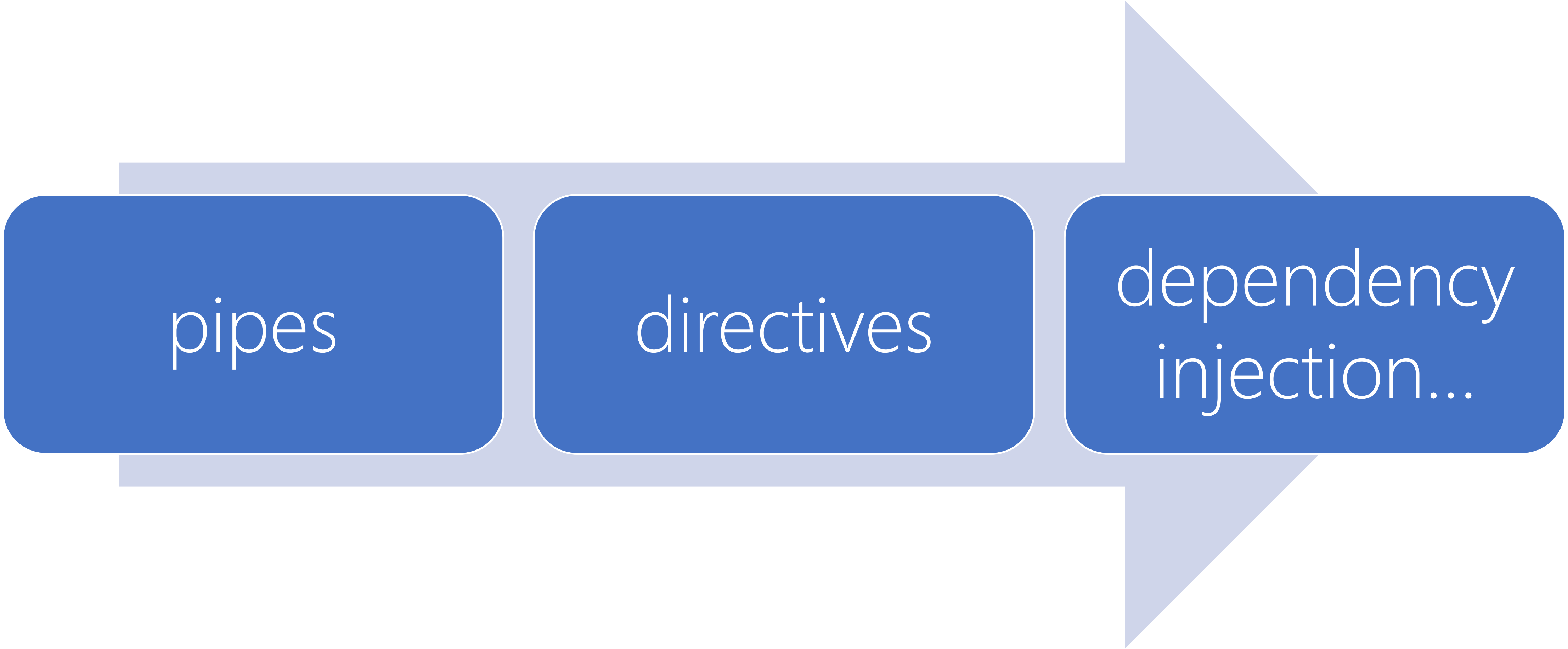
part 7

kamil mijacz

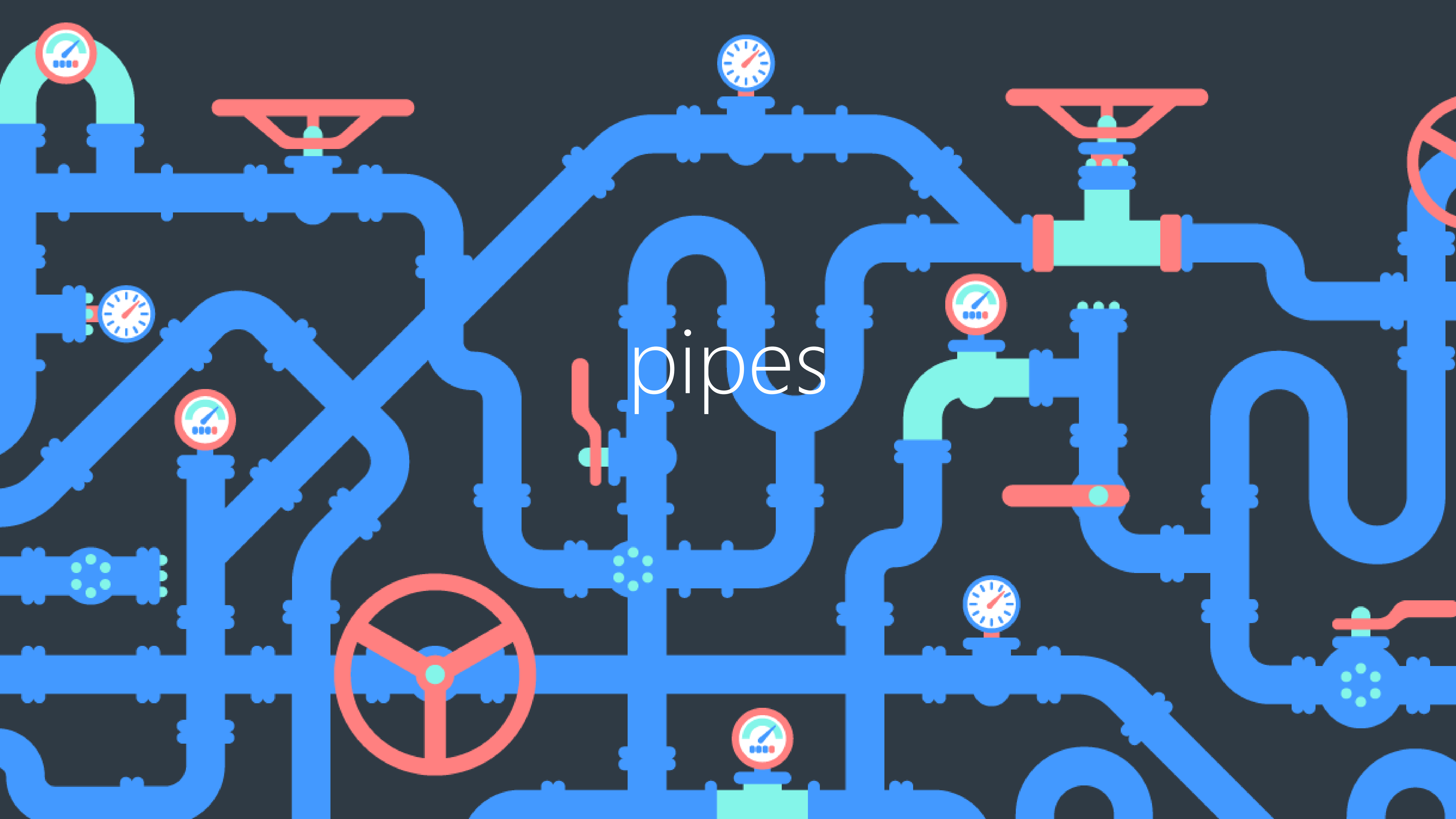
plan



pipes

directives

dependency injection…

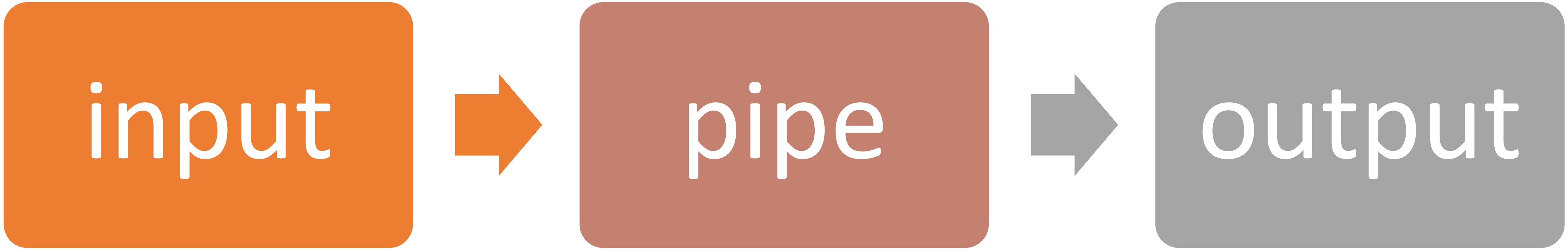


