

AI

AI DANIřMAN

ÜCRETSİZ SÜRÜM

Claude Code 101

Sohbet penceresine kod yapıştırmayı bırakmak. Claude Code dosyalarını okur, düzenler, komut çalıştırır ve işi bitirir. Bu rehber sıfırdan ilk çalışan işe kadar götürüyor.

AI Danışman – Ücretsiz Eğitim Serisi

Bu belgeyi dilediğin gibi paylaşabilirsin. Satılamaz, değiştirilemez.

Eylül 2026 · aidanisman.pages.dev · t.me/aidanisman

İÇİNDEKİLER

01 Ne olduğu — ve sohbet arayüzünden farkı

02 Kimin için

03 Kurulum ve ilk oturum

04 İlk gerçek iş: nasıl konuşulur

05 CLAUDE.md — projeye hafıza vermek

06 İzinler ve güvenlik

07 Git ile çalışmak

08 Ne zaman kullanılmaz

09 İlk yedi gün

01 Ne olduğu — ve sohbet arayüzünden farkı

Claude Code, kod tabanını okuyan, dosyaları düzenleyen, komut çalıştıran ve geliştirme araçlarıyla bütünleşen bir asistan. Terminalde, editörde, masaüstü uygulamasında ve tarayıcıda çalışıyor.

Sohbet arayüzünden farkı tek cümleyle: **sohbet sana metin verir, Claude Code işi yapar.**

	Sohbet arayüzü	Claude Code
Bağlam	Sen ne yapıştırırsan onu görür	Projeyi kendi okur, gerekli dosyayı kendi bulur
Çıktı	Metin — sen kopyalarsın	Dosyayı doğrudan değiştirir
Doğrulama	Sen çalıştırırsın	Komutu kendi çalıştırır, hatayı görür, düzeltir
Döngü	Sen ↔ pencere arası kopyala-yapıştır	Kendi içinde döngü kurar

ASIL FARK: GERİ BİLDİRİM DÖNGÜSÜ

Sohbet arayüzünde model kodu yazar ama *çalışıp çalışmadığını göremez*. Claude Code testi çalıştırıp hatayı okuyabildiği için kendi hatasını düzeltebiliyor. Kalite farkının kaynağı daha iyi bir model değil, **kapalı bir geri bildirim döngüsü**.

02 Kimin için

Adında "code" geçtiği için yazılımcılara özel sanılıyor. Kısmen doğru ama eksik.

Kim	Ne için
Yazılımcı	Asıl kitle. Özellik geliştirme, hata ayıklama, test yazma, refactor, kod inceleme
Veri ile çalışan	Dosya dönüştürme, toplu düzenleme, rapor üretme, betik yazdırma
Teknik olmayan ama meraklı	Dosya düzenleme, klasör düzeni, tekrar eden bilgisayar işleri, statik site kurma

Üçüncü satır sürpriz olabilir. Claude Code dosya okuyup yazabildiği ve komut çalıştırabildiği için, **kod yazmayan biri de kullanabiliyor** — yeter ki ne istediğini tarif edebilsin ve çıktıyı doğrulayabilsin.

03 Kurulum ve ilk oturum

Kurulum yolları

Ortam	Komut / yol
macOS, Linux, WSL	<code>curl -fsSL https://claude.ai/install.sh bash</code>
Windows PowerShell	<code>irm https://claude.ai/install.ps1 iex</code>
Homebrew	<code>brew install --cask claude-code</code>
Windows WinGet	<code>winget install Anthropic.ClaudeCode</code>
VS Code / Cursor	Eklenti mağazasından "Claude Code"
JetBrains	Marketplace eklentisi (CLI ayrıca kurulur)
Masaüstü uygulaması	İndir ve kur — CLI'ı ayrıca kurmaya gerek yok
Tarayıcı	<code>claude.ai/code</code> — hiç kurulum yok

NEREDEN BAŞLAMALI

Hiç denemediysen **tarayıcıdan** başla: kurulum yok, riski yok. Alışınca terminale geç — asıl güç orada. Homebrew ve WinGet kurulumları kendiliğinden güncellenmiyor; yerel kurulum (install.sh) arka planda güncelleniyor.

