

Claude için Beceri (Skill) Oluşturma

Kapsamlı Türkçe Rehber

Anthropic'in resmi rehberinden Türkiye bağlamına uyarlanan bu doküman; eğitim teknolojileri alanında yapay zekâ becerilerini nasıl oluşturacağınızı, test edip dağıtacağınızı adım adım anlatır.

İÇİNDEKİLER

- 01** Giriş: Beceri Nedir ve Neden Önemlidir?
- 02** Temel Kavramlar ve Tasarım İlkeleri
- 03** MCP ile Becerilerin Güçlü Birliği
- 04** Planlama ve Tasarım Süreci
- 05** Teknik Gereksinimler ve Yapı
- 06** Etkili Beceri Yazma Sanatı
- 07** Test, Ölçme ve İterasyon
- 08** Dağıtım ve Paylaşım Stratejileri
- 09** Kalıplar ve En İyi Uygulamalar
- 10** Sorun Giderme Rehberi
- 11** Türkiye Bağlamında Değerlendirme
- 12** Kaynaklar ve Referanslar

BÖLÜM 01

Giriş: Beceri Nedir ve Neden Önemlidir?

Şöyle bir kurum düşünün: Her hafta aynı PowerPoint şablonunu kullanarak eğitim raporları hazırlayan bir öğrenme ve gelişim ekibi. Her seferinde Claude'a aynı talimatları veriyorlar; kurumsal renk kodları, başlıkların formatı, grafiklerin yerleştirme kuralları... Her konuşmada sıfırdan başlıyorlar.

Şimdi bunu bir de şöyle düşünün: Tüm bu tercihler, süreçler ve alan uzmanlığı tek bir klasörde paketlenmiş. Claude bu klasörü bir kez okuyor ve her konuşmada bu bilgiyi otomatik olarak uyguluyor. İşte bu, bir **beceri (skill)**'dir.

- ▶ Beceri, basit bir klasör halinde paketlenmiş talimatlar setidir. Claude'a belirli görevleri veya iş akışlarını nasıl yürüteceğini öğretir. Bir kez öğretin, her kullanımda bundan faydalanın.

Beceriler Hangi Durumlarda Güçlü?

- Spec dosyalarından tutarlı ön yüz (frontend) tasarımları üretmek
- Tutarlı metodoloji ile araştırma yapmak
- Takımın stil rehberine uygun dokümanlar oluşturmak
- Çok adımlı süreçleri orkestre etmek
- MCP entegrasyonlarını güçlü iş akışlarına dönüştürmek

Tekrarlanabilir iş akışlarınız varsa — bir eğitim içerik şablonu, haftalık rapor formatı veya müşteri onboarding süreci gibi — beceriler tam size göre.

BÖLÜM 02

Temel Kavramlar ve Tasarım İlkeleri

Bir Beceri Klasörünün Anatomisi

Her beceri, belirli bir klasör yapısında organize edilir. Bu yapı hem Claude'un talimatları doğru okumasını sağlar hem de insan okuyucular için anlaşılır bir düzen sunar.

Dosya / Klasör	Durum	Açıklama
SKILL.md	Zorunlu	Ana talimat dosyası. YAML ön bilgi + Markdown.
scripts/	Opsiyone	Çalıştırılabilir kodlar (Python, Bash vb.)
references/	Opsiyone	İhtiyaç duyuldukça yüklenen ek dokümantasyon.
assets/	Opsiyone	Şablonlar, fontlar, ikonlar gibi çıktı dosyaları.

Üç Temel Tasarım İlkesi

Aşamalı Açıklama (Progressive Disclosure)

Beceriler üç katmanlı bir sistem kullanır. Birinci katman (YAML ön bilgi) her zaman Claude'un sistem prompt'una yüklenir; Claude'un beceriyi ne zaman kullanacağını bilmesi için yeterli bilgiyi içerir. İkinci katman (SKILL.md gövdesi) Claude becerinin ilgili olduğunu düşündüğünde yüklenir. Üçüncü katman (bağlantılı dosyalar) sadece ihtiyaç duyulduğunda keşfedilir. Bu yaklaşım token kullanımını en aza indirir.

Birleştirilebilirlik (Composability)

Claude aynı anda birden fazla beceri yükleyebilir. Becerinizi diğer becerilerle birlikte çalışabilmeli; tek yetenek olduğunu varsaymamalıdır. Örneğin bir doküman oluşturma becerisi, bir MCP entegrasyon becerisiyle birlikte kullanılabilir.

