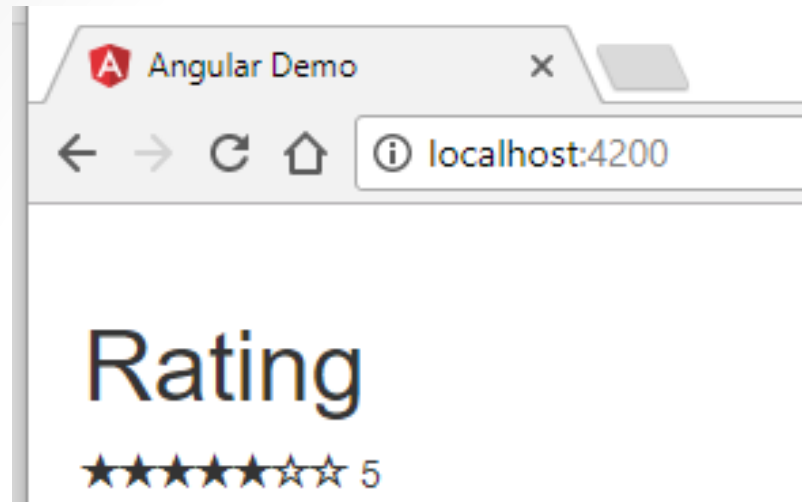
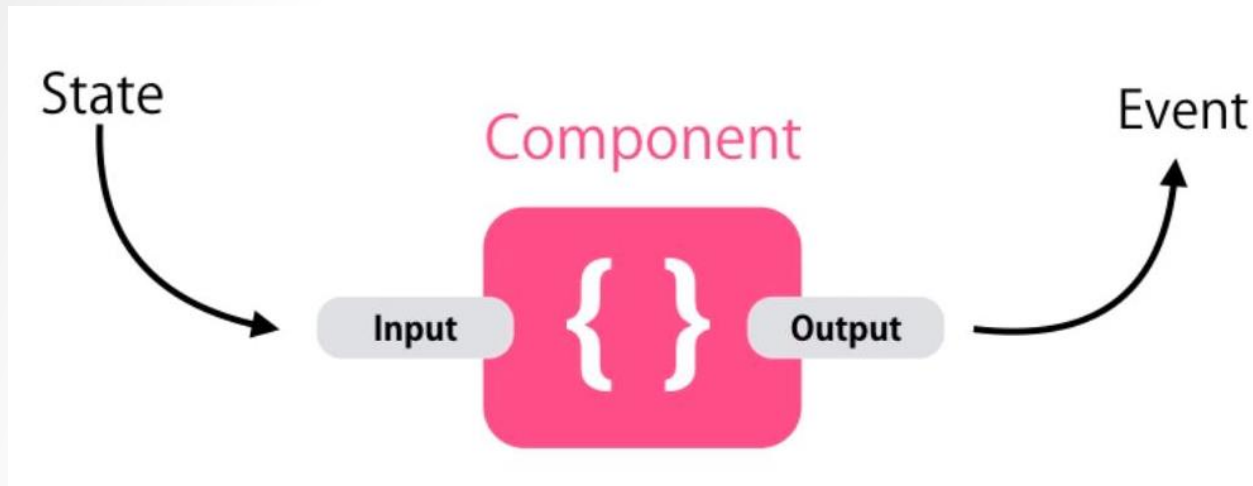


# Reusable Components



# API für Component

- PropertyBinding als Input
- EventBinding → Notification



# Rating-Component

- Verwendung in app.component
- Maximales Rating über Property max injizieren
- Verständigung bei neuem Rating über EventBinding

<> app.component.html ✕

```
1 <h1>Rating</h1>
2 <span>
3   <app-rating [max]="7" (onRating) = onRating($event) ></app-rating>
4   <span>{{actRating}}</span>
5 </span>
```

# Input max setzt das Maximum des Ratings

- Property mit Input annotieren
- Importieren

```
private _max : number = 5;  
@Input() set max(value:number) {  
    this._max = value;  
    this.actRating = Math.floor(this.max / 2);  
}  
get max(): number {return this._max}
```

