Implements a domain-specific language (DSL) for defining web animation sequences for HTML elements as multiple transformations over time.

Use this API to define how an HTML element can move, change color, grow or shrink, fade, or slide off the page. These changes can occur simultaneously or sequentially. You can control the timing of each of these transformations. The function calls generate the data structures and metadata that enable Angular to integrate animations into templates and run them based on application states.

Animation definitions are linked to components through the `{@link Component.animations animations}` property in the `@Component` metadata, typically in the component file of the HTML element to be animated. The `trigger()` function encapsulates a named animation, with all other function calls nested within. Use the trigger name to bind the named animation to a specific triggering element in the HTML template.

Angular animations are based on CSS web transition functionality, so anything that can be styled or transformed in CSS can be animated the same way in Angular. Angular animations allow you to:

- Set animation timings, styles, keyframes, and transitions.
- Animate HTML elements in complex sequences and choreographies.
- Animate HTML elements as they are inserted and removed from the DOM, including responsive real-time filtering.
- Create reusable animations.
- Animate parent and child elements.

Additional animation functionality is provided in other Angular modules for animation testing, for route-based animations, and for programmatic animation controls that allow an end user to fast forward and reverse an animation sequence.

@see Find out more in the animations guide. @see See what polyfills you might need in the browser support guide.