Taşınabilirlik (Portability)

Beceriler Claude.ai, Claude Code ve API'de aynı şekilde çalışır. Bir kez oluşturun, tüm ortamlarda değişiklik yapmadan kullanın. Becerinizin bağımlılık gerektirip gerektirmediğini compatibility alanında belirtebilirsiniz.

BÖLÜM 03

MCP ile Becerilerin Güçlü Birliği

Mutfak Analojisi

Anthropic bu ilişki için güzel bir benzetme kullanıyor: MCP profesyonel mutfağı sağlar; araçlara, malzemelere ve ekipmana erişim. Beceriler ise tarifleri sağlar; değerli bir şey oluşturmak için adım adım talimatlar. Birlikte, kullanıcıların her adımı kendileri çözmeye çalışmadan karmaşık görevleri tamamlamasını sağlarlar.

MCP (Bağlantı)	Beceri (Bilgi)
Claude'u servisimize bağlar (Notion, Asana, Linear vb.)	Claude'a servisimizi etkili kullanmayı öğretir
Gerçek zamanlı veri erişimi ve araç çağrısı sağlar	İş akışlarını ve en iyi uygulamaları yakalar
Claude ne yapabilir?	Claude nasıl yapmalı?

Beceri Olmadan vs. Beceri ile

Beceri Olmadan	Beceri ile
Kullanıcılar MCP bağlar ama ne yapacaklarını bilmez	Önceden hazırlanmış iş akışları otomatik tetiklenir
Her konuşma sıfırdan başlar	Tutarlı ve güvenilir araç kullanımı
Tutarsız sonuçlar (herkes farklı prompt yazar)	En iyi uygulamalar her etkileşime gömülür
Yüksek öğrenme eğrisi	Düşük öğrenme eğrisi

BÖLÜM 04

Planlama ve Tasarım Süreci

Kod yazmadan önce 2-3 somut kullanım senaryosu belirleyin. Bu adım beceri geliştirmenin en kritik aşamasıdır. İyi tanımlanmış bir kullanım senaryosu; tetikleyiciyi, adımları ve beklenen sonucu net şekilde ortaya koyar.

- Örnek Senaryo: Proje Sprint Planlaması — Tetikleyici: Kullanıcı 'bu sprinti planlamama yardım et' dediğinde. Adımlar: 1) Mevcut proje durumunu çek, 2) Takım hızını analiz et, 3) Görev önceliklendirmesi öner, 4) Görevleri oluştur. Sonuç: Görevleri oluşturulmuş, tamamen planlanmış sprint.

Üç Yaygın Beceri Kategorisi

Kategori 1: Doküman ve Varlık Oluşturma

Tutarlı, yüksek kaliteli çıktılar üretmek için kullanılır. PowerPoint sunumları, Word dokümanları, web arayüzleri, tasarım dosyaları gibi. Gömülü stil rehberleri, şablon yapıları ve kalite kontrol listeleri kullanır. Harici araç gerektirmez.

Kategori 2: İş Akışı Otomasyonu

Tutarlı metodoloji gerektiren çok adımlı süreçler için. Doğrulama kapılarıyla adım adım iş akışı, şablonlar ve iteratif iyileştirme döngüleri içerir. Birden fazla MCP sunucusu arasında koordinasyon sağlanabilir.

Kategori 3: MCP Geliştirme

MCP sunucusunun sağladığı araç erişimini zenginleştirmek için iş akışı rehberliği sunar. Birden fazla MCP çağrısını sırayla koordine eder, alan uzmanlığını gömer ve kullanıcıların aksi halde belirtmesi gereken bağlamı sağlar.

Başarı Kriterlerini Tanımlayın

Becerinizi çalıştığını nasıl anlayacaksınız? Anthropic hem nicel hem nitel metrikler öneriyor. Bunlar kesin eşikler değil, yaklaşık ölçütlerdir.

Nicel Metrikler	Nitel Metrikler
İlgili sorguların %90'ında beceri tetiklenmeli	Kullanıcılar sonraki adımları sormaya ihtiyaç duymamalı
İş akışı X araç çağrısında tamamlanmalı	İş akışları kullanıcı düzeltmesi olmadan tamamlanmalı
İş akışı başına 0 başarısız API çağrısı	Oturumlar arası tutarlı sonuçlar

BÖLÜM 05

Teknik Gereksinimler ve Yapı

Kritik Kurallar

! SKILL.md Adlandırma: Dosya adı tam olarak SKILL.md olmalıdır. Büyük/küçük harf duyarlıdır. SKILL.MD, skill.md gibi varyasyonlar kabul edilmez.

