

Motivation

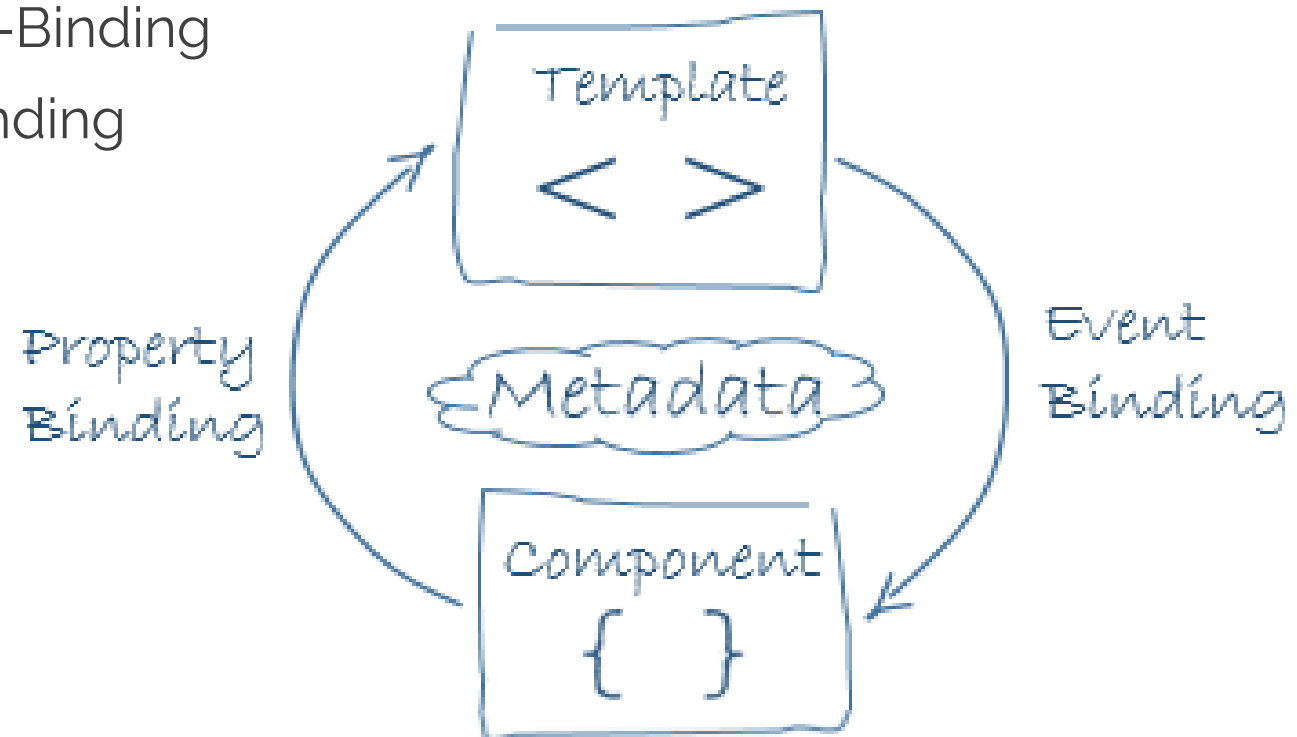
- Datenänderungen einfach in HTML widerspiegeln
- Auf Benutzerinteraktionen mit der Webseite dynamisch reagieren

Lernziele

- Sie erklären, wie Informationen dynamisch in HTML ausgegeben werden.
- Sie erklären den Mechanismus, über den eine Webanwendung auf Benutzerinteraktionen reagiert.

