Magic UI Components – Compiled Reference

This is a complete, section-by-section breakdown of every Magic UI component URL provided, including installation through CLI and direct usage code, as found in the official documentation.

Theme Toggler

Installation

```
npx magicui-cli add animated-theme-toggler
```

Usage

```
import { AnimatedThemeToggler } from "@/components/magicui/animated-theme-toggler";

<AnimatedThemeToggler className="your-class" />
```

Props

Prop	Type	Default	Description
className	String	" "	Additional classes to be added to the component

Credits

- Credit to Nazam Kalsi [1]

Scroll Progress

Note

The scroll progress is shown below the navbar of the page.

Installation

```
npx magicui-cli add scroll-progress
```

Usage

```
import { ScrollProgress } from "@/components/magicui/scroll-progress";

<ScrollProgress className="your-class" />
```

Props

Prop	Type	Default	Description
className	string	-	The class name to be applied to the component

The `ScrollProgress` component also accepts all properties of the `HTMLDivElement` type.

Credits

- Credit to dipesh_the_dev

Safari

Installation

```
npx magicui-cli add safari
```

Examples

- With Image
- With Video
- Simple Mode

Usage

```
import { Safari } from "@/components/magicui/safari";

<Safari
  url="https://your-url.com"
  imageSrc="/path/image.png"
  width={1203}
  height={753}
  mode="default"
/>
```

Props

Prop	Type	Default	Description
url	string	-	The URL to display in the Safari address bar
imageSrc	string	-	The source URL of the image to display
videoSrc	string	-	The source URL of the video to display
width	number	1203	The width of the Safari window
height	number	753	The height of the Safari window
mode	SafariMode	"default"	The display mode ("default" or "simple")

The `Safari` component also accepts all properties of the `SVGElement` type.[2]

Animated Beam

Installation

```
npx magicui-cli add animated-beam
```

Examples

- Uni-Directional
- Bi-Directional
- Multiple Inputs
- Multiple Outputs

Usage

```
import { AnimatedBeam } from "@/components/magicui/animated-beam";

<AnimatedBeam
  fromRef={refA}
  toRef={refB}
  curvature={30}
  duration={2}
  pathColor="blue"
/>
```

Props

Prop	Type	Default	Description
className	string	-	The class name for the component
containerRef	ref	-	The container ref
fromRef	ref	-	The ref of the element from which the beam starts
toRef	ref	-	The ref of the element to which the beam ends
curvature	number	0	The curvature of the beam
reverse	boolean	false	Whether the beam should be reversed
duration	number	5	The duration of the beam
delay	number	0	The delay of the beam
pathColor	string	gray	The color of the beam
pathWidth	number	2	The width of the beam
pathOpacity	number	0.2	The opacity of the beam
gradientStartColor	string	"#ffaa40"	Start color of the gradient
gradientStopColor	string	"#9c40ff"	Stop color of the gradient
startXOffset	number	0	The start x offset of the beam
startYOffset	number	0	The start y offset of the beam
endXOffset	number	0	The end x offset of the beam
endYOffset	number	0	The end y offset of the beam

Credits

- Credit to @itsarghyadas for foundational work [3]

Confetti

Installation

```
npx magicui-cli add confetti
```

Examples

- Basic
- Random Direction
- Fireworks

- Side Cannons
- Stars
- Custom Shapes
- Emoji

Usage

```
import { Confetti } from "@/components/magicui/confetti";

<Confetti particleCount={100} spread={70} origin={{ y: 0.6 }} />
```

Props (Confetti)