! Klasör Adlandırma: Kebab-case kullanın: notion-project-setup (doğru). Boşluk, alt çizgi veya büyük harf kullanmayın.

! README.md Yasak: Beceri klasörünüzün içine README.md dosyası koymayın. Tüm dokümantasyon SKILL.md veya references/ içinde olmalı.

YAML Ön Bilgi (Frontmatter)

YAML ön bilgi, Claude'un becerinizi yükleyip yüklemeyeceğine karar verdiği yerdir. Bu kısmı doğru yapmak, becerinizin başarısı için hayati önem taşır.

Alan	Durum	Açıklama
name	Zorunlu	Kebab-case, boşluk veya büyük harf yok. Klasör adıyla eşleşmeli.
description	Zorunlu	NE yaptığını VE NE ZAMAN kullanılacağını içermeli. 1024 karakter limiti.
license	Opsiyon el	Açık kaynak: MIT, Apache-2.0 gibi.
compatibility	Opsiyon el	Ortam gereksinimleri. 1-500 karakter.
metadata	Opsiyon el	Özel anahtar-değer çifti: author, version, mcp-server vb.

Güvenlik Kısıtlamaları

- ! Ön bilgide XML açılı parantezleri (< >) kullanmak kesinlikle yasaktır. Bunun nedeni ön bilginin Claude'un sistem prompt'unda görünmesidir; kötü niyetli içerik talimat enjeksiyonuna yol açabilir. Ayrıca beceri adında 'claude' veya 'anthropic' kullanmak da yasaktır; bunlar ayrılmış adlardır.

BÖLÜM 06

Etkili Beceri Yazma Sanatı

Description Alanının Önemi

Description alanı, becerinizin kalbi gibidir. Claude'un becerinizi yüklemeye karar vermesinin tek yolu bu alanı okumaktır. Yapı şu formülü izlemeli: Ne yapar + Ne zaman kullanılır + Temel yetenekler.

İyi Örnekler

- ✓ Figma tasarım dosyalarını analiz eder ve geliştirici handoff dokümantasyonu üretir. Kullanıcı .fig dosyası yüklediğinde, 'tasarım spec'leri', 'bileşen dokümantasyonu' veya 'tasarım-kod geçişi' istediğinde kullanın.
- ✓ Linear proje iş akışlarını yönetir; sprint planlama, görev oluşturma ve durum takibi dahil. Kullanıcı 'sprint', 'Linear görevleri', 'proje planlama' dediğinde veya 'bilet oluştur' istediğinde kullanın.

Kötü Örnekler

- ✗ Projelere yardımcı olur. — Çok belirsiz, tetikleyici ifade yok.
- ✗ Sofistike çok sayfalı dokümantasyon sistemleri oluşturur. — Tetikleyiciler eksik.

Talimat Yazma En İyi Uygulamaları

Spesifik ve Eyleme Geçirilebilir Olun

Doğru: 'Veri formatını kontrol etmek için python scripts/validate.py --input {dosyaadı} çalıştırın. Doğrulama başarısız olursa yaygın sorunlar şunlardır: eksik zorunlu alanlar, geçersiz tarih formatları.' Yanlış: 'Devam etmeden önce veriyi doğrulayın.'

Hata Yönetimi Ekleyin

MCP bağlantısı başarısız olursa: 1) MCP sunucusunun çalıştığını doğrulayın, 2) API anahtarının geçerli olduğunu onaylayın, 3) Yeniden bağlanmayı deneyin.

Paketlenmiş Kaynaklara Net Referans Verin

Sorgular yazmadan önce, hız sınırlaması rehberliği, sayfalama kalıpları ve hata kodları için `references/api-patterns.md` dosyasına başvurun.

Aşamalı Açıklamayı Kullanın

`SKILL.md` dosyasını temel talimatlara odaklı tutun. Ayrıntılı dokümantasyonu `references/` klasörüne taşıyın ve bağlantı verin.

