* **Adım 1: HERKES İÇİN TEMEL DERSLER**

**Yazılım, Kodlama ve İnternet Konseptleri**

**Merhaba, Teknoloji Kaşifi!** 🚀  
Bu dersle, internet ve yazılım dünyasının gizemlerini çözmeye hazır mısın? Hayal et: İnternet dev bir şehir, yazılım ise o şehri yöneten sihirbazlar! Ama önce bu büyülü dünyayı anlamak için birkaç temel kavramı öğrenmemiz gerekiyor. Endişelenme, her şey kolay ve eğlenceli olacak! 🧙‍♀️✨

**1. Ağ ve İnternet Kavramları**

Hayal et: Ağ (network), birbirine bağlı evlerin oluşturduğu bir mahalle.  
İnternet ise bu mahalleleri birbirine bağlayan uçsuz bucaksız bir otoyol!

* **LAN (Local Area Network):** Evdeki Wi-Fi ağı, mahalle arası.
* **WAN (Wide Area Network):** Şehirlerarası otoban.
* **TCP/IP:** Postacı, mesajları doğru yere ulaştırır.
* **DNS (Domain Name System):** Arkadaşının adres defteri – web sitelerini numaralarıyla değil, isimleriyle bulmamızı sağlar!
* **HTTP/HTTPS:** Web siteleriyle konuşma dilimiz.

🎯 **Eğlenceli gerçek:** Wi-Fi’nin tam açılımı “Wireless Fidelity” değilmiş, sadece cool bir isimmiş! 😎

**2. Yazılım Nedir?**

Yazılım, bilgisayarların beynindeki büyüdür! Bilgisayarları birer kahramana dönüştürür. Yazılım ikiye ayrılır:

1. **Sistem Yazılımı:** Bilgisayarın çalışmasını sağlayan sihir (ör. işletim sistemleri).
2. **Uygulama Yazılımı:** Hayatı kolaylaştıran araçlar (ör. sosyal medya, oyunlar).

🎯 **Eğlenceli gerçek:** Tüm yazılımlar bir “Hello World!” ile başlar! Çünkü her yeni yazılımcının selamı böyledir. 👋

**3. Google Arama Teknikleri**

Google’da arama yapmak tam bir hazine avı gibidir! Ama doğru haritayı kullanmazsan, boşuna zaman kaybedebilirsin. İşte birkaç kısa tüyo:

1. **Dosya türü ara:** filetype:pdf yazarak sadece PDF belgeleri bul. 🗂️
2. **Cümle tamamla:** Unuttuğun yerlere \* koy, Google senin yerine tamamlasın!
3. **Belirli sitede ara:** site:kodluyoruz.org ile sitelerde kaybolmaya son!
4. **Eş anlamlılar:** ~yazılım yaz, Google eş anlamlılarla sana seçenek sunsun.

🎯 **Eğlenceli gerçek:** Google’ın ilk adı “BackRub” idi. Neyse ki fikir değiştirdiler! 😅

**4. Kendinizi Güncel Tutmak İçin Eğlenceli Platformlar**

Yazılım dünyasında ayakta kalmak için her zaman öğrenmeye devam etmelisin. İşte seni sıkmadan geliştirecek platformlar:

* **Stack Overflow:** Hatalarını düzeltmek için yazılımcıların buluşma noktası!
* **CodeWars:** Kodlama yeteneklerini savaşa hazırla. 🎮
* **Project Euler:** Matematik sevenlerin oyun alanı!
* **LeetCode:** Kodlama ninja’sı olmak için birebir. 🥋
* **Repl.it:** Anında kodla, hemen test et!
  1. **Stack Overflow**

Kodlama hatası mı aldın? Merak etme, dünyanın dört bir yanından gelen yazılımcılar bu platformda sana yardım etmeye hazır. Sor, cevapla veya sadece dolaş ve öğren!  
🎯 **Eğlenceli gerçek:** Stack Overflow’da günde yaklaşık 10.000 yeni soru soruluyor!

**2. HackerRank**

Kodlama yarışmalarına katıl, firmalarla bağlantı kur. Hem eğlen hem de CV’ne parlak bir yıldız ekle.  
🎯 **Eğlenceli görev:** Bir problemi çöz ve sonucunu sosyal medyada paylaş!