pipes

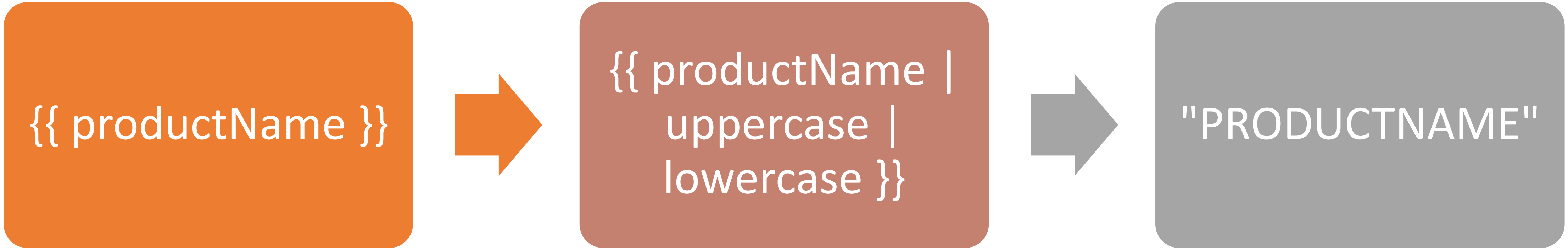
what is pipe?

- classes
- prepare data for display
- transforming text into another
- @Pipe decorator