Data Binding

- Template zeigt statischen und dynamischen Inhalt
- Template (HTML) hat Zugriff auf Objektvariablen und Methoden der Komponentenkasse
- Komponente reagiert auf Events im Template
 - **HTML-Attribute:** Property-Binding
 - **Events in HTML:** Event-Binding



<https://angular.io/generated/images/guide/architecture/component-databinding.png>

Property-Binding

- Idee: Daten der Komponente werden im Template angezeigt und bei Änderung automatisch aufgefrischt
- Wie funktioniert diese Bindung?
 - **Richtung:** Von Komponentenkasse nach Template
 - HTML-Attribute im Template greifen auf z.B. Objektvariablen der Komponentenkasse zu
- Notation: Eckige-Klammer-Notation „[]“ im HTML-Markup

Property-Binding

```
@Component({ ... })
export class HelloWorldComponent {
  public backgroundColor: string = 'red';
  public id: string = 'hello-world-button';
  public label: string = 'Log HelloWorld!';
}

<div id="button-container"
      [style.backgroundColor]="backgroundColor">
  <button [id]="id">{{label}}</button>
</div>
```



Im Template spezifiziert

```
@Component({ ... })
```

```
export class HelloWorldComponent {  
  public backgroundColor: string = 'red';  
  public id: string = 'hello-world-button';  
  public label: string = 'Log HelloWorld!';  
}
```

```
<div id="button-container"  
  style="background-color: red;">  
  <button id="hello-world-button">Log HelloWorld!</button>  
</div>
```



Event-Binding

- Idee: Aktionen des Nutzers werden von der Komponente registriert und bearbeitet
- Wie funktioniert diese Bindung?
 - **Richtung:** Von Template nach Komponentenklasse
 - Nutzerinteraktion (Event) mit dem HTML-Dokument wird an Komponente gesendet
- Notation: Runde-Klammer-Notation „()“ im HTML-Markup
- **Beispiel:** Klickfunktion, um eine Nachricht zu zeigen

Event-Binding

```
@Component({ ... })
```

```
export class HelloWorldComponent {
```

```
  public backgroundColor: string = 'red';
```

```
  public id: string = 'hello-world-button';
```

```
  public log(valueToLog: string): void {
```

```
    console.log(valueToLog);
```

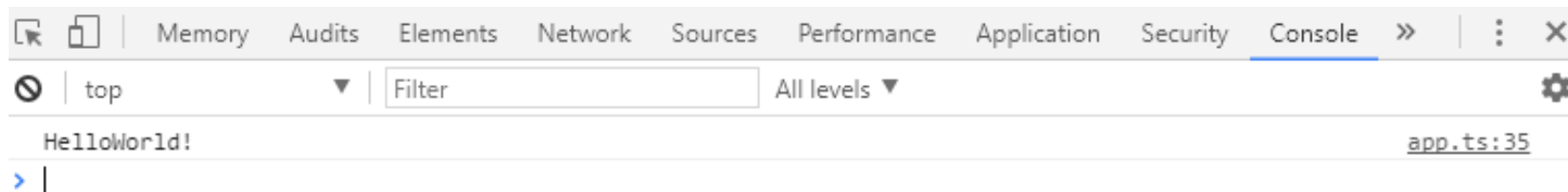
```
  }
```

```
}
```

```
<div>
```

```
  <button (click)="log('HelloWorld!')">Log HelloWorld!</button>
```

```
</div>
```



Zusammenfassung

Um was ging es in diesem Modul?

- Data Binding, Property Binding, Event Binding
- Dynamische Ausgabe von Informationen im HTML
- Funktionalität in der Komponente aus dem Template aufrufen

Wozu brauche ich das? Was werde ich damit machen?

- Daten aus der Applikation automatisch in HTML anzeigen
- Auf Nutzerhandlungen reagieren und den Status der Anwendung ändern

Kontrollfragen

- Was sind üblicherweise die Bestandteile einer Komponente?
- Welche Möglichkeiten existieren, um ein Template für eine Komponente anzugeben?
- Wo muss eine Komponente bekannt gemacht werden, damit sie im HTML-Template anderer Komponenten nutzbar ist?
- Mittels welcher Syntax können z. B. Objektvariablen einer Komponentenkasse im Template ausgegeben werden?
- Was ist der Unterschied zwischen Services und Komponenten?
- Wie kann eine Webanwendung auf Benutzerinteraktionen reagieren?