**3. Codility**

Şirketler genelde burayı işe alım süreçlerinde kullanır. Yani bir gün burada karşılaşacağın testleri çözmek, kariyer yolculuğunda süper bir adım olabilir!

**4. Project Euler**

Matematiği seviyorsan, bu site tam sana göre. Her problemi çözdüğünde bir zafer kazanmış gibi hissedeceksin!  
🎯 **İpucu:** Küçük bir problem seç ve kendine bir zaman sınırı koy.

**5. CodeWars**

Adeta bir dövüş arenası! Burada her seviyeden kodlama problemi seni bekliyor. Seviye atladıkça kendini daha güçlü hissedeceksin.  
🎯 **Eğlenceli görev:** En az 3 problemi çözmeden arenadan çıkma!

**6. LeetCode**

Yazılımcıların en çok tercih ettiği platformlardan biri. Özellikle büyük firmalarda çalışmak istiyorsan buradaki problemleri çözmek seni zirveye taşıyabilir.

**7. Edabit**

Başlangıç seviyesindeysen, burası tam sana göre. Kısa, eğlenceli ve öğretici görevlerle yazılıma ısınırsın.  
🎯 **Eğlenceli görev:** Her gün bir görevi çözmeye çalış!

**8. Repl.it**

Kod yaz ve anında test et! Online ortamda projeler geliştirmek için mükemmel bir yer. Ayrıca ekip arkadaşlarınla birlikte çalışabilirsin.

**9. CodePen**

Frontend geliştiricileri için yaratıcı bir alan. HTML, CSS ve JavaScript becerilerini burada sergileyebilirsin.  
🎯 **Eğlenceli gerçek:** İnsanlar genelde burada ilk “buton tasarımını” yapar!

**10. Exercism.io**

Burada kodlama öğrenirken eğlenmek garantili! Hangi programlama dilinde kendini geliştirmek istiyorsan, seni bir rehber eşliğinde ileri taşıyabilir.

🎯 **Eğlenceli gerçek:** Dünyadaki ilk kodlama dili “Ada Lovelace” isimli bir kadın matematikçiden esinlenmiştir. 💪

**Merhaba, Kodlama Kâşifi!** 🧭  
Bugün algoritmaların büyülü dünyasına adım atıyoruz. Bir algoritmayı nasıl düşüneceğini öğrendiğinde, sen de birer dijital sihirbaz olacaksın. Hadi eğlenerek öğrenelim! 🪄

**1. Algoritma Nedir?**

Bir yemek tarifi düşün: Malzemeleri sırayla koyuyor, pişiriyor ve sonunda lezzetli bir yemek elde ediyorsun. İşte algoritma da bilgisayarların "yemek tarifi"! Sadece doğru sırayla yazarsan, sonuç harika olur.

🎯 **Eğlenceli gerçek:** Bir algoritma ters yazılırsa, sabah kahvesi yerine kahve çekirdeği yiyor olabilirdik! 😅

**2. Flowgorithm Nedir?**

Algoritmayı resme dökme uygulaması. Çizim yapar gibi algoritmalarını görselleştiriyorsun!  
Ama ah, sevgili MacBook kullanıcıları, Flowgorithm şimdilik sadece Windows için. 😢 Ama üzülmeyin, alternatifler bulmak mümkün! Mesela **PSeInt**, Türkçe destekli ve Mac’te de çalışıyor. 🎉

**3. Pseudo Code Nedir?**

Gerçek kod yazmadan önce beyninin kodlama kaslarını ısıtma turu! 😎 Pseudo Code, insana özgü bir dille algoritmanı tasarlamak için kullanılan yöntemdir.

🎯 **Kısa örnek:**

markdown

Kodu kopyala

1. Kullanıcıdan bir sayı al.

2. Sayıyı 2 ile çarp.

3. Sonucu ekrana yaz.

**4. Algoritma Oluştururken Nelere İhtiyacımız Var?**

1. **Hedef:** Ne yapmak istiyorsun?
2. **Adımlar:** Hangi sırayla ilerleyeceksin?
3. **Mantık:** Adımları nasıl bağlayacaksın?