BÖLÜM 07

Test, Ölçme ve İterasyon

Beceriler, ihtiyaçlarınıza göre farklı sıklık düzeylerinde test edilebilir. Claude.ai'de manuel test ile hızlı iterasyon yapabilir, Claude Code'da scriptli test ile tekrarlanabilir doğrulama sağlayabilir veya Skills API ile programatik değerlendirme paketleri oluşturabilirsiniz.

- Pro İpucu: En etkili beceri oluşturucular, Claude tek bir zorlu görevde başarılı olana kadar iterasyon yapar, sonra kazanan yaklaşımı beceriye çıkarır. Bu, Claude'un bağlam içi öğrenme yeteneğinden yararlanır ve geniş testlerden daha hızlı sinyal sağlar.

Üç Test Alanı

1. Tetikleme Testleri

Becerinizin doğru zamanlarda yüklendiğini doğrulayın. Bariz görevlerde ve farklı ifadelerle tetiklenmeli; ilgisiz konularda tetiklenmemelidir. Claude'a sorun: 'Bu beceriyi ne zaman kullanırdın?' Description'ı geri okuyacaktır.

2. Fonksiyonel Testler

Becerinizin doğru çıktılar ürettiğini doğrulayın. Geçerli çıktılar mı üretiliyor? API çağrıları başarılı mı? Hata yönetimi çalışıyor mu? Kenar durumları karşılanıyor mu?

3. Performans Karşılaştırması

Beceri ile ve beceri olmadan aynı görevin karşılaştırması. Beceri olmadan: 15 ileri-geri mesaj, 3 başarısız API çağrısı, 12.000 token. Beceri ile: 2 açıklama sorusu, 0 başarısız çağrı, 6.000 token.

Geri Bildirime Dayalı İterasyon

Sinyal	Belirti	Çözüm
Yetersiz tetiklenme	Beceri gerektiği zaman yüklenmiyor	Description'a daha fazla detay ve tetikleyici ifade ekleyin
Aşırı tetiklenme	Beceri ilgisiz sorgularda yükleniyor	Negatif tetikleyiciler ekleyin, daha spesifik olun

Yürütme sorunları	Tutarsız sonuçlar, API hataları	Talimatları iyileştirin, hata yönetimi ekleyin
-------------------	---------------------------------	--

BÖLÜM 08

Dağıtım ve Paylaşım Stratejileri

Beceriler, MCP entegrasyonunuzu daha eksiksiz kılar. Kullanıcılar bağlayıcıları karşılaştırdıkça, becerilere sahip olanlar değere daha hızlı ulaşım yolu sunar.

Dağıtım Yöntemleri

Bireysel Kullanıcılar İçin

1) Beceri klasörünü indirin, 2) Gerekiyorsa zip'leyin, 3) Claude.ai'de Ayarlar > Yetenekler > Beceriler yolunu izleyerek yükleyin. Alternatif olarak Claude Code beceri dizinine yerleştirin.

Organizasyon Düzeyi

Yöneticiler becerileri tüm çalışma alanına dağıtabilir (Aralık 2025'ten itibaren). Otomatik güncellemeler ve merkezileştirilmiş yönetim sunar.

API Üzerinden

/v1/skills endpoint'i ile programatik yönetim. Messages API isteklerine container.skills parametresiyle beceri ekleyebilirsiniz. Claude Agent SDK ile uyumludur.

Açık Standart

Anthropic, Agent Skills'i açık bir standart olarak yayınladı. MCP gibi, becerilerin araçlar ve platformlar arasında taşınabilir olması gerektiğine inanıyorlar. Aynı beceri ister Claude'da ister diğer yapay zekâ platformlarında çalışabilir.

Becerinizi Konumlandırın

Doğru Yaklaşım

Yanlış Yaklaşım

ProjectHub becerisi, takımların eksiksiz proje çalışma alanlarını saniyeler içinde oluşturmasını sağlar.

ProjectHub becerisi, YAML ön bilgi ve Markdown talimatları içeren bir klasördür.

BÖLÜM 09

Kalıplar ve En İyi Uygulamalar

Bu kalıplar erken benimseyiciler ve dahili takımlar tarafından oluşturulan becerilerden ortaya çıkmıştır. İki temel yaklaşım vardır: Problem-öncelikli ('Mutfak dolabımı tamir etmem lazım') ve Araç-öncelikli ('Yeni bir matkabım var, nasıl kullanırım?').

