



mirzabuilds
@MIRZABUILDS



CLAUDE CODE · 5 ADIM · 2026

Claude Code'u 10 dakikada kur

Beş adım. Kurulum, kurallar, dört klasör ve ilk test.

Claude Code'u kurmak otuz saniye sürüyor. Asıl fark kurduktan sonra ne yaptığında ortaya çıkıyor. Kurallarını yazmayan biri her oturumda aynı şeyleri baştan anlatıyor. Kurallarını yazan biri projesini tanyan bir ekiple çalışıyor. Bu rehber ikincisini kuruyor: beş adım, kopyalanabilir komutlar, hazır bir CLAUDE.md şablonu ve dört klasörün örnekli içeriği.

Sırayla git

Beş adım birbirine dayanıyor. Üçüncüyü atarsan dördüncü işe yaramıyor.

Küçük projede dene

İlk kurulumu küçük bir repoda yap, sonra asıl projene geç.

KURULUM

Beş Adım

Her adımda dört şey var: **ne yaptığı, kopyalanabilir komut, sırayla ne yapacağın ve o adımın tuzağı.**

ADIM 01 Kurulum

Claude Code terminalden çalışan bir kodlama aracı. Tek komutla kuruluyor, iki yolu var.

ÖNERİLEN, NODE GEREKMİYOR

```
curl -fsSL https://claude.ai/install.sh | bash
```

NODE KURULUYSA

```
npm install -g @anthropic-ai/claude-code
```

SIRAYLA

1. Terminali aç, iki komuttan birini çalıştır.
2. Kurulum bitince **claude** yazarak çalıştığını doğrula.
3. Sürüm kontrolü için **claude --version** yazabilirsin.

İPUCU

Kurulumu sudo ile çalıştırma. Kullanıcı dizinine kuruluyor, yönetici hakkı gerekmiyor.

TUZAĞ

İki yöntemi birden deneme. İkiisi ayrı yerlere kuruyor ve iki kurulum çakişınca hangi sürümün çalıştığını anlamak zorlaşıyor.

ADIM 02 Başlat ve doğrula

Claude'u proje klasörünün içinden başlatıyorsun. Böylece bütün kod tabanını kendisi okuyor, dosyaları tek tek yüklemen gerekmiyor.

PROJE KLASÖRÜNE GİR VE BAŞLAT

```
cd ~/projem  
claude
```

SIRAYLA

1. Komutu çalıştırınca doğrulama için tarayıcı açılıyor.
2. Hesabınla giriş yap, terminale geri dön.
3. Giriş yapıldı satırını görünce hazırsın.

İPUCU

Claude'u her zaman projenin kök klasöründen başlat. Alt klasörden başlatırsan üstteki dosyaları göremiyor.

TUZAĞ

Ev dizininde çalıştırma. Bütün diskini tarayacak bir bağlam açıyorsun ve hem yavaşlıyor hem alakasız dosyaları okuyor.

ADIM 03 Kuralları yaz

CLAUDE.md projenin hafızası. Stack'in, kuralların ve dokunulmayacak yerlerin burada duruyor. Claude her oturumda onu okuyor.

REPODAN OTOMATİK OLUŞTUR

```
/init
```

SIRAYLA

1. Claude içindeyken **/init** yaz.
2. Repoyu okuyup bir CLAUDE.md taslağı üretiyor.
3. Taslağı aç, kendi kurallarını ekle ve gereksizleri sil.

İPUCU

Kural yazarken yasak değil tercih yaz. "Şunu yapma" yerine "şu durumda şunu tercih et" daha iyi çalışıyor.

TUZAĞ

CLAUDE.md'yi şişirme. Yüzlerce satırlık dosya her oturumda bağlam yiyor ve kuralların arasında kaybolan kural uygulanmıyor.

ADIM 04 Dört klasörü kur

Projenin içinde bir .claude klasörü açıyorsun. İçindeki dört klasör asistanı, kurallarını bilen bir ekibe çeviriyor.

KLASÖRLERİ OLUŞTUR

```
mkdir -p .claude/agents .claude/commands .claude/hooks .claude/skills
```

SIRAYLA

1. **agents/** içine rol dosyaları koy: frontend, backend, test.
2. **commands/** içine tekrar eden işleri koy: /deploy, /review.
3. **hooks/** içine düzenleme öncesi ve sonrası çalışacak kontroller koy.
4. **skills/** içine gerektiğinde kendiliğinden yüklenecek tarifleri koy.

İPUCU

Dördünü birden doldurmaya çalışma. Bir tane commands dosyasıyla başla, en çok tekrar ettiğin işi oraya al.

TUZAĞ

hooks klasörüne kalıcı silme ya da otomatik push yapan bir şey yazma. Hook'lar sen onaylamadan çalışıyor.

ADIM 05 Kendi kodunla test et

Kurulum bitti, şimdi farkı görme zamanı. Küçük ama gerçek bir iş veriyorsun.

GERÇEK BİR İSTEK YAZ

```
/pricing sayfası ekle: üç paket ve ödeme butonu
```

SIRAYLA

1. Çıktıyı okuma, önce hangi dosyaları açtığına bak.
2. CLAUDE.md'yi okuduğunu görüyorsan kurulum doğru.
3. Sonucu çalıştır, build geçiyorsa adım tamam.

İPUCU

İlk testi mevcut kodunu açıklatarak yap. Ne kadarını doğru anladığını en hızlı orada görüyorsun.

TUZAĞ

İlk öğrenen büyük bir refactor isteme. Kuralların oturmadan verilen büyük iş, geri alması en zor iş oluyor.