**Kendine sor:**

* "Ne yapmak istiyorum?"
* "Başlamak için neye ihtiyacım var?"
* "Sonuç doğru mu?"

🎯 **Eğlenceli görev:** Günlük işlerini algoritma gibi sıralamayı dene!

**5. Operatörler: Kodlamanın Minik Alet Çantası**

* **Aritmetik Operatörler:** Toplama (+), Çıkarma (-), Çarpma (\*), Bölme (/).
  + **Eğlenceli gerçek:** Bilgisayarlar da senin gibi matematik yapıyor!
* **Karşılaştırma Operatörleri:** Büyük mü? Küçük mü? Eşit mi? (>, <, ==).
  + **Eğlenceli görev:** Kodunda 5>3 yaz, bakalım bilgisayar doğruyu söylüyor mu?
* **Mantıksal Operatörler:** "VE" (&&), "VEYA" (||), "DEĞİL" (!).
  + **Mini oyun:** Bir kod yaz, hem tatlı hem kahve içmek isteyen birine uygun bir öneri bulsun!

**6. Değişkenler: Kodlamanın Hazine Sandığı**

Değişkenler, bilgisayarın aklında tuttuğu şeylerdir.

* **Tam sayılar:** 1, 42, -7.
* **Ondalıklı sayılar:** 3.14, 0.99.
* **Metinsel:** "Merhaba Dünya!"
* **Mantıksal:** Doğru (True) veya Yanlış (False).

🎯 **İsimlendirme ipucu:**  
Değişken isimlerini anlamlı seç:

* yas: Kullanıcının yaşını saklar.
* toplamPuan: Oyun puanlarını toplar.

🎯 **Eğlenceli gerçek:** Değişken isimlendirme bazen kodlamadan daha zor olabilir!

**7. Ah Bir de İngilizcemiz Olsaydı...**

Evet, yazılım dünyasının dili İngilizce. Ama üzülme! "If" (Eğer) ve "Else" (Aksi takdirde) gibi birkaç temel kelimeyi öğrenerek başlangıç yapabilirsin. Unutma, İngilizce yazılımda işini kolaylaştıracak bir kapıdır. 🚪

🎯 **Eğlenceli görev:** Kod yazarken sık kullanılan İngilizce kelimeler için bir liste yap ve pratik et!

**Visual Studio Code: Kodlama Dünyasına Açılan Kapı**

**Merhaba, Kodlama Kaşifi!** 🚀  
Visual Studio Code (VS Code) kodlama dünyasının sihirli çubuğu gibidir. Hem kolay kullanımı hem de havalı özellikleriyle seni yazılım dünyasında bir yıldız yapacak! Hadi şimdi eğlenceli bir şekilde neler yapabileceğini keşfedelim. ✨

**1. Visual Studio Code Nedir?**

Düşün, dev bir sihirli kitap! JavaScript’ten Python’a, C++’tan .NET’e kadar her dili öğrenebilir ve yazabilirsin. Üstelik Mac, Windows ve Linux dostudur. Yani kodlarını her yerde büyü yapar gibi yazabilirsin! 🔮

🎯 **Eğlenceli gerçek:** İlk başta biraz karmaşık görünebilir ama kullandıkça "Bu iş tamam!" diyeceksin.

**2. Proje Klasörü ve Dosyalarla Çalışma**

Tüm projelerini bir çantada topladığını hayal et. VS Code’un **Explorer** bölümü, projelerini düzenlemen için harika bir yer!

* **Dosya oluştur:** Sağ tıkla, "Yeni Dosya" de ve isim ver!
* **Arama yap:** "Search" kısmından bir kelimeyi bul ve "Replace" ile yenisiyle değiştir. (Kodluyoruz yazan her satırı "Kodluyoruz.org" ile değiştirmek gibi!)

🎯 **İpucu:** Proje dosyalarını düzgün organize edersen, kodlama maceran çok daha kolay olur!

**3. Versiyon Kontrolü ve GitHub**

**Source Control** bölümü, kodlarını yedekleyip GitHub’a göndermeni sağlar. Böylece bir hata yaptığında "Ah, keşke eski haline dönebilsem!" demezsin.