- for code reuse

understanding pipe



understanding pipe

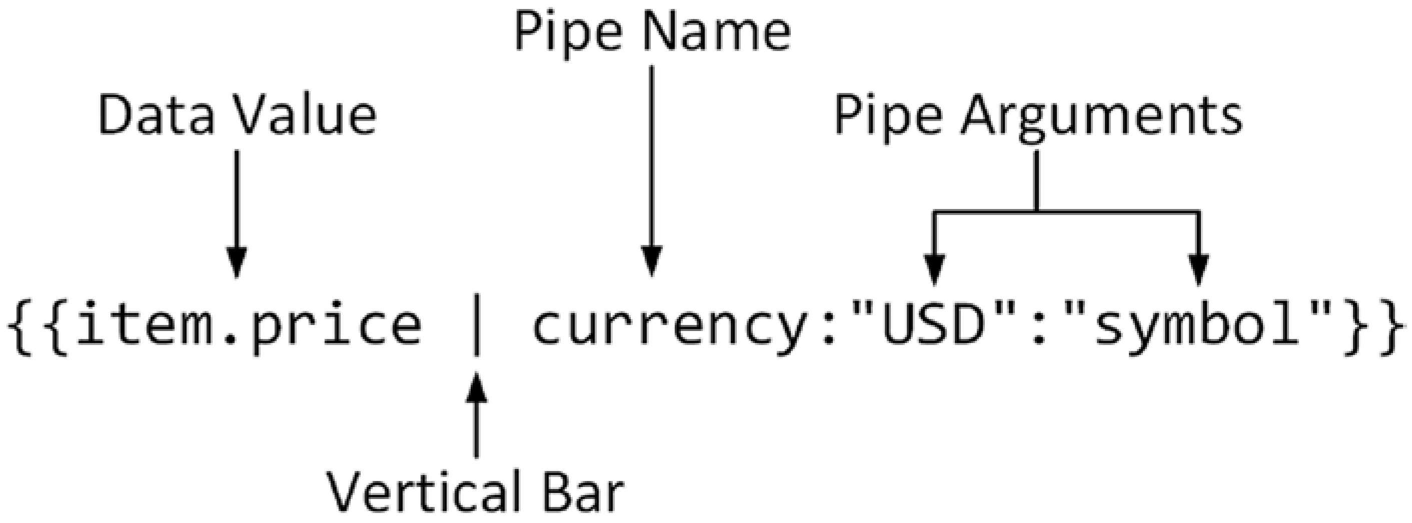


{{ productName }}

{{ productName | uppercase | lowercase }}

"PRODUCTNAME"

