SVELTE • REFERENCE

svelte/transition

ON THIS PAGE

```
import {
  blur,
  crossfade,
  draw,
  fade,
  fly,
  scale,
  slide
} from 'svelte/transition';
```

blur

Animates a `blur` filter alongside an element's opacity.

```
function blur(
  node: Element,
  {
    delay,
    duration,
    easing,
    amount,
    opacity
  }?: BlurParams | undefined
): TransitionConfig;
```

crossfade

Docs

element is 'sent', it looks for a corresponding element being 'received', and generates a transition that transforms the element to its counterpart's position and fades it out. When an element is 'received', the reverse happens. If there is no counterpart, the `fallback` transition is used.

```
function crossfade({
  fallback,
  ...defaults
}: CrossfadeParams & {
  fallback?: (
    node: Element,
    params: CrossfadeParams,
    intro: boolean
  ) => TransitionConfig;
}): [
  (
    node: any,
    params: CrossfadeParams & {
      key: any;
    }
  ) => () => TransitionConfig,
  (
    node: any,
    params: CrossfadeParams & {
      key: any;
    }
  ) => () => TransitionConfig
];
```

draw

Animates the stroke of an SVG element, like a snake in a tube. `in` transitions begin with the path invisible and draw the path to the screen over time. `out` transitions start in a visible state and gradually erase the path. `draw` only works with elements that have a `getTotalLength` method, like `<path>` and `<polyline>`.

```
  {
    delay,
    speed,
    duration,
    easing
  }?: DrawParams | undefined
): TransitionConfig;
```

fade

Animates the opacity of an element from 0 to the current opacity for `in` transitions and from the current opacity to 0 for `out` transitions.

```
function fade(
  node: Element,
  { delay, duration, easing }?: FadeParams | undefined
): TransitionConfig;
```

fly

Animates the x and y positions and the opacity of an element. `in` transitions animate from the provided values, passed as parameters to the element's default values. `out` transitions animate from the element's default values to the provided values.

```
function fly(
  node: Element,
  {
    delay,
    duration,
    easing,
    x,
    y,
    opacity
  }?: FlyParams | undefined
```

Animates the opacity and scale of an element. `in` transitions animate from an element's current (default) values to the provided values, passed as parameters. `out` transitions animate from the provided values to an element's default values.

```
function scale(
  node: Element,
  {
    delay,
    duration,
    easing,
    start,
    opacity
  }?: ScaleParams | undefined
): TransitionConfig;
```

slide

Slides an element in and out.

```
function slide(
  node: Element,
  {
    delay,
    duration,
    easing,
    axis
  }?: SlideParams | undefined
): TransitionConfig;
```

BlurParams

```
interface BlurParams {…}
```

```
duration?: number;
```

```
easing?: EasingFunction;
```

```
amount?: number | string;
```

```
opacity?: number;
```

CrossfadeParams

```
interface CrossfadeParams {…}
```

```
delay?: number;
```

```
duration?: number | ((len: number) => number);
```

```
easing?: EasingFunction;
```

DrawParams

```
interface DrawParams {…}
```

```
delay?: number;
```

```
speed?: number;
```

EasingFunction

```
type EasingFunction = (t: number) => number;
```

FadeParams

```
interface FadeParams {…}
```

```
delay?: number;
```

```
duration?: number;
```

```
easing?: EasingFunction;
```

FlyParams

```
interface FlyParams {…}
```

```
delay?: number;
```

```
duration?: number;
```

```
easing?: EasingFunction;
```

```
opacity?: number;
```

ScaleParams

```
interface ScaleParams {…}
```

```
delay?: number;
```

```
duration?: number;
```

```
easing?: EasingFunction;
```

```
start?: number;
```

```
opacity?: number;
```

SlideParams

```
interface SlideParams {…}
```

```
delay?: number;
```

```
duration?: number;
```

```
easing?: EasingFunction;
```

TransitionConfig

```
interface TransitionConfig {…}
```

```
delay?: number;
```

```
duration?: number;
```

```
easing?: EasingFunction;
```

```
css?: (t: number, u: number) => string;
```

```
tick?: (t: number, u: number) => void;
```

Edit this page on GitHub

PREVIOUS
svelte/store

NEXT
Compiler errors