</> CSS · İleri



CSS — İleri Seviye

Tema sistemi, ileri Grid, transform/animasyon ve cam efektiyle premium arayüz.

SowaCodeHK — Ders Notları
sowacodehk.com.tr · Hazar Kanat

Bu notta ne öğreneceksin? CSS'in ileri konuları: açık/koyu tema sistemi, ileri Grid teknikleri, dönüşümler (transform) ve gelişmiş animasyonlar, filtreler & efektler, ve kodu düzenli tutma yaklaşımı. "CSS — Orta" notunu bitirdiysen buraya hazırsın.

Açık / Koyu Tema Sistemi

Modern sitelerin olmazsa olmazı. CSS değişkenleri ile tek bir `data-theme` özelliğini değiştirerek tüm site rengini çevirebilirsin:

```
:root {
    --bg: #ffffff;
    --metin: #1a2233;
    --kart: #f4f6fb;
}
[data-theme="dark"] {
    --bg: #0d1526;
    --metin: #e6ecff;
    --kart: #101a33;
}

body {
    background: var(--bg);
    color: var(--metin);
}
.kart { background: var(--kart); }
```

JavaScript tarafında tek satırla tema değişir:

```
// Koyu temaya geç
document.documentElement.setAttribute("data-theme", "dark");
```

İpucu: Kullanıcının tercihini `localStorage`'a kaydet ki sayfa yenilenince teması korunsun. Ayrıca `@media (prefers-color-scheme: dark)` ile sistem temasını da algılayabilirsin.

İleri Grid: Alan İsimlendirme

Karmaşık sayfa düzenlerini isimlendirilmiş alanlarla kurmak çok okunaklıdır:

```
.sayfa {
    display: grid;
    grid-template-areas:
        "header header"
        "menu   icerik"
        "footer footer";
    grid-template-columns: 200px 1fr;
    gap: 16px;
}
.ust    { grid-area: header; }
.yan    { grid-area: menu; }
.govde  { grid-area: icerik; }
.alt    { grid-area: footer; }
```

Dönüşümler (transform)

```
.kutu:hover {
    transform: scale(1.05) rotate(2deg);   /* büyüt + hafif döndür */
}
.ikon {
    transform: translateY(-4px);   /* yukarı taşı */
}
```

- `scale()` — Boyutlandırma. `rotate()` — Döndürme.
- `translate()` — Taşıma. `skew()` — Eğme.
- Birden fazlası boşlukla birlikte yazılır: `transform: scale(1.1) rotate(5deg);`

Gelişmiş Animasyon

```
@keyframes nabiz {
    0%, 100% { transform: scale(1); opacity: 1; }
    50%      { transform: scale(1.08); opacity: 0.85; }
}
.canli-nokta {
    animation: nabiz 1.5s ease-in-out infinite;
}
```

`infinite` sonsuz döngü, `ease-in-out` yumuşak giriş-çıkış sağlar. Yükleme animasyonları, "canlı" göstergeleri hep bu mantıkla yapılır.

Filtreler ve Efektler

```
.gorsel { filter: grayscale(100%); }        /* siyah-beyaz */
.gorsel:hover { filter: none; }             /* üstüne gelince renkli */

.cam {
    background: rgba(255,255,255,0.1);
    backdrop-filter: blur(10px);            /* arkasını bulanıklaştır (cam efekti) */
}

.golge {
    box-shadow: 0 20px 40px rgba(0,0,0,0.15);
}
```

> **İpucu:** `backdrop-filter: blur()` modern "glassmorphism" (buzlu cam) görünümünü verir; premium arayüzlerde çok kullanılır.

Kodu Düzenli Tutmak

- Renk/ölçüleri `:root`'ta değişken yap; sihirli sayılar dağıtma.
- Anlamlı class isimleri kullan: `.kart-baslik`, `.menu-link`.
- Tekrarlayan stilleri ortak bir class'ta topla.
- Büyük projelerde CSS'i mantıklı dosyalara böl (temel, bileşenler, sayfa özel).

Özet

- Açık/koyu tema, CSS değişkenleri + `data-theme` ile kolayca kurulur.
- İsimlendirilmiş Grid alanları karmaşık düzenleri okunur kılar.
- `transform`, `@keyframes`, `filter` ve `backdrop-filter` premium görünümü sağlar.
- Değişkenler ve düzenli isimlendirme, sürdürülebilir CSS'in anahtarıdır.
- Bu seviyede artık modern, animasyonlu ve temalı arayüzler tasarlayabilirsin.