İlk oturum

Bir proje klasörüne gir ve başlat:

TERMİNAL

```
cd proje-klasorun
claude
```

İlk kullanımda giriş yapman istenir. Sonrası bir sohbet penceresi gibi görünür — ama değil: yazdığın her şey o klasör bağlamında yorumlanır.

İlk komut: soru sor, değişiklik isteme

İlk oturumda hiçbir şey değiştirtme. Önce anlatsın:

İLK ÜÇ İSTEK

Bu projede ne var? Klasör yapısını ve her parçanın ne işe yaradığını özetle.

Bu proje nasıl çalıştırılıyor? Bağımlılıklar ne, hangi komutlar var?

Bu kod tabanında en riskli veya en karışık bölüm neresi ve neden?

Üçüncü soru özellikle değerli. Kendi projende bile fark etmediğin bir şey söyleyebilir — ve sana modelin projeyi gerçekten okuyup okumadığını gösterir.

04 İlk gerçek iş: nasıl konuşulur

Claude Code ile konuşmak, sohbet arayüzüyle konuşmaktan farklı. Üç kural.

1. Sonucu tarif et, adımları değil

"Şu dosyayı aç, 42. satıra git, şunu değiştir" demek yerine ne olmasını istediğini söyle. Adımları o bulur — ve senin bilmediğin bağımlı yerleri de görür.

KÖTÜ

utils.js dosyasını aç, formatDate fonksiyonunu bul, içindeki toLocaleDateString'i değiştir.

İYİ

Tarihler her yerde Türkçe biçimde görünsün (7 Eylül 2026).
Şu an bazı yerlerde İngilizce çıkıyor. Tüm kullanım yerlerini bul ve tutarlı hâle getir.

2. Doğulamayı iste

En yüksek kaliteyi veren tek cümle bu. İşin yapıldığını değil, *çalıştığını* görmesini iste.

KALIP

[iş tarifi]

Değişikliği yaptıktan sonra testleri çalıştır ve geçtiğini göster. Test yoksa, değişikliği doğrulayan küçük bir kontrol yap ve sonucunu göster.

3. Belirsizlikte durmasını söyle

KALIP

Emin olmadığın bir karar noktasına gelirsen dur ve bana sor.
Varsayımla ilerleme – özellikle veri kaybına yol açabilecek işlerde.

Çıkarım: Sonucu tarif et, doğrulamayı iste, belirsizlikte durdur. Bu üç cümle, çıktı kalitesindeki farkın büyük kısmını açıklıyor.

05 CLAUDE.md – projeye hafıza vermek

Proje köküne koyduğun CLAUDE.md dosyası her oturumun başında okunuyor. Bu, ilk derste anlatılan "bağlam kartı"nın proje düzeyindeki karşılığı.

Ayrıca Claude Code çalışırken kendi kendine **otomatik hafıza** da biriktiriyor – oturumlar arası öğrendiklerini sen yazmadan saklıyor.

ÖRNEK CLAUDE.MD

```
# Proje adı

## Ne olduğu
Bu proje [tek cümle].

## Nasıl çalıştırılır
- Kurulum: ...
- Geliştirme: ...
- Test: ...

## Kurallar
- Yeni bağımlılık ekleme, önce sor
- Bu klasöre dokunma: [yol]
- Değişiklikten sonra testleri çalıştır
- Yorum satırlarını Türkçe yaz

## Bilinmesi gerekenler
- [Sürprizli bir davranış veya geçmiş hata]
```

EN DEĞERLİ BÖLÜM HANGİSİ

"Bilinmesi gerekenler". Kod okunarak anlaşılmayan şeyler buraya yazılır: neden böyle yapıldığı, hangi çözümün daha önce denenip başarısız olduğu, hangi kısmın görünenden riskli olduğu. Bu bölüm zamanla projenin en pahalı bilgisini biriktirir.

06 İzinler ve güvenlik

Claude Code dosya değiştirebildiği ve komut çalıştırabildiği için, izin modeli öğrenilmesi gereken ilk konulardan.

