

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width,initial-scale=1" />
<title>Happy Birthday! 🎂 — 3D Photo Cube</title>
<style>
/* ----- CSS VARIABLES (easy customization) ----- */
:root{
  --bg1: #0f1724;
  --bg2: #07101a;
  --accent: #ff00de;
  --text-color: #ffffff;
  --cube-size: 40vmin; /* try 300px or 40vmin for responsiveness */
  --cube-duration: 15s;
  --face-border: rgba(255,255,255,0.35);
  --face-bg: rgba(255,255,255,0.06);
  --glow-shadow: 0 0 20px rgba(255,0,222,0.35);
}

/* ----- Basic Page Layout ----- */
html,body{
  height:100%;
  margin:0;
  font-family: Inter, "Segoe UI", Roboto, Arial, sans-serif;
  background: radial-gradient(1200px 600px at 10% 10%, var(--bg1), var(--bg2));
  color:var(--text-color);
  -webkit-font-smoothing:antialiased;
  -moz-osx-font-smoothing:grayscale;
}

.page {
  min-height:100vh;
  display:flex;
  align-items:center;
  justify-content:center;
  gap:clamp(16px,4vw,48px);
  padding:32px;
  position:relative;
  overflow:hidden;
  flex-direction:column;
}

/* ----- Header / Message ----- */
.message {
  font-size:clamp(1.2rem, 2.3vmin, 2.25rem);
  letter-spacing:0.4px;
  text-align:center;
  text-shadow: 0 0 10px #fff, 0 0 20px var(--accent);
  animation: glow 1.6s infinite alternate;
  margin:0;
  z-index:30;
}
@keyframes glow {
  from { text-shadow: 0 0 6px #fff, 0 0 12px var(--accent);}
  to { text-shadow: 0 0 12px #fff, 0 0 28px var(--accent), 0 0 40px var(--accent); }
}

/* ----- Cube container with perspective ----- */
.cube-wrap {

```

```

width:var(--cube-size);
height:var(--cube-size);
perspective: 1200px;
position:relative;
z-index:20;
}

.cube {
width:100%;
height:100%;
position:relative;
transform-style:preserve-3d;
transition: transform 300ms ease;
animation: rotateCube linear infinite;
animation-duration: var(--cube-duration);
will-change: transform;
outline: none;
}

/* Pause rotation when hovered or keyboard-focused */
.cube-wrap:hover .cube,
.cube:focus-within {
animation-play-state: paused;
cursor:grab;
}

@keyframes rotateCube {
0% { transform: rotateX(0deg) rotateY(0deg); }
25% { transform: rotateX(90deg) rotateY(90deg); }
50% { transform: rotateX(180deg) rotateY(180deg); }
75% { transform: rotateX(270deg) rotateY(270deg); }
100% { transform: rotateX(360deg) rotateY(360deg); }
}

/* Respect users who prefer reduced motion */
@media (prefers-reduced-motion: reduce) {
.cube { animation: none; transform: none; }
.message { animation: none; }
}

/* ----- Faces ----- */
.face {
position:absolute;
width:100%;
height:100%;
left:0; top:0;
display:flex;
justify-content:center;
align-items:center;
background: var(--face-bg);
border: 2px solid var(--face-border);
box-shadow: var(--glow-shadow);
overflow:hidden;
backface-visibility: hidden;
-webkit-backface-visibility: hidden;
border-radius:8px;
transition: transform 400ms ease, box-shadow 200ms ease;
}

.face img {

```

```

width:100%;
height:100%;
object-fit:cover;
display:block;
opacity:0.98;
transform-origin:center;
user-select:none;
pointer-events:none;
}

```

```

/* Position each face in 3D */

```

```

.front { transform: translateZ(calc(var(--cube-size) / 2)); }
.back { transform: rotateY(180deg) translateZ(calc(var(--cube-size) / 2)); }
.right { transform: rotateY(90deg) translateZ(calc(var(--cube-size) / 2)); }
.left { transform: rotateY(-90deg) translateZ(calc(var(--cube-size) / 2)); }
.top { transform: rotateX(90deg) translateZ(calc(var(--cube-size) / 2)); }
.bottom { transform: rotateX(-90deg) translateZ(calc(var(--cube-size) / 2)); }

```