understanding pipe



Pipe Name

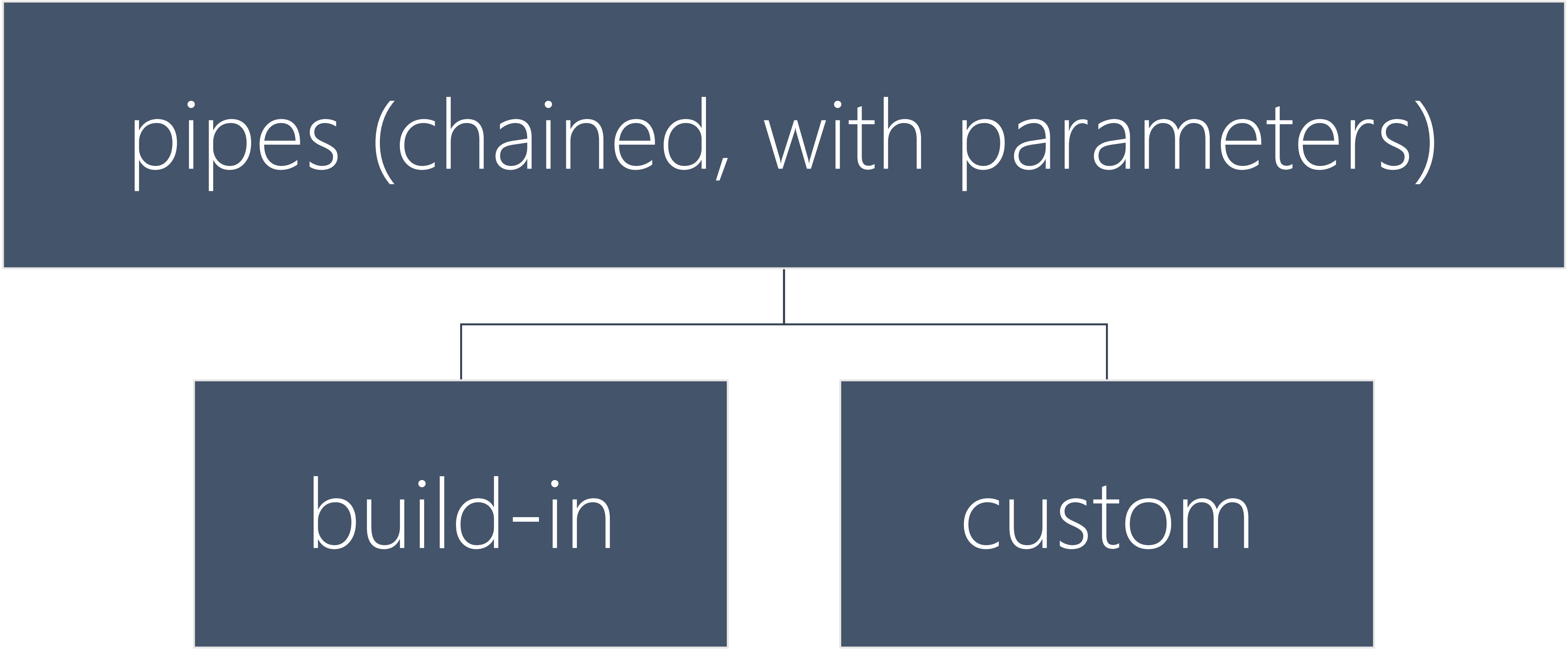
Data Value

Pipe Arguments

```
{{item.price | currency:"USD":"symbol"}}
```

Vertical Bar

types of pipes

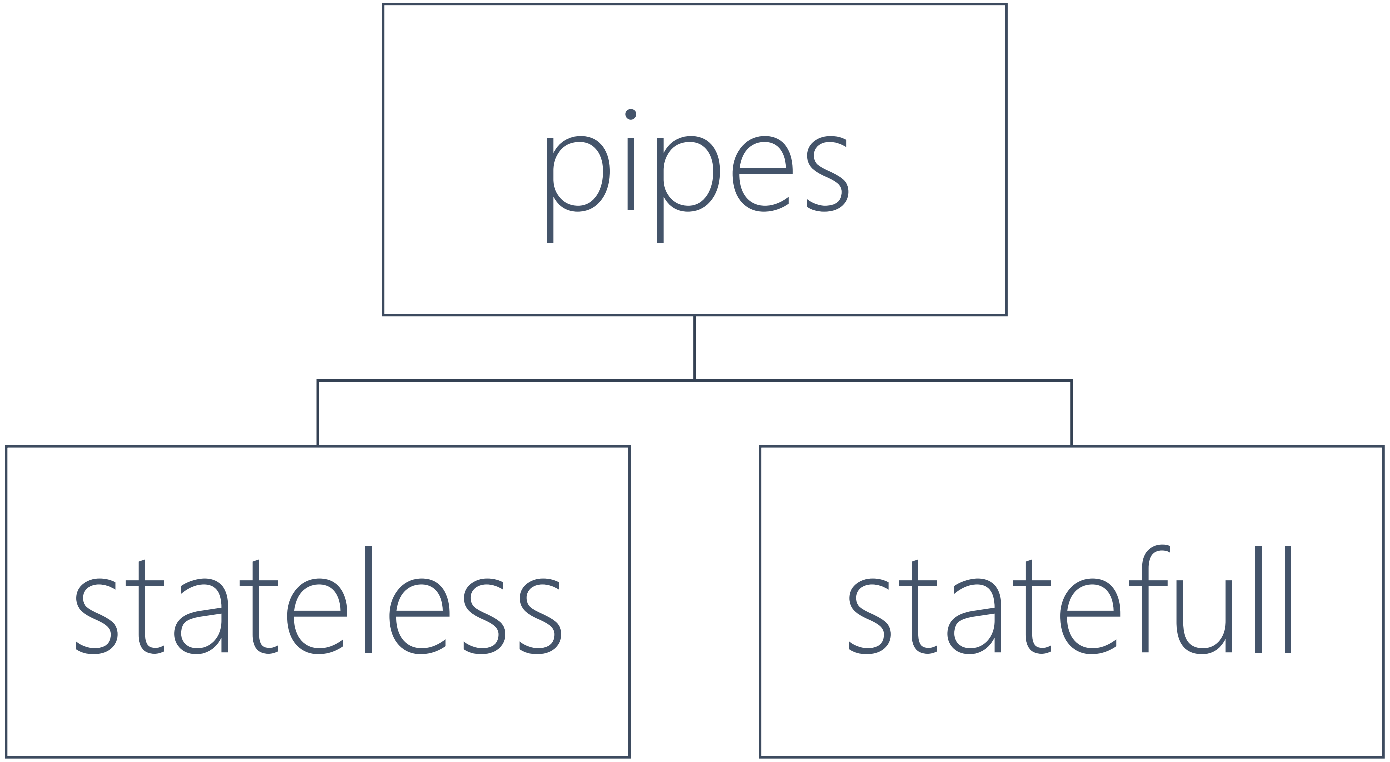


pipes (chained, with parameters)

build-in

custom

types of pipes



pipes

stateless

statefull

examples

```
{{ 'Hello World' | uppercase }}
{{ 'hELLO wORLD' | lowercase }}
{{ 'jOHN' | titlecase }}
{{ currentDate | date }}
{{ currentDate | date:'shortDate' }}
{{ '12.123456' | number }}
{{ 150 | currency:'GBP' }}
{{ complexObject | json }}
```