🎯 **Mini görev:** GitHub hesabı aç ve ilk kodunu paylaş! 🌟

**4. Kod Yazarken Turbo Mod: IntelliSense ve Emmet**

* **IntelliSense:** Kod yazarken seni tamamlayan bir kod dostu. Fonksiyonları, özellikleri ve daha fazlasını gösterir. 🧠
* **Emmet:** "Kısa yoldan kral olurum!" dedirten bir özellik. Örneğin div>ul>li\*3 yazıp Tab tuşuna bas, bir anda 3’lü liste hazır!

🎯 **Eğlenceli görev:** Bir HTML sayfası aç ve Emmet ile şablon oluştur.

**5. Eklentiler: VS Code’un Sihirli Gücü**

VS Code’u özelleştirmek için eklentilere ihtiyacın var. Ama dikkat! Çok fazla eklenti bilgisayarını yavaşlatabilir.

* **Live Server:** HTML sayfanı anında görüntüle.
* **Prettier:** Kodlarını otomatik düzenle.  
  🎯 **İpucu:** Eklentileri yüklerken açıklamalarını oku ve işine yarayanları seç!

**6. Kısayollar: Kodlama Süper Gücü**

Kodlama hızını artırmak için bazı kısayollar:

* **CTRL + P:** İstediğin dosyayı aç.
* **CTRL + Tab:** Dosyalar arasında geçiş yap.
* **CTRL + S:** Kodlarını kaydet, unutmamak için harika bir yol.

🎯 **Eğlenceli gerçek:** Kısayolları öğrenmek seni kodlama dünyasında bir ninja yapar! 🥷

**7. Terminal: Kodunla Konuşmanın Yolu**

Terminal, bilgisayarına emir verdiğin yerdir. Komutları öğrenmek içpin küçük bir maceraya atılmaya hazır mısın?

* **pwd:** Hangi klasörde olduğunu gösterir.
* **cd [klasörismi]:** Belirtilen klasöre gir.
* **mkdir [isim]:** Yeni bir klasör oluştur.

🎯 **Mini görev:** Bir klasör oluştur, içine bir dosya ekle ve terminalde hepsini listele!

**8. MultiCursor: Çoklu İmleç ile Zamandan Tasarruf**

Aynı kelimeyi birden fazla yerde mi değiştireceksin? CTRL + D ile hepsini seç ve aynı anda düzenle!

🎯 **Deneme:** "Kodluyoruz" yazan tüm satırları "Kodluyoruz.org" ile değiştir. Bir sihir gibi, değil mi? ✨

**9. Snippets: Kodlama İçin Hızlı Notlar**

Kodda sıkça kullandığın uzun bir ifadeyi kısa bir komutla çağırmak için **Snippets** kullanabilirsin. Örneğin, <h1>Kodluyoruz</h1> yerine "k1" yazıp Tab tuşuna bas!

🎯 **Görev:** Kendi snippets’ını oluştur ve her projede kullan.

**10. Diğer Kod Editörleri**

VS Code harika ama başka alternatifler de var:

* **Sublime Text:** Hafif ve hızlı bir editör.
* **Atom:** GitHub’ın geliştirdiği havalı bir araç.
* **IntelliJ IDEA:** Büyük projeler için güçlü ama biraz RAM canavarı.

🎯 **Eğlenceli görev:** Birini indir ve küçük bir proje başlat.

**VS Code’a Hazır mısın?** 🎉  
Şimdi VS Code dünyasında kaybolmaya ve kendi projeni oluşturmaya başlayabilirsin. Unutma, öğrenirken eğlenmek senin en büyük gücün! 🚀

Tabii! İşte **GIT** ders notlarını daha eğlenceli ve anlaşılır bir hale getirdiğim örnek:

# 🚀 **Merhaba, Teknoloji Kaşifi!** 🚀

Bugün GIT dünyasını keşfetmeye hazır mısın? GIT'i anlamak, yazılım dünyasının "zaman makinesi" kullanabilmek gibi! Hadi başlayalım. 💻✨

## 1. GIT Nedir?

Bir nevi **versiyon kontrol sistemi**! 🎯  
Hayal et: Bir kitabı her yazdığında, eski versiyonlarını da tarihsel kayıt defteri gibi saklayabilen bir sistem var. İşte GIT tam olarak bunu yapar:  
📌 **GIT**, dosyalarını ve projelerini:

* Takip eder.
* Sürüm oluşturur (versiyonlar).
* Geri dönmeni sağlar (hata mı yaptın? Sıkıntı yok!).