```

/* small hover lift effect for the face (makes clicks feel nicer) */

```

```

.face:focus,
.face:hover {
  transform: translateZ(calc(var(--cube-size) / 2)) scale(1.02);
  box-shadow: 0 8px 30px rgba(0,0,0,0.5), 0 0 40px rgba(255,0,222,0.2);
}

```

```

/* Face label overlay (optional) */

```

```

.label {
  position:absolute;
  bottom:8px;
  left:8px;
  right:8px;
  font-size:0.85rem;
  color:var(--text-color);
  text-shadow: 0 2px 6px rgba(0,0,0,0.6);
  display:flex;
  justify-content:space-between;
  gap:8px;
  pointer-events:none;
  z-index:2;
}

```

```

/* ----- Particles canvas ----- */

```

```

#particles {
  position:absolute;
  inset:0;
  z-index:0;
  pointer-events:none;
}

```

```

/* ----- Modal / Lightbox ----- */

```

```

.modal {
  position:fixed;
  inset:0;
  display:none;
  align-items:center;
  justify-content:center;
  background: linear-gradient(180deg, rgba(0,0,0,0.4), rgba(0,0,0,0.7));
  z-index:60;
}
.modal[open] { display:flex; }

```

```

.modal-content {
  max-width:min(90vw,900px);
  max-height:90vh;
  border-radius:10px;
  overflow:hidden;
  box-shadow: 0 20px 50px rgba(0,0,0,0.6);
  background:#000;
}
.modal img {
  width:100%;
  height:100%;
  object-fit:contain;
  display:block;
  background:#000;
}
.modal .close {
  position:absolute;
  top:18px; right:18px;
  background:rgba(255,255,255,0.08);
  color:#fff;
  border:0;
  padding:8px 10px;
  border-radius:8px;
  cursor:pointer;
  z-index:70;
}

/* ----- Small screens tweaks ----- */
@media (max-width:520px){
  :root { --cube-size: 70vmin; }
  .message { font-size:1.25rem; }
}
</style>
</head>
<body>
  <main class="page" role="main">
    <h1 class="message" aria-live="polite">Happy Birthday! 🎂 </h1>

    <!-- particles canvas -->
    <canvas id="particles" aria-hidden="true"></canvas>

    <!-- Cube wrapper -->
    <div class="cube-wrap" title="3D photo cube — hover to pause">
      <!-- Make cube focusable for keyboard users -->
      <div class="cube" role="list" tabindex="0" aria-label="Rotating photo cube">
        <!-- Each face is a clickable element (opens lightbox) -->
        <button class="face front" data-src="https://via.placeholder.com/800/ff7f7f/333?
text=Photo+1" aria-label="Photo 1 — front" role="listitem">
          
          <span class="label">Front <strong>Photo 1</strong></span>
        </button>

        <button class="face back" data-src="https://via.placeholder.com/800/7f7fff/333?
text=Photo+2" aria-label="Photo 2 — back" role="listitem">
          
          <span class="label">Back <strong>Photo 2</strong></span>
        </button>

        <button class="face right" data-src="https://via.placeholder.com/800/7fff7f/333?
text=Photo+3" aria-label="Photo 3 — right" role="listitem">

```

```

        
        <span class="label">Right <strong>Photo 3</strong></span>
    </button>

    <button class="face left" data-src="https://via.placeholder.com/800/ffff7f/333?
text=Photo+4" aria-label="Photo 4 — left" role="listitem">
        
        <span class="label">Left <strong>Photo 4</strong></span>
    </button>

    <button class="face top" data-src="https://via.placeholder.com/800/7fffff/333?
text=Photo+5" aria-label="Photo 5 — top" role="listitem">
        
        <span class="label">Top <strong>Photo 5</strong></span>
    </button>

    <button class="face bottom" data-src="https://via.placeholder.com/800/ff7fff/333?
text=Photo+6" aria-label="Photo 6 — bottom" role="listitem">
        
        <span class="label">Bottom <strong>Photo 6</strong></span>
    </button>
</div>
</div>

<!-- Modal / Lightbox -->
<dialog class="modal" id="modal" aria-hidden="true">
    <div class="modal-content" role="document">
        <img id="modal-img" src="" alt="Expanded photo">
    </div>
    <button class="close" aria-label="Close (Esc)" id="modal-close">X</button>
</dialog>
</main>

