</> CSS · Orta



CSS — Orta Seviye

Grid, animasyon, pozisyonlama ve CSS değişkenleriyle modern, profesyonel tasarım.

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

Bu notta ne öğreneceksin? CSS'in bir üst seviyesi: CSS Grid ile sayfa düzeni, geçişler (transition) ve animasyonlar, pozisyonlama (position), CSS değişkenleri ve modern responsive teknikleri. "CSS Temelleri" notunu bitirdiysen buraya hazırsın.

CSS Grid — Güçlü Sayfa Düzeni

Flexbox tek boyutlu (satır ya da sütun) hizalama içindir. **Grid** ise iki boyutlu (satır + sütun) düzenler için idealdir — asıl sayfa iskeletini kurmakta çok güçlüdür:

```
.galeri {
    display: grid;
    grid-template-columns: repeat(3, 1fr); /* 3 eşit sütun */
    gap: 16px;
}
```

`repeat(3, 1fr)` = 3 eşit genişlikte sütun. `1fr` "bir birim boş alan" demektir. Duyarlı bir grid için:

```
.galeri {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    gap: 16px;
}
```

> **İpucu:** `auto-fit + minmax` sihirli ikilidir: sütunlar en az 200px olur, ekran genişledikçe otomatik çoğalır. Medya query yazmadan responsive grid elde edersin.

Pozisyonlama (position)

Bir elemanın sayfada nasıl konumlanacağını belirler:

- `static` — Varsayılan, normal akış.
- `relative` — Kendi yerine göre kaydırılır.
- `absolute` — En yakın konumlu atasına göre yerleşir.
- `fixed` — Ekrana sabitlenir (kaydırınca kalır — sabit menü gibi).
- `sticky` — Belli noktaya gelince yapışır (yapışkan başlık).

```
.ust-menu {
    position: sticky;
    top: 0;             /* yukarı yapışır */
    background: #fff;
    z-index: 100;       /* diğerlerinin üstünde kalır */
}
```

Geçişler (transition)

Bir özelliğin değişimini yumuşatır — ani değil, akıcı olur. Genelde `:hover` ile kullanılır:

```
.buton {
    background: #1f47e6;
    color: #fff;
    padding: 10px 20px;
    transition: all 0.3s ease;   /* tüm değişimler 0.3sn yumuşasın */
}
.buton:hover {
    background: #f0a518;
    transform: translateY(-3px);  /* hafif yukarı kalk */
}
```

Animasyonlar (@keyframes)

Sürekli ya da otomatik hareketler için animasyon tanımlanır:

```
@keyframes yukariGel {
    from { opacity: 0; transform: translateY(20px); }
    to   { opacity: 1; transform: translateY(0); }
}

.kart {
    animation: yukariGel 0.6s ease forwards;
}
```

> **İpucu:** Animasyonları abartma. Az ve yerinde kullanılan geçişler siteyi premium gösterir; her yerde dönen-zıplayan şeyler ise ucuzlatır.

CSS Değişkenleri (Custom Properties)

Renk ve ölçüleri tek yerde tanımlayıp her yerde kullanırsın. Tema yönetimini çok kolaylaştırır:

```
:root {
    --ana-renk: #1f47e6;
    --vurgu: #f0a518;
    --bosluk: 16px;
}

.buton {
    background: var(--ana-renk);
    padding: var(--bosluk);
}
.baslik { color: var(--ana-renk); }
```

Tek bir `--ana-renk` değerini değiştirince tüm site güncellenir. Açık/koyu tema da bu mantıkla yapılır.

Modern Responsive Yaklaşımı

```
/* Mobil öncelikli: önce mobil, sonra büyük ekran */
.kutular { grid-template-columns: 1fr; }        /* mobil: tek sütun */

@media (min-width: 768px) {
    .kutular { grid-template-columns: repeat(2, 1fr); }  /* tablet: 2 */
}
@media (min-width: 1024px) {
    .kutular { grid-template-columns: repeat(3, 1fr); }  /* masaüstü: 3 */
}
```

Özet

- CSS Grid, iki boyutlu düzenler için Flexbox'tan güçlüdür (`auto-fit + minmax` ile responsive).
- `position` (relative/absolute/fixed/sticky) ile hassas konumlandırma yapılır.
- `transition` yumuşak geçiş, `@keyframes` animasyon sağlar.
- CSS değişkenleri (`--degisken`) tema ve bakım kolaylığı verir.
- Mobil öncelikli responsive yaklaşımı modern standarttır.