### 🧠 **Kültürel Bir Bakış**:

“GIT” kelimesi aslında "küçük, işe yaramaz bir şey" anlamına gelir. Linus Torvalds (Linux’un yaratıcısı), GIT’i yazarken şaka yollu bu ismi koymuş. 😄

## 2. GIT Ne Sağlar?

🎁 GIT bir hediye gibi:

* 📂 **Dağıtık Depolama**: Dosyalarını birçok yerde tutar. Cihaz bağımsız erişebilirsin.
* 📸 **Snapshot (Anlık Görüntü)**: İstediğin anda projeyi kayıt altına al! ⏳
* 🤝 **Takım Çalışması**: Kim ne yaptı? Ne ekledi, ne sildi? GIT her şeyi sana raporlar.
* 🗑️ **İstemediğin Dosyaları Dışlama**: Bazı dosyalar versiyonlanmasın diyorsan, .gitignore kullan.

## 3. ****GIT mi, GitHub mı? 🤔****

* **GIT** → Sadece versiyon kontrol sistemi.
* **GitHub** → GIT’i kullanan, sosyal ve online bir depolama hizmeti. 💡

| **Servis** | **Özellik** |
| --- | --- |
| GitHub | Sosyal medya gibi. Projelerini paylaş. |
| GitLab | Şirketler kendi sunucularında kullanır. 🏢 |
| BitBucket | Daha kişisel. Küçük projeler için idealdir. |

## 4. ****Mac vs Windows GIT Kurulumu 🌐****

* **Mac**: Terminal aç → git yaz → Çalışıyorsa kurulu demektir. 👌  
  (Not: GIT, uygulama olarak görünmez. Çünkü terminal üzerinden çalışır!)
* **Windows**: GIT Bash kullan! Kurulum sonrası komutları GIT Bash’te çalıştır. 🖥️

## 5. ****GIT Temel Terimler ve Komutlar****

| **Terim** | **Açıklama** |
| --- | --- |
| **untracked** | Takip edilmeyen yeni dosya. |
| **unstaged** | Hazırlanmamış, güncellenmiş dosya. |
| **staged** | Commit’e hazır dosya. |
| **deleted** | Silinmiş ama GIT’ten kaldırılmamış dosya. |

### 🎯 **En Önemli Komutlar:**

1. 🛠️ **git init**: Yeni bir GIT deposu oluştur.
2. ➕ **git add**: Dosyaları staged ortama gönder.
3. ✅ **git commit**: Değişiklikleri kaydet. **-m "mesaj"** ekle.
4. 🔍 **git status**: Projenin durumunu gör.
5. 🔁 **git push**: Commit’lerini uzak sunucuya gönder.
6. 📥 **git pull**: Uzak sunucudan güncellemeleri çek.
7. 📦 **git clone**: Bir projeyi kopyala.
8. 🔄 **git merge**: Branch’leri birleştir.
9. 🗑️ **git rm**: Dosya takibini bırak/sil.
10. 🕵️ **git log**: Commit geçmişini görüntüle.

GIT'i bir süre kullandıktan sonra:

* Hatalardan korkmayacaksın. 💪
* Tüm projelerin "kontrol altında" olacak.

🚀 Hadi GIT kullanmaya başla ve yazılım dünyasına adım at!

## 🧠 ****GIT İle İlgili Daha Derin Bilgiler ve Notlar****

### 1. **Branching (Dallanma) Sistemi** 🌱

GIT'in en güçlü özelliklerinden biri **branch** (dal) yapısıdır. Ana projeni bozmadan yeni özellikler üzerinde çalışabilirsin.