Kalıp 1: Sıralı İş Akışı Orkestrasyonu

Kullanıcıların belirli bir sırada çok adımlı süreçlere ihtiyaç duyduğu durumlarda kullanın. Açık adım sıralama, adımlar arası bağımlılıklar, her aşamada doğrulama ve hatalar için geri alma talimatları içerir. Örnek: Müşteri onboarding süreci — Hesap oluştur > Ödeme kur > Abonelik başlat > Karşılama e-postası gönder.

Kalıp 2: Çoklu MCP Koordinasyonu

İş akışları birden fazla servisi kapsar. Örnek: Tasarım-geliştirme geçişi — Figma'dan tasarım varlıkları çıkar > Drive'a yükle > Linear'da görevler oluştur > Slack'te bildirim paylaş. Net faz ayrımı ve merkezi hata yönetimi içerir.

Kalıp 3: İteratif İyileştirme

Çıktı kalitesinin iterasyonla arttığı durumlarda kullanın. İlk taslak > Kalite kontrolü (doğrulama scripti ile) > İyileştirme döngüsü > Sonlandırma. Açık kalite ölçütleri ve doğrulama scriptleri içerir.

Kalıp 4: Bağlam Duyarlı Araç Seçimi

Aynı sonuç, farklı bağlamlara göre farklı araçlar. Dosya tipi ve boyutuna göre en uygun depolama yeri belirlenir. Büyük dosyalar için bulut, iş birlikçi dokümanlar için Notion, kod için GitHub. Net karar ölçütleri içerir.

Kalıp 5: Alan Spesifik Zekâ

Araç erişiminin ötesinde uzmanlaşmış bilgi eklemek içindir. Örnek: Ödeme işleme öncesi uyumluluk kontrolü — yaptırım listeleri taraması, yargı bölgesi doğrulaması, risk değerlendirmesi. İşlem sonrasında denetim izi oluşturma.

BÖLÜM 10

Sorun Giderme Rehberi

Beceri Yüklenemiyor

'SKILL.md bulunamadı' hatası: Dosya adı tam olarak SKILL.md olmalı (büyük/küçük harf duyarlı). 'Geçersiz ön bilgi' hatası: YAML --- sınırlandırıcılarını kontrol edin. 'Geçersiz beceri adı': İsimde boşluk veya büyük harf kullanmayın.

Beceri Tetiklenmiyor

Description alanını gözden geçirin: Çok genel mi? Tetikleyici ifadeler içeriyor mu? Dosya tipleri belirtilmiş mi? Claude'a 'Bu beceriyi ne zaman kullanırdın?' diye sorun. Description'ı geri okuyacaktır.

Beceri Çok Sık Tetikleniyor

1) Negatif tetikleyiciler ekleyin: 'Basit veri keşfetme için kullanmayın.' 2) Daha spesifik olun: 'Dokümanları işler' yerine 'PDF hukuki dokümanlarını sözleşme incelemesi için işler.' 3) Kapsamı netleştirin.

MCP Bağlantı Sorunları

1) MCP sunucusunun bağlı olduğunu doğrulayın (Ayarlar > Uzantılar), 2) API anahtarlarının geçerli olduğunu kontrol edin, 3) MCP'yi bağımsız test edin (beceri olmadan doğrudan Claude'a MCP çağrısı yaptırın), 4) Araç adlarının büyük/küçük harf duyarlı olduğunu unutmayın.

Talimatlar Takip Edilmiyor

Yaygın nedenler: 1) Talimatlar çok uzun — kısa ve öz tutun. 2) Kritik kurallar gömülü — en önemli talimatları en üste koyun. 3) Belirsiz dil — 'Şeyleri düzgün doğrulayın' yerine açık ve net kontrol listesi verin. 4) Model tembelliğine karşı teşvik ekleyin: 'Kalite hızdan daha önemli.'

Büyük Bağlam Sorunları

Beceri içeriği çok büyükse veya aynı anda çok fazla beceri etkinse yavaşlık oluşabilir.
Çözümler: SKILL.md'yi 5.000 kelimenin altında tutun, ayrıntılı dokümantasyonu references/ altına taşıyın.