Prop	Type	Default	Description
particleCount	Integer	50	Number of confetti particles to launch
angle	Number	90	The angle to launch confetti (degrees)
spread	Number	45	Spread (degrees) of confetti
startVelocity	Number	45	Initial velocity
decay	Number	0.9	Rate at which confetti slows down
gravity	Number	1	Gravity applied
drift	Number	0	Horizontal drift
flat	Boolean	false	If particles are flat
ticks	Number	200	Number of frames to last
origin	Object	{ x:0.5,y:0.5 }	Origin point
colors	Array of Strings	['#26ccff',...]	HEX color strings
shapes	Array of Strings	['square',...]	Shapes for confetti
zIndex	Integer	100	z-index
disableForReducedMotion	Boolean	false	Disable for reduced motion
useWorker	Boolean	true	Use Web Worker for performance
resize	Boolean	true	Resize canvas
canvas	HTMLCanvasElement/null	null	Custom canvas
scalar	Number	1	Scaling factor

Props (ConfettiButton)

Prop	Type	Default	Description
options	Object	{}	Options for the confetti
children	React.ReactNode	null	Children inside the button

Credits

- Credit to Bankk
- Inspired by canvas-confetti [4]

Blur Fade

Installation

```
npx magicui-cli add blur-fade
```

Usage

```
import { BlurFade } from "@/components/magicui/blur-fade";

<BlurFade delay={0.3} direction="down">
  <h1>Hello World ▯ Nice to meet you</h1>
</BlurFade>
```

Props

Prop	Type	Default	Description
children	React.ReactNode	-	Content to be animated
className	string	-	Class name applied
variant	object	-	Custom animation variant
duration	number	0.4	Duration (seconds)
delay	number	0	Delay (seconds) before animation
offset	number	6	Offset for animation
direction	string	"down"	Animation direction (`up`, `down`, `left`, `right`)
inView	boolean	false	Trigger animation in view
inViewMargin	MarginType	"-50px"	Margin for animation trigger
blur	string	"6px"	Blur to apply during animation

^5

Animated Shiny Text

Installation

```
npx magicui-cli add animated-shiny-text
```

Usage

```
import { AnimatedShinyText } from "@/components/magicui/animated-shiny-text";

<AnimatedShinyText shimmerWidth={180}>
  Shiny Magic Text
</AnimatedShinyText>
```

Props

Prop	Type	Default	Description
children	node	-	The text to be shimmered
className	string	-	Class applied to shimmer effect
shimmerWidth	number	100	Width of the shimmer (px)

^6

Text Reveal

Installation

```
npx magicui-cli add text-reveal
```

Usage

```
import { TextReveal } from "@/components/magicui/text-reveal";

<TextReveal>
  Magic UI will change the way you design.
</TextReveal>
```

Box Reveal Animation

Installation

```
npx magicui-cli add box-reveal
```

Usage

```
import { BoxReveal } from "@/components/magicui/box-reveal";

<BoxReveal boxColor="#5046e6" duration={0.5} className="footer-reveal">
  Reveal Content Here!
</BoxReveal>
```

Props

Prop	Type	Default	Description
className	string	-	Class applied to the component
boxColor	string	#5046e6	Color of the box overlay
duration	number	0.5	Duration (seconds) of the animation

^7

Interactive Hover Button

Installation

```
npx magicui-cli add interactive-hover-button
```

Usage

```
import { InteractiveHoverButton } from "@/components/magicui/interactive-hover-button";

<InteractiveHoverButton text="Click Me" className="important-btn" />
```

Props

Prop	Type	Default	Description
text	string	"Button"	Text inside the button
className	string	-	Additional class names to style the component

Credits

- Credit to @AayushBharti

[5]

About Magic UI

- 20+ free and open-source animated components built with React, Typescript, Tailwind CSS, and Framer Motion.
- 100% open-source and customizable.

This document includes every CLI installation and usage code snippet, as well as all important documentation details, straight from the official Magic UI websites for each specified component.

⁂

1. https://jimmysong.io/en/ai/magentic-ui/
2. https://magicui.design/docs/installation/manual
3. https://magicui.design/docs/installation
4. https://www.npmjs.com/package/magicui-cli
5. https://magicui.design/docs
6. https://github.com/magicuidesign/magicui