Annie-Mation Unleashed

Keyframe Orchestration

Welcome to the show — this is where motion stops being a reaction and becomes a performance. We're leaving simple A → B transitions behind. It's time to break free of states entirely and craft custom motion paths, timed sequences, and full-blown choreography.

Professor Solo

NotebookLM

🎭 What We'll Master

- `@keyframes`: custom motion sculpted frame-by-frame
- Multi-step sequences
- Direction, fill modes, and repetition
- Pausing, reversing, and ping-ponging
- Combining `transforms`, `opacity`, and colour shifts
- Triggering animations via `class` , events, and intersection
- Keeping motion performant
- When animation enhances UX — and when it gets in the way

If Ion Drive was about propulsion, Annie-Mation is about choreography.

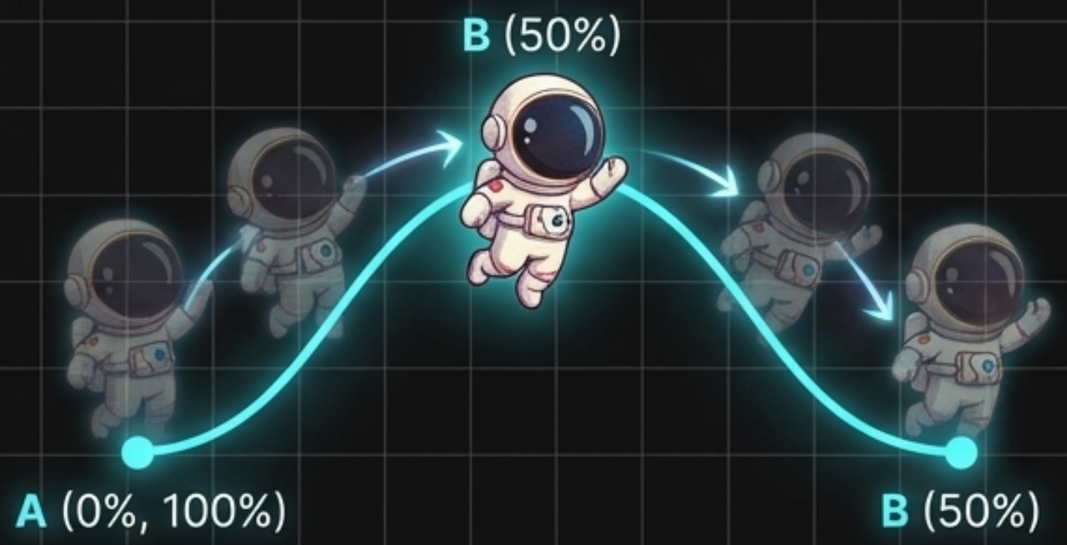
NotebookLM

The Key to Animation: Declaring “@keyframes”

> *“Transitions react. Animations perform.”*

With `@keyframes`, we design the entire journey, not just the endpoints. A keyframe block defines how something changes over time—a full motion arc.

Declaring a Keyframe



Attaching an Animation

```
@keyframes float {
  0%   { transform: translateY(0); }
  50%  { transform: translateY(-12px); }
  100% { transform: translateY(0); }
}

.orb {
  animation: float 2s ease-in-out infinite;
}
```

Links the animation name

NotebookLM

Sculpting Motion: From Keywords to Multi-Step Sequences

🧩 Using Keywords

For simple A→B sequences, you can use `from` and `to` instead of percentages. Perfect for basic fade or slide effects.

```
from {
  opacity: 0;
}
to {
  opacity: 1;
}
```

🎛️ Multi-Step Sequences

The real power comes from adding multiple steps. This is how we shape character and personality—dynamic, playful, dramatic.

```
@keyframes spinPop {
  0%   { transform: scale(0.8) rotate(0deg); }
  30%  { transform: scale(1) rotate(180deg); }
  70%  { transform: scale(1.15) rotate(270deg); }
  100% { transform: scale(1) rotate(360deg); }
}
```

NotebookLM

Chapter: Inside the Cockpit: Animation Properties

***Keyframes* define what happens.**
***Animation properties* define how, when, and for how long it happens.**

Here's the full control panel: every switch, lever, and dial we can use to shape animation behaviour. Let's run the checklist.

NotebookLM

The Control Panel: Timing & Repetition

`animation-duration`

How long one full cycle takes.

```
animation-duration: 600ms;
```

`animation-timing-function`

The easing curve for the entire animation.

```
animation-timing-function: ease-out;
```

`animation-delay`

Starts the animation after a pause.
Great for staging or staggering.

```
animation-delay: 200ms;
```

`animation-iteration-count`

How many times the animation repeats.
Can be a number or `infinite`.

```
animation-iteration-count: infinite;
animation-iteration-count: 3;
```

NotebookLM

The Control Panel: Behaviour & Flow

🔁 `animation-direction`

Controls how each iteration plays. Perfect for yo-yo, bounce, or ping-pong effects.

```
animation-direction: normal;
animation-direction: reverse;
animation-direction: alternate;
animation-direction: alternate-reverse;
```

🎭 `animation-fill-mode`

Determines how the element looks before and after the animation runs. Crucial for staged entrances.

```
animation-fill-mode: none; /* default */
animation-fill-mode: forwards; /* keep final state
animation-fill-mode: backwards; /* apply initial st
animation-fill-mode: both; /* combine both */
```

⏸ `animation-play-state`

Play and pause animations without removing them. Useful for scroll-triggered or hover-triggered sequences.

```
animation-play-state: paused;
animation-play-state: running;
```

NotebookLM

Choreography: Telling Stories with Multi-Step Animations

Multi-step animations let us: Stage reveals, build anticipation, add overshoots and rebounds, create 'stories' inside motion, and sync multiple properties. This is where Annie-Mation takes the dancefloor.