**Önemli Branch Komutları:**

* git branch <branch-ismi> → Yeni bir branch oluştur.
* git branch → Tüm branch’leri listele.
* git checkout <branch-ismi> → Branch’e geçiş yap.
* git merge <branch-ismi> → Başka bir branch’ı mevcut branch’e birleştir.
* git branch -d <branch-ismi> → Branch’i sil.

📌 **Not**:

* **master/main** → Ana branch.
* Yeni geliştirme branch'leri feature/ ya da bugfix/ gibi isimlendirilir.

### 2. **Remote Repository (Uzak Depolar)** 📤

Projeni **uzak sunucu** ile senkronize ederek ekip çalışması yaparsın.

**Remote Komutları:**

* git remote add origin <repo-adresi> → Remote repo’yu ekle.
* git push origin <branch-ismi> → Branch’i uzak sunucuya gönder.
* git pull origin <branch-ismi> → Uzak sunucudan branch’i çek.
* git fetch → Uzak sunucudaki değişiklikleri indir ama projeye yansıtma.

### 3. **Conflict (Çakışma) Çözme** ⚔️

Aynı dosyada farklı kişiler değişiklik yaparsa **merge conflict** oluşur. Çakışmayı çözmek için:

1. git status ile çakışan dosyaları gör.
2. Dosyayı aç ve GIT'in eklediği işaretler arasında tercih yap.

plaintext

Kodu kopyala

<<<<<<< HEAD

Senin değişikliğin

=======

Başka kişinin değişikliği

>>>>>>> branch-ismi

1. Çakışmayı düzelt, dosyayı kaydet.
2. git add ve git commit ile çözümü tamamla.

### 4. **Stash (Geçici Kaydetme)** 📝

Çalışırken bir anda başka bir branch’e geçmen gerekiyorsa:

* git stash → Değişiklikleri saklar.
* git stash list → Saklanan değişiklikleri listeler.
* git stash apply → Saklanan değişiklikleri geri yükler.
* git stash drop → Saklanan değişiklikleri siler.

### 5. **Revert vs Reset vs Amend** 🕒

Bazen değişiklikleri geri almak istersin:

* git revert <commit-id> → Eski bir commit’i **tersine çevirir**, yeni bir commit oluşturur.
* git reset --hard <commit-id> → Değişiklikleri tamamen geri alır (dikkatli ol!).
* git commit --amend → Son commit mesajını veya dosyalarını düzenler.

### 6. **Git Ignore Dosyası** 📋

Versiyon kontrolüne dahil edilmesini istemediğin dosyaları .gitignore dosyasına ekle:

plaintext

Kodu kopyala

# .gitignore örneği

node\_modules/

\*.log

\*.env

.DS\_Store

### 7. **Tag (Etiketleme)** 🏷️

Versiyon oluştururken önemli commit’leri **etiketle**.

* git tag <etiket-ismi> → Yeni bir etiket oluştur.
* git tag → Etiketleri listele.
* git push origin <etiket-ismi> → Etiketi uzak sunucuya gönder.

### 8. **Alias (Kısa Yol Oluşturma)** 🛠️

Sürekli aynı komutları yazmak yerine kısa yollar oluştur:

bash

Kodu kopyala

git config --global alias.st status

git config --global alias.ci commit

git config --global alias.co checkout

Artık git status yerine git st yazabilirsin! 🎉

## 📝 ****Profesyonel Notlar****

1. **Küçük ve Sık Commit'ler Yap**: Her değişiklikte açıklayıcı bir commit mesajı kullan.
2. **Branch Kullanımı**: Ana branch’i temiz tut; geliştirme için ayrı branch oluştur.
3. **Commit Mesajı Kuralları**:
   * İlk satır kısa ve öz olsun.
   * Ne yaptığını açıkla: "fix: hatalı form gönderimi düzeltildi" gibi.
4. **Kod İncelemesi (Pull Request)**: Büyük projelerde kodu birleştirmeden önce gözden geçir.

## 🎯 ****Özet ve İpuçları****

GIT, basit gibi görünse de profesyonelleştikçe ihtiyaçların artar.

* Projelerini bölümlere ayırmak için **branch** kullan.
* Değişiklikleri kayıt altına almak için **commit** yapmayı alışkanlık edin.
* Hatalarda eski versiyonlara kolayca dönmek için **snapshot** mantığını anla.