…

custom pipe

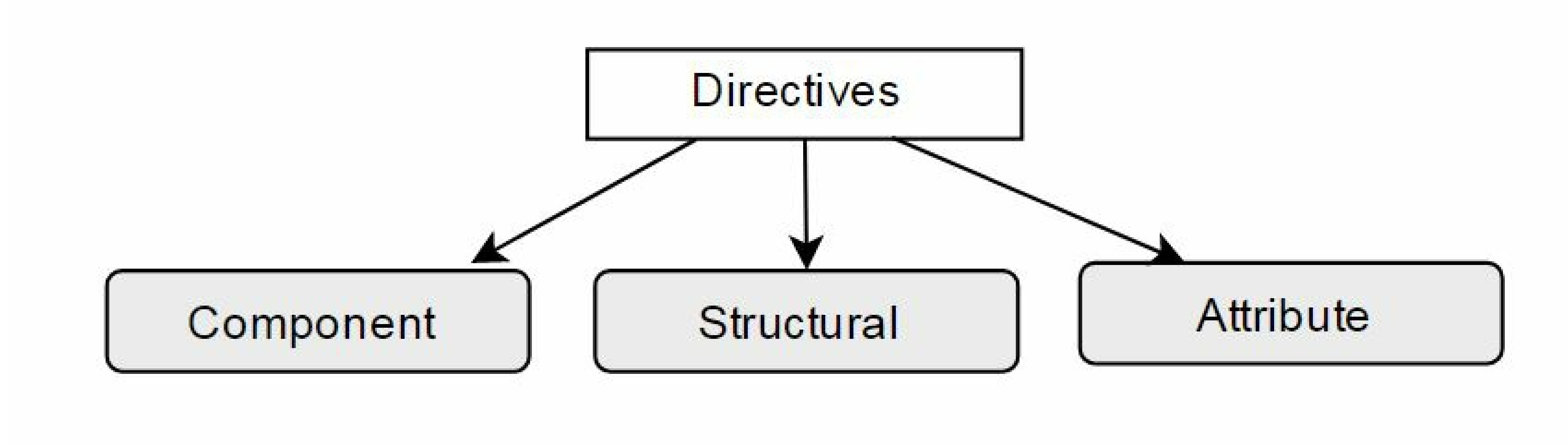
```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({ name: 'fileSize’ })
export class FileSizePipe implements PipeTransform {

  transform(bytes: number = 0, decimals: number = 2): string {
    if (bytes === 0) return '0 Bytes';
    const k = 1024, dm = decimals || 2,
    sizes = ['Bytes', 'KB', 'MB', 'GB’, …];
    i = Math.floor(Math.log(bytes) / Math.log(k));
    return parseFloat((bytes / Math.pow(k, i)).toFixed(dm)) + ' ' + sizes[i];
 }
}
```

