

Animations

2. Animation state machine

State Machine

/academeeez



Animation state machine

- Each state can have styles assigned to the state
- A transition will determine a move from state to state
- Animation player will determine how a change in style is animated
- @angular/animations provides us tool to create an animation state machine
- The animation state machine is declared in the component metadata
@Component.animations
- That state machine can be placed on an element in the template

EX - our first animation state machine

- Create a component with a button and a text
- pressing the button should toggle the text and fade in/out the text



style

- contain css styling data that is associated to an animation
- the css attributes is camel cased (backgroundColor and not background-color)

```
import { style } from '@angular/animations';

style({
  opacity: 1
})
```

state

- a state has a name
- a state has an associated styles
- In our animation state machine we have one state that is active
- When a state is active the associated styles apply on the element

```
import { state, style } from '@angular/animations';  
  
state('visible', style({  
  opacity: 1  
})))
```

animate

- animate defines the following
 - the time it takes for the animation
 - the offset
 - the transition effect (ease, ease-in, ease-out, ease-in-out)

```
import { animate } from '@angular/animations';  
  
animate("1s ease-in-out")
```

transition

- represents movement from state to state
- attach an animation to that movement to determine the time, offset, and animation function of the movement

```
import { transition, animate } from '@angular/animations';  
  
transition("visible <=> invisible", animate("1s ease-in-out"))
```


trigger

- Wraps the style, state, transition, animation in a named triggered animation
- this will be placed in the **animations** of the component metadata

```
<h1 [afadeInOut]="state">
  visibility animation
</h1>

trigger("fadeOut", [
  state('visible', style({
    opacity: 1
  })),
  state('invisible', style({
    opacity: 0
  })),
  transition("visible <=> invisible", animate("1s ease-in-out")),
])
```

Summary

- An animation state machine is made from
 - states that have styles in them
 - transition that describe movement from states
 - trigger that allow us to attach the animation to an element
- Now that you know how to create the basic animation state machine let's try and create the toggle visibility component

Thank You

Next Lesson: 3. enter leave animations