# Output: Notification bei neuem Rating

- EventEmitter anlegen

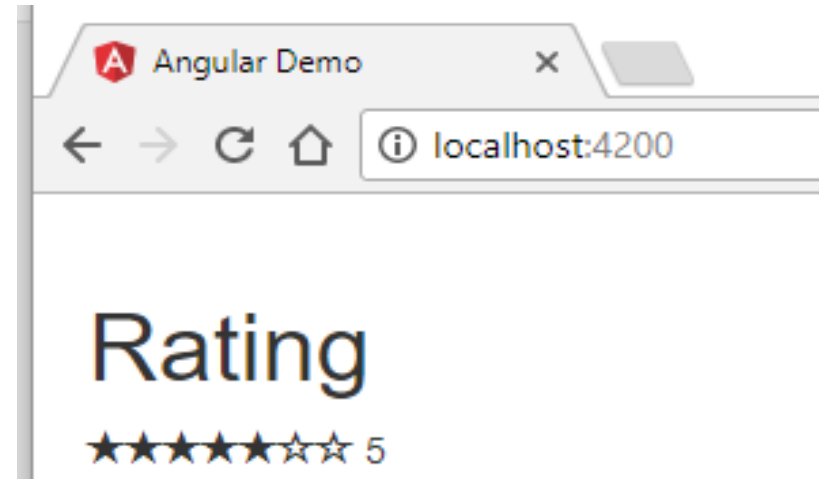
```
@Output() onRating = new EventEmitter<Number>();
```

- Bei Änderung Notification verschicken

```
set actRating(value:number) {  
  this._actRating = value;  
  this.onRating.emit(this._actRating);  
}
```

# Übung: Rating

- InputProperty
- OutputEmitter
- Glyphicon
- \*ngFor
- ClassBinding



<> rating.component.html x

```
1 <span class="glyphicon" *ngFor = "let r of ratings">
2   |   <span [ngClass]="r <= this.actRating ? 'glyphicon-star' : 'glyphicon-star-empty'"
3   |   | (click)="setRating(r)"></span>
4 </span>
```

Übung

# Ergebnis

Angular Demo

localhost:4200

## Rating

★★★★★★★☆☆ 3

Inspektor Konsole Debugger {} Stilbearbeitung @ Laufzeitanalyse Speicher Netzwerkanalyse

Ausgabe filtern

[WDS] App updated. Reloading...

undefined

Angular is running in the development mode. Call `enableProdMode()` to enable the production mode.

4

6

3

# Glyphicons - Bootstrap

## Bootstrap Glyph Example

```
<p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>
<p>Search icon: <span class="glyphicon glyphicon-search"></span></p>
<p>Print icon: <span class="glyphicon glyphicon-print"></span></p>
```

[Try it Yourself »](#)

## Available Glyphs

Glyph	Description
✱	glyphicon glyphicon-asterisk
✚	glyphicon glyphicon-plus
✚	glyphicon glyphicon-minus
€	glyphicon glyphicon-euro
☁	glyphicon glyphicon-cloud
✉	glyphicon glyphicon-envelope
✎	glyphicon glyphicon-pencil
Y	glyphicon glyphicon-glass
♪	glyphicon glyphicon-music
Q	glyphicon glyphicon-search
♥	glyphicon glyphicon-heart
★	glyphicon glyphicon-star



# Aliasing

- Sowohl für Input- als auch für Output-Properties
- Sorgen für ein stabiles Interface
  - „Magic Strings“
  - Refactoring

# View-Encapsulation – Shadow-DOM

- Scope für Styledefinitionen auf Component

```
@Component({
  selector: 'favorite',
  templateUrl: './favorite.component.html',
  styleUrls: ['./favorite.component.css'],
  encapsulation: ViewEncapsulation.
})
export class F
```

Emulated (enum member) ViewEncapsulation.Emulated  
Native  
None

# Component IsFavorite

- Wieder ein Stern mit Toggle-Switch
  - Voll, wenn IsFavorite, sonst leer
- Verständigt „Abonnennten“, wenn sich der Zustand ändert
  - Gibt den neuen Zustand mit



```
Angular is running in the development mode.  
enable the production mode.
```

```
Favorite changed: false
```

```
Favorite changed: true
```

Übung

- d

```
<span  
  class="glyphicon"  
  [class.glyphicon-star]="isFavorite"  
  [class.glyphicon-star-empty]="!isFavorite"  
  (click)="onClick()"  
></span>
```