BÖLÜM 11

Türkiye Bağlamında Değerlendirme

Şöyle bir tablo düşünün: Türkiye'de kurumsal eğitim alanında çalışan bir öğrenme ve gelişim ekibi. Her ay farklı departmanlar için uyumluluk eğitimi içerikleri hazırlıyorlar, KVKK süreçlerini takip ediyorlar, çok dilli içerik üretimi yapıyorlar. Her seferinde Claude'a aynı kurumsal gereksinimleri anlatmak yerine, tüm bu bilgiyi bir beceriye dönüştürebilirler.

Türkiye'ye Özgü Kullanım Senaryoları

Kurumsal Eğitim İçerik Üretimi

Türkiye'deki büyük şirketler için harmanlanmış öğrenme içerikleri üretmek; kurumsal renk kodları, Türkçe dil kuralları, KVKK uyumlu içerik yapısı ve ölçme-değerlendirme kriterlerini gömülü olarak taşımak. Bir kez tanımlayın, her eğitim dönemine tutarlı şekilde uygulayın.

Çok Dilli İçerik Yerelleştirme

İspanyolca'dan Türkçe'ye içerik çeviri ve uyarlama süreçleri. Sadece metin çevirisi değil; kültürel bağlam, örnek senaryolar ve terminoloji yerelleştirme kurallarını beceriye gömün. Çok dilli platformlarda tutarlılık sağlayın.

KVKK Uyumluluk Dokümantasyonu

KVKK gereksinimlerine uygun eğitim materyalleri ve uyumluluk dokümanları oluşturma. Kişisel veri işleme kuralları, aydınlatma metinleri ve çalışan eğitim içerikleri için standart şablonlar tanımlayarak her seferinde aynı kalite standardını yakalayın.

Eğitim Teknolojileri Ekosistem Raporları

EdTech Türkiye gibi platformlar için; veri odaklı analiz raporları, trend değerlendirmeleri ve sektör analizleri üretmek. Editöryal ton, SEO kuralları ve hikâyeleştirme yaklaşımını beceriye gömerek her içerik üretiminde tutarlı bir ses yakalamak.

2026'ya Giderken Bu Bizi Nereye Taşıyor?

Beceriler, yapay zekânın 'genel amaçlı asistan' kimliğinden 'uzman iş ortağı' kimliğine geçişinin somut bir adımıdır. Türkiye'deki eğitim teknolojileri ekosistemi için bu, birkaç önemli anlam taşır:

- Kurumsal bilgi birikimi koruması: Deneyimli içerik tasarımcılarının bilgi ve metodolojisi becerilere aktarılabilir. Kurumsal hafıza kaybolmaz.
- Ölçeklenebilir kalite standartları: Bir kez tanımlanmış en iyi uygulamalar, takım büyüklüğünden bağımsız olarak tutarlı uygulanır.
- Ekosistem iş birliği: Açık standart sayesinde Türkiye'deki eğitim teknolojileri şirketleri, birbirleriyle uyumlu beceriler geliştirebilir.
- İnsan merkezli yapay zekâ: Beceriler teknolojiyi amaç değil, araç olarak konumlandırır. Odak davranış değişimi ve deneyim tasarımında kalır.

BÖLÜM 12

Kaynaklar ve Referanslar

Resmî Dokümantasyon

- Anthropic En İyi Uygulamalar Rehberi: docs.anthropic.com
- Beceriler Dokümantasyonu: docs.anthropic.com/skills
- API Referansı: docs.anthropic.com/api
- MCP Dokümantasyonu: modelcontextprotocol.io

Blog Yazıları

- Introducing Agent Skills — Anthropic Blog
- Equipping Agents for the Real World — Anthropic Engineering
- How to Create Skills for Claude — Anthropic Blog
- Improving Frontend Design through Skills — Anthropic Blog

Örnek Beceriler ve Araçlar

- Açık Kaynak Beceri Deposu: github.com/anthropics/skills
- skill-creator becerisi: Claude.ai'de yerleşik beceri oluşturma rehberi
- Partner Becerileri: Asana, Atlassian, Canva, Figma, Sentry, Zapier vb.

Bu rehber, Anthropic'in 'The Complete Guide to Building Skills for Claude' dokümanının Türkiye bağlamına uyarlanmış, EdTech Türkiye editöryal çerçevesiyle hazırlanmış kapsamlı Türkçe versiyonudur. Amacımız Türkiye'de eğitim teknolojileri kültürünü birlikte inşa etmektir. Üretmek. Paylaşmak. Ölçmek.

edtechturkiye.com
Birlikte öğreniyoruz, birlikte büyüyoruz.