The Three-Beat Example: Small → big → settle.
Tension → anticipation → release.


```css
@keyframes glowLift {
  /* Start: Small, invisible, and below */
  0% {
    opacity: 0;
    transform: translateY(12px);
    box-shadow: 0 0 0 rgba(0,0,0,0);
  }
  /* Midpoint: Leaping, overshooting, and glowing */
  50% {
    opacity: 1;
    transform: translateY(-4px);
    box-shadow: 0 4px 14px rgba(0,0,0,0.25);
  }
  /* End: Settled, visible, and at rest */
  100% {
    opacity: 1;
    transform: translateY(0);
    box-shadow: 0 2px 8px rgba(0,0,0,0.15);
  }
}
```

NotebookLM

The Ensemble: Combining & Orchestrating Animations

`Layering Animations`

Attach multiple animation tracks by comma-separating them. Each runs independently. This is our “multi-track recording” moment.

```
.animation-stack {
  animation: float 3s infinite, glowPulse 1.4s infinite;
}
```

`Orchestrating a Scene`

Animations don’t have to be stacked—they can be coordinated across elements with different durations, easings, and delays to create one coherent scene.

```
.star { animation: twinkle 5s linear infinite; }
.comet { animation: streak 10s ease-out forwards 2s; }
.planet { animation: rotate 60s linear infinite; }
```

NotebookLM

Ready, Click, Go: Triggering Animations

Triggering is where animation becomes interactive, purposeful, and context-aware.

The Classic: Toggling a Class

Still the cleanest approach. Toggle a class with JavaScript to fire a pre-defined animation.

```css
/* CSS */
.card {
  ...
.card.reveal {
  animation: ...
}
```

```javascript
// JavaScript
card.classList.add('reveal');
```

Responding to Events

Perfect for button click feedback, panel reveals, and navigation transitions.

```javascript
button.addEventListener('click', () => {
  //... trigger animation class
});
```

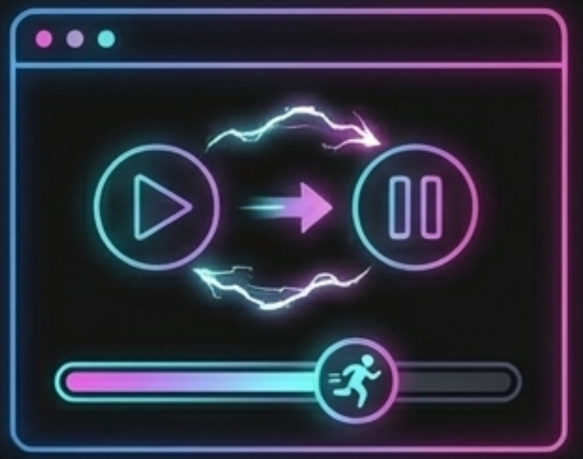
NotebookLM

Advanced Triggers & Playback Control

📜 Triggering on Scroll

A lightweight and reliable way to reveal elements as the user scrolls them into view.

```
window.addEventListener('scroll', () => {
  if (el.getBoundingClientRect().top < ...) {
    //... add animation class
  }
});
```



⏯ Play & Pause with `animation-play-state`

Gives you direct control to pause and resume animations. Perfect for scroll-controlled sequences or hold-to-activate behaviours.

```
.runner { animation-play-state: running; }
.runner.paused { animation-play-state: paused; }
runner.classList.toggle('paused');
```

🌀 The Reflow Trick

Need to rerun an animation on command? Force a browser reflow. Browser jutsu. Works every time.

```
el.classList.remove('run');
void el.offsetWidth; // Force reflow
el.classList.add('run');
```

NotebookLM

Chapter: Performance & Best Practices

Annie-Mation is powerful—too powerful if unleashed carelessly. This is the safeguard: how to keep motion fast, meaningful, and respectful of the user experience.

🚀 Stick to the Fast Lane

Stick to `transform` and `opacity` whenever possible. These properties are handled by the GPU, resulting in hardware-accelerated, buttery-smooth animations. Other properties can trigger expensive layout recalculations.

GPU Accelerated

```
transform: translate();
opacity: 0.8;
```

vs.

⚠ Layout Thrashing

```
width: 200px;
top: 20px;
left: 30px;
```

NotebookLM

The Rules of Respectful Motion

Duration Discipline

Good UI motion is 150–300ms for interaction, 400–600ms for entrances. Anything longer feels sluggish unless it's for a cinematic moment.

Avoid Motion Overload

If everything moves, nothing stands out. Use animation to reinforce hierarchy, guide attention, and enhance clarity.

Respect `prefers-reduced-motion`

Always include it. Zero effort, huge accessibility win.

```
@media (prefers-reduced-motion: reduce) {
  * {
    animation-duration: 0ms !important;
    transition-duration: 0ms !important;
  }
}
```

Test on Low-End Devices

Animations that are perfect on your high-end machine might stutter elsewhere. If it janks, simplify.

NotebookLM

The Performance Meter

NotebookLM

From Propulsion to Choreography

You've completed Annie-Mation. You now have the tools to move beyond simple state changes and layer transforms, transitions, and keyframes into expressive, modern UI experiences. You can create motion that has character, tells a story, and serves a purpose.

p.s. »

p.s., keep learning!

NotebookLM