<script>
/* ----- Particle background (simple, performant) ----- */
(function(){
    const canvas = document.getElementById('particles');
    const ctx = canvas.getContext('2d');
    let dpr = Math.max(1, window.devicePixelRatio || 1);
    let particles = [];
    const COUNT = 40;

    function resize() {
        dpr = Math.max(1, window.devicePixelRatio || 1);
        canvas.width = innerWidth * dpr;
        canvas.height = innerHeight * dpr;
        canvas.style.width = innerWidth + 'px';
        canvas.style.height = innerHeight + 'px';
        ctx.scale(dpr, dpr);
    }

    function rand(min, max){ return Math.random()*(max-min)+min; }

    function initParticles(){
        particles = [];
        for(let i=0;i<COUNT;i++){
            particles.push({
                x: rand(0, innerWidth),
                y: rand(0, innerHeight),
                r: rand(1.5, 4.5),
            });
        }
    }

```

```

        vx: rand(-0.2, 0.6),
        vy: rand(-0.2, 0.6),
        alpha: rand(0.12, 0.5)
    });
}
}

function draw(){
    ctx.clearRect(0,0,canvas.width,canvas.height);
    for(const p of particles){
        ctx.beginPath();
        ctx.fillStyle = 'rgba(255, 0, 222,' + p.alpha + ')';
        ctx.arc(p.x, p.y, p.r, 0, Math.PI*2);
        ctx.fill();
    }
}

function update(){
    for(const p of particles){
        p.x += p.vx;
        p.y += p.vy;
        // wrap
        if(p.x > innerWidth + 10) p.x = -10;
        if(p.y > innerHeight + 10) p.y = -10;
        if(p.x < -10) p.x = innerWidth + 10;
        if(p.y < -10) p.y = innerHeight + 10;
    }
}

let raf;
function loop(){
    update();
    draw();
    raf = requestAnimationFrame(loop);
}

function start(){
    cancelAnimationFrame(raf);
    resize();
    initParticles();
    loop();
}

window.addEventListener('resize', start);
// Do not run heavy animation if user prefers reduced motion
const reduced = window.matchMedia('(prefers-reduced-motion: reduce)').matches;
if(!reduced) start();

})();

/* ----- Lightbox / Modal behavior ----- */
(function(){
    const modal = document.getElementById('modal');
    const modallmg = document.getElementById('modal-img');
    const closeBtn = document.getElementById('modal-close');

    // open modal when face clicked
    document.querySelectorAll('.face').forEach(face => {
        face.addEventListener('click', (e) => {
            const src = face.getAttribute('data-src') || face.querySelector('img').src;

```

```

    modallmg.src = src;
    modallmg.alt = face.querySelector('img').alt || 'Photo';
    modal.showModal();
    modal.setAttribute('open', '');
    modal.setAttribute('aria-hidden', 'false');
    // trap focus simply
    closeBtn.focus();
  });
});

function closeModal(){
  modal.close();
  modal.removeAttribute('open');
  modal.setAttribute('aria-hidden', 'true');
  // clear image to release memory on mobile
  modallmg.src = '';
}

closeBtn.addEventListener('click', closeModal);
modal.addEventListener('click', (ev) => {
  if(ev.target === modal) closeModal();
});

// close on Esc
document.addEventListener('keydown', (e) => {
  if(e.key === 'Escape' && modal.hasAttribute('open')) closeModal();
});
})();

/* ----- Optional: keyboard rotate control (light) ----- */
(function(){
  const cube = document.querySelector('.cube');
  if(!cube) return;
  // Allow arrow keys to nudge rotation on focus
  let rx = 0, ry = 0;
  document.querySelector('.cube-wrap').addEventListener('keydown', (e) => {
    if(e.key === 'ArrowLeft') ry -= 15;
    if(e.key === 'ArrowRight') ry += 15;
    if(e.key === 'ArrowUp') rx -= 15;
    if(e.key === 'ArrowDown') rx += 15;
    cube.style.transform = `rotateX(${rx}deg) rotateY(${ry}deg)`;
    // briefly pause animation so user sees the change
    cube.style.animationPlayState = 'paused';
    clearTimeout(cube._resumeTimeout);
    cube._resumeTimeout = setTimeout(()=> cube.style.animationPlayState = '', 1200);
  });
})();
</script>
</body>
</html>

```