Üç temel alışkanlık

- **Git kullan.** En önemli güvenlik ağı. Değişiklik beğenilmezse geri alınır. Git olmayan bir klasörde çalıştırmadan önce yedek al.
- **İzin isteklerini okuyup onayla.** Otomatik onayı ancak ne yaptığını tam bildiğin, geri dönülebilir işlerde aç.
- **Sırları dosyaya yazdırma.** API anahtarı, şifre ve token kod içine değil ortam değişkenine girer — bunu CLAUDE.md'ye kural olarak yaz.

SİLME VE ÜZERİNE YAZMA

Geri dönüşü olmayan işlerde ("şu klasörü temizle", "eskileri sil") komutu onaylamadan önce ne sildiğini kendin gör. Git izlemeyen dosyalar (build çıktıları, yerel yapılandırma) geri gelmiyor.

07 Git ile çalışmak

Claude Code git ile doğrudan çalışıyor: değişiklikleri hazırlıyor, commit mesajı yazıyor, dal açıyor, pull request gönderiyor.

YAYGIN İSTEKLER

Değişikliklerimi anlamlı bir commit mesajıyla commit'le.

Yeni bir dal aç, şu değişikliği orada yap ve pull request açıklaması hazırla.

Son commit'ten bu yana ne değişti? Özetle.

Önerilen akış

1. Yeni bir dalda çalış — ana dalda değil
2. İş küçük parçalara böl, her parçadan sonra commit'le
3. Commit mesajını okumadan onaylama; neyin neden değiştiğini anlatmalı
4. Beğenmediğin değişikliği tartışma — geri al, yeniden tarif et

08 Ne zaman kullanılmaz

Durum	Neden
Ne istediğini bilmiyorsun	Araç hızlı ama yön senden gelir. Belirsizlik hızla çoğalır.
Çıktıyı hiç doğrulayamıyorsun	Bilmediğin bir dilde, denetleyemeyeceğin kod üretmek risk biriktirir
Canlı sistemde doğrudan	Önce yerelde veya test ortamında. Geri alınamayan işlem yaptırma.
Yedeği olmayan klasörde	Git yoksa geri dönüş yok
Mimari karar	Seçenekleri sıralatabilirsin ama kararı sen ver — sonucu sen taşıyacaksın

09 İlk yedi gün

Gün	İş
1	Kur ve sadece soru sor. Hiçbir şey değiştirtme. Projeyi anlattır.
2	Küçük ve geri alınabilir bir iş: yorum ekleme, isim düzeltme, biçimlendirme. Git'le geri almayı dene.
3	CLAUDE.md yaz. Nasıl çalıştırılır, kurallar, bilinmesi gerekenler.
4	Gerçek ama küçük bir iş. "Doğrulamayı iste" kalıbını kullan.
5	Bir hata ayıkla. Hata mesajını yapıştır, kök sebebi buldur.
6	Git akışını dene: dal aç, commit'let, farkı özetlet.
7	Bir şeyi kasten kötü tarif et ve ne olduğunu gör. Sınırı öğrenmenin en hızlı yolu.

Yedinci günün sonunda aracın ne yaptığını, nerede güçlü olduğunu ve nerede durdurulması gerektiğini bilirsin. Bu, çoğu kullanıcının aylarca ulaşamadığı yer — çünkü çoğu ilk günden büyük iş yaptırmayı deneyip hayal kırıklığına uğruyor.

Buradan sonrası

Bu rehber ilk çalışan işe kadar götürüyor. Alt ajanlar, MCP ile dış araçlara bağlanma, kendi komutlarını yazma (skills), otomatik kapılar (hooks), CLI'ı boru hattında kullanma, zamanlanmış görevler ve ekip kurulumu 'nda.

Türkçe Claude Code içeriklerini Telegram kanalında paylaşıyorum.

t.me/aidanisman · aidanisman.pages.dev

Bu belgedeki bilgiler eğitim amaçlıdır ve yazıldığı tarihteki (Eylül 2026) resmî dokümantasyona dayanır. Araç hızlı geliyor; kurulum komutları ve özellikler için güncel dokümantasyonu kontrol et.