Animated.Value

Standard value for driving animations. One `Animated.Value` can drive multiple properties in a synchronized fashion, but can only be driven by one mechanism at a time. Using a new mechanism (e.g. starting a new animation, or calling `setValue`) will stop any previous ones.

Typically initialized with `new Animated.Value(0);`

Reference

Methods

setValue()

```
setValue(value: number);
```

Directly set the value. This will stop any animations running on the value and update all the bound properties.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
value	number	Yes	Value

setOffset()

```
setOffset(offset: number);
```

Sets an offset that is applied on top of whatever value is set, whether via `setValue`, an animation, or `Animated.event`. Useful for compensating things like the start of a pan gesture.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
offset	number	Yes	Offset value

flattenOffset()

```
flattenOffset();
```

Merges the offset value into the base value and resets the offset to zero. The final output of the value is unchanged.

extractOffset()

```
extractOffset();
```

Sets the offset value to the base value, and resets the base value to zero. The final output of the value is unchanged.

addListener()

```
addListener(callback: (state: {value: number}) => void): string;
```

Adds an asynchronous listener to the value so you can observe updates from animations. This is useful because there is no way to synchronously read the value because it might be driven natively.

Returns a string that serves as an identifier for the listener.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
callback	function	Yes	The callback function which will receive an object with a `value` key set to the new value.

removeListener()

```
removeListener(id: string);
```

Unregister a listener. The `id` param shall match the identifier previously returned by `addListener()`.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
id	string	Yes	Id for the listener being removed.

removeAllListeners()

```
removeAllListeners();
```

Remove all registered listeners.

stopAnimation()

```
stopAnimation(callback?: (value: number) => void);
```

Stops any running animation or tracking. `callback` is invoked with the final value after stopping the animation, which is useful for updating state to match the animation position with layout.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
callback	function	No	A function that will receive the final value.

resetAnimation()

```
resetAnimation(callback?: (value: number) => void);
```

Stops any animation and resets the value to its original.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
callback	function	No	A function that will receive the original value.

interpolate()

```
interpolate(config: InterpolationConfigType);
```

Interpolates the value before updating the property, e.g. mapping 0-1 to 0-10.

See `AnimatedInterpolation.js`

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
config	object	Yes	See below.

The `config` object is composed of the following keys:

- `inputRange`: an array of numbers
- `outputRange`: an array of numbers or strings
- `easing` (optional): a function that returns a number, given an input number
- `extrapolate` (optional): a string such as 'extend', 'identity', or 'clamp'
- `extrapolateLeft` (optional): a string such as 'extend', 'identity', or 'clamp'
- `extrapolateRight` (optional): a string such as 'extend', 'identity', or 'clamp'

animate()

```
animate(animation, callback);
```

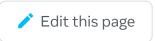Typically only used internally, but could be used by a custom Animation class.

Parameters:

NAME	TYPE	REQUIRED	DESCRIPTION
animation	Animation	Yes	See `Animation.js`.
callback	function	Yes	Callback function.

Is this page useful? 👍 👎

✏️ Edit this page

*Last updated on **Jun 21, 2023***