ŞABLON

Hazır CLAUDE.md

Aşağıdakini olduğu gibi kopyala, köşeli parantezleri doldur ve projenin köküne **CLAUDE.md** adıyla kaydet. Claude her oturumda bunu okuyor.

```
# Proje: [proje adı]  
  
## Stack  
- [framework ve sürüm]  
- [veritabanı]  
- [öedeme, kimlik doğrulama, diğer servisler]  
  
## Kurallar  
- Direkt ol, kötü fikre itiraz et.  
- Emin olmadığın yerde tahmin etme, sor.  
- Büyük değişiklikten önce planı yaz, onay bekle.  
- Test yazmadan yeni özellik ekleme.  
  
## Dokunma  
- /lib/engine sormadan değiştirilmez.  
- Migration dosyaları elle düzenlenmez.  
- .env ve anahtarlar hiçbir çıktıda görünmez.  
  
## Komutlar  
- Geliştirme: [komut]  
- Test: [komut]  
- Build: [komut]
```

İPUCU

Kural sayısını on ile sınırla. Uzun dosya her oturumda bağlam yiyor ve kuralların arasında kaybolan kural uygulanmıyor.

EKİP

Dört Klasör

Projenin kökünde bir **.claude** klasörü açıyorsun. İçindeki dört klasörün her biri farklı bir işi görüyor.

agents/ • Uzmanlar

Adıyla çağırıldığın rol dosyaları. Her dosya bir uzmanın kim olduğunu ve nasıl çalıştığını anlatıyor.

örnek: frontend.md, backend.md, test.md

commands/ • Kısayollar

Tekrar eden işler. Bir kez yazıyorsun, sonra tek kelimeyle çağırıyorsun.

örnek: deploy.md, review.md, release.md

hooks/ • Güvenlik telleri

Düzenleme öncesi ve sonrası kendiliğinden çalışan kontroller. Formatlama, lint, test.

örnek: pre-edit.sh, post-edit.sh

skills/ • Oyun kitapları

Gerektiğinde kendiliğinden yüklenen tarifler. Sen çağırıyorsun, konu geçince devreye giriyor.

örnek: api-tasarımı/, migration/, hata-ayıklama/

İPUCU

Dördünü birden doldurmaya çalışma. En çok tekrar ettiğin işi bir commands dosyasına al, gerisi zamanla geliyor.

SORUN GİDERME

Tıkandığın Yerler

Bu beşi en sık karşılaşılanlar. Sırayla bak, çoğu tek ayarda çözülüyor.

Komut bulunamadı diyor

Kurulum dizini PATH'te değil. Terminali kapatıp aç, düzelmezse kurulumun yazdığı satırı kabuk profiline (.zshrc ya da .bashrc) ekle.

Tarayıcı açılmıyor ya da giriş dönmüyor

Uzak sunucuda çalışıyorsan tarayıcı açlamıyor. Terminalde çıkan bağlantıyı elle kopyalayıp kendi tarayıcında aç, dönen kodu terminale yapııştır.

Kuralları uygulamiyor

CLAUDE.md projenin kökünde mi, ona bak. Alt klasörden başlattıysan görmüyor. Bir de dosyanın çok uzun olup olmadığını kontrol et.

Çok yavaş çalışıyor

Bağlam şişmiştir. node_modules, build çıktısı ve büyük veri klasörlerini yoksay. Uzun oturumları bölmek de belirgin fark yaratıyor.

Yanlış dosyayı değiştirdi

Dokunulmayacak yerleri CLAUDE.md'ye yazmadıysan bilemiyor. Değişikliği geri al, kuralı yaz, tekrar dene. Git kullanmadan çalışma.

KAYNAKLAR

Resmî Belgeler

Hepsi Anthropic'in kendi dokümanı. Bir şey değiştğinde ilk buralara bak.

Claude Code • genel bakış	https://docs.claude.com/en/docs/claude-code/overview
Kurulum	https://docs.claude.com/en/docs/claude-code/setup
CLAUDE.md ve hafıza	https://docs.claude.com/en/docs/claude-code/memory
Alt ajanlar	https://docs.claude.com/en/docs/claude-code/sub-agents
Slash komutları	https://docs.claude.com/en/docs/claude-code/slash-commands
Hook'lar	https://docs.claude.com/en/docs/claude-code/hooks
Skill'ler	https://docs.claude.com/en/docs/claude-code/skills
Ayarlar	https://docs.claude.com/en/docs/claude-code/settings

UYARI

Beklentini Doğru Kur

DÜRÜST NOT

Claude Code ücretli planlarda sunuluyor ve kullanım limitleri var. Kurmadan önce kendi planına bak, limitler dönem dönem değişiyor.

DÜRÜST NOT

Git olmadan çalışma. Claude dosyalarını değiştiriyor ve geri almanın tek güvenli yolu sürüm kontrolü. Her oturumdan önce çalışan bir commit'in olsun.

DÜRÜST NOT

Kodu okumadan birleştirme. Kurallarını bilse bile çıktığı gözden geçiren sensin, sorumluluk sende kalıyor.

DÜRÜST NOT

Anahtarlarını ve .env dosyanı bağlam dışında tut. Yoksayma listesine ekle, çıktıda görünmesin.

DÜRÜST NOT

Bu rehber on dakikada kurulumu bitiriyor ama on dakikada uzman yapmıyor. Kurallar ilk haftada oturuyor, gerçek fark orada başlıyor.



@mirzabuilds • Claude Code kuruLumu • yoruma KURULUM yaz