DIRECTIVE

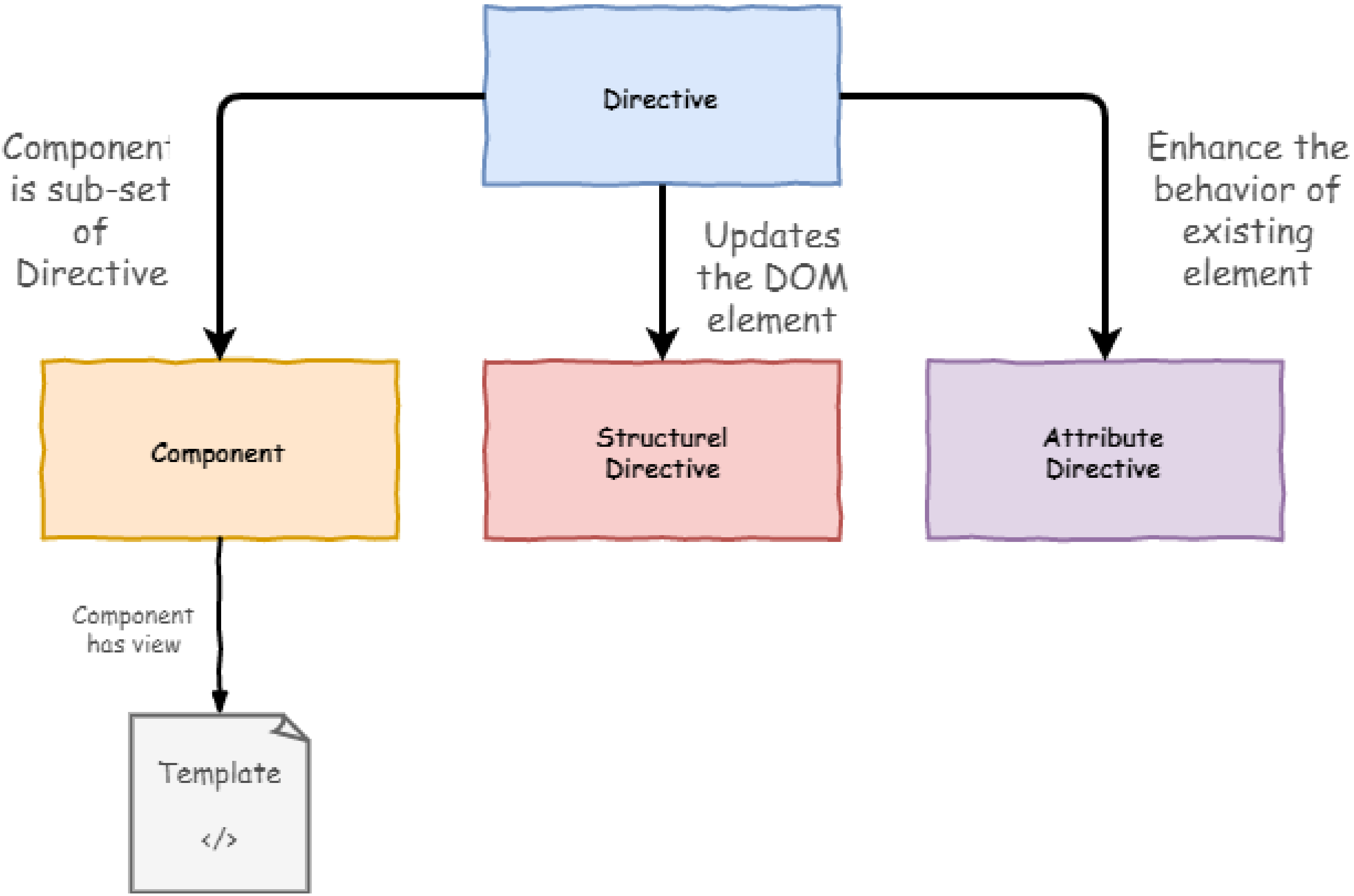
directives

- extend the behavior of the elements
- can manipulate DOM
- @Directive
- 3 types

directive types



directive types



directive types

components

- directives with a template

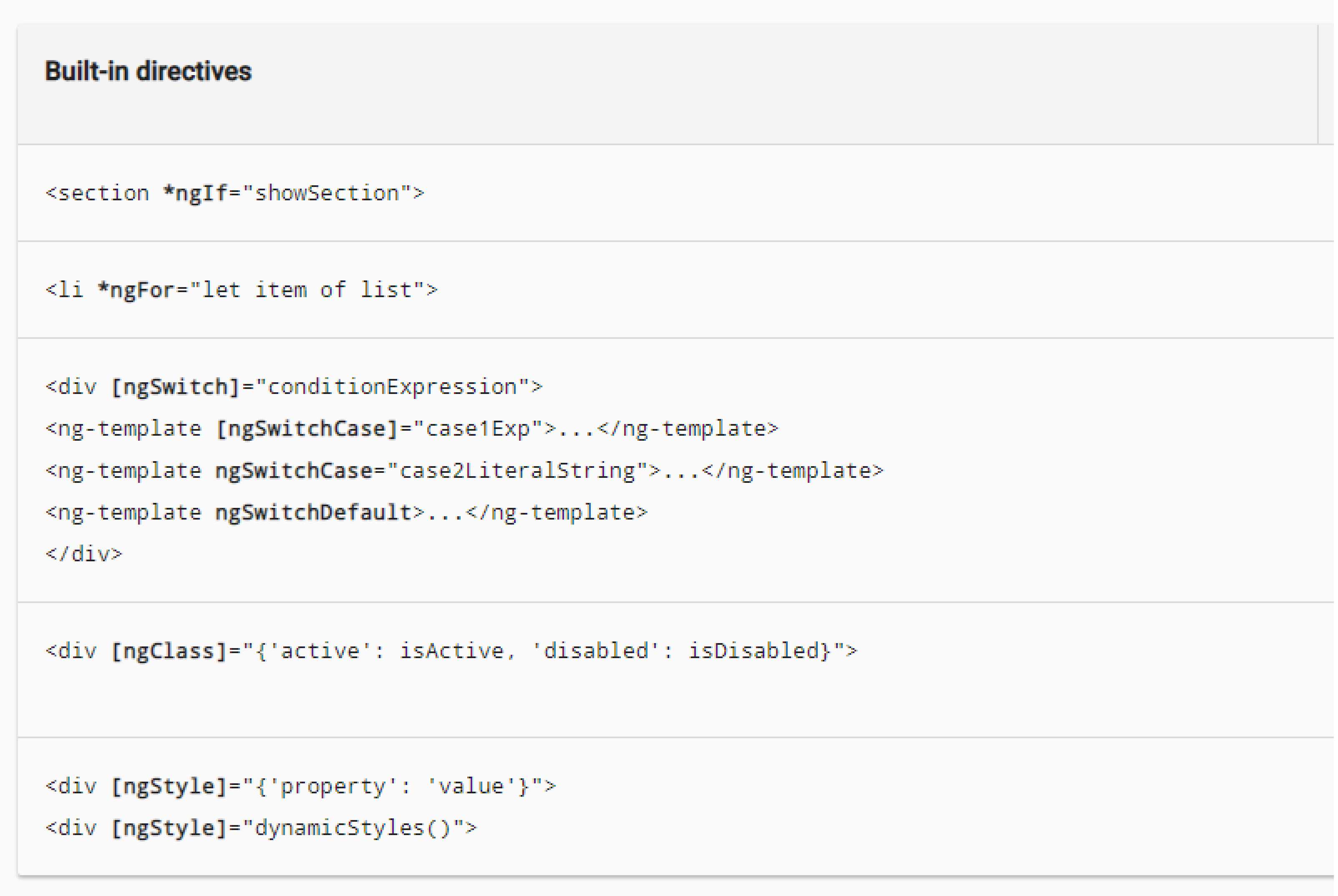
structural directives

- change the DOM layout by adding and removing DOM elements

attribute directives

- change the appearance or behavior of an element/component/directive

https://angular.io/guide/cheatsheet



Built-in directives

```
<section *ngIf="showSection">
```

```
<li *ngFor="let item of list">
```

```
<div [ngSwitch]="conditionExpression">
<ng-template [ngSwitchCase]="case1Exp">...</ng-template>
<ng-template ngSwitchCase="case2LiteralString">...</ng-template>
<ng-template ngSwitchDefault>...</ng-template>
</div>
```

```
<div [ngClass]="{'active': isActive, 'disabled': isDisabled}">
```

```
<div [ngStyle]="{'property': 'value'}">
<div [ngStyle]="dynamicStyles()">
```

custom directive

```
ng generate directive custom-directive
```

custom directive

```
<sizer>…<sizer>
<div sizer="72px">
   {{title}}
</div>
```

custom directive

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

@Directive({ selector: '[sizer]' })
export class SizerDirective {
…
  constructor(
   private element: ElementRef,
   private renderer: Renderer
  ) { }
}
```

custom directive

```
@Directive({ selector: '[sizer]' })
export class SizerDirective implements OnInit {
  @Input() sizer: string;

  constructor(private element: ElementRef,private renderer: Renderer)
  { }

  ngOnInit() {
    this.renderer.setElementStyle(
     this.element.nativeElement, 'font-size', this.sizer);
  }
}
```

https://github.com/Banndzior

#slack

kamil.mijacz@gmail.com

kamil.mijacz@softwarehut.com