Python ile Kodlama Macerasına İlk Adım

Programlama dünyasını keşfetmek için bir başlangıç rehberi.

HALUK TANRIKULU - Pitonca.com

Haluk Tanrıkulu'nun 'Python ile Programlamaya Giriş' adlı eserinden uyarlanmıştır.

Pitonca Education

PYTHON PROGRAMLAMA - ISBN No: 978-625-00-2406-5

Neden Python? Modern Dünyayı Şekillendiren Dil

1991 yılında Guido van Rossum tarafından geliştirilen Python, temiz ve okunabilir sözdizimi ile karmaşık işlemleri basitleştirir. Adını Monty Python komedi grubundan alan bu dil, programlamaya yeni başlayanlar için ideal bir başlangıç noktasıdır.

- **Kolay Öğrenilir**: Açık ve anlaşılır sözdizimi sayesinde hızlı bir başlangıç sağlar.
- **Çok Yönlü**: Web geliştirme, veri bilimi, yapay zeka ve otomasyon gibi sayısız alanda kullanılır.
- **Geniş Topluluk**: Herhangi bir sorun karşısında destek bulmayı kolaylaştıran devasa bir kullanıcı topluluğu ve kütüphane desteği sunar.
- **Platform Bağımsızlığı**: Windows, macOS ve Linux gibi farklı işletim sistemlerinde sorunsuzca çalışır.

Maceraya Hazırlık: Kurulum ve İlk Komutunuz

Python'u kullanmaya başlamak için ilk adım, onu sisteminize kurmaktır. İşletim sisteminize göre aşağıdaki basit adımları izleyebilirsiniz.

Windows

1. `python.org` adresinden en son sürümü indirin.
2. Yükleyiciyi çalıştırırken 'Add Python to PATH' seçeneğini işaretleyin.

macOS

1. `python.org` üzerinden indirin veya Terminal'de `brew install python` komutunu kullanın.

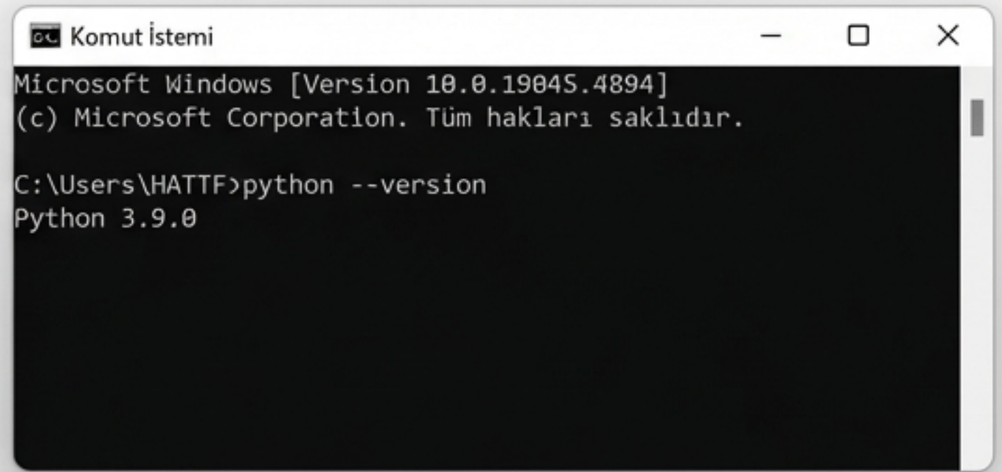
Linux

1. Çoğu dağıtımda kurulu gelir. Değilse, Terminal'de `sudo apt-get install python3` komutunu çalıştırın.

Kurulumu kontrol etmek için Komut İstemi'ni veya Terminal'i açın ve şu komutu yazın:

```
python --version
```

Dikkat! Bilgisayarınızın 32 bit mi 64 bit mi olduğunu önceden bilmeli ve Python'ı ona göre indirmelisiniz.



Bilginin Yapı Taşları: Değişkenler ve Veri Tipleri

Python'da değişkenler, verileri saklamak için kullanılan kaplardır. Tür belirtmeden tanımlanabilirler; Python, atanan değere göre türü dinamik olarak belirler.

int (Tam Sayı)

```
sayi = 1000
```

float
(Ondalıklı Sayı)

```
pi = 3.14
```

complex
(Karmaşık Sayı)

```
z = 2 + 3j
```

Metin
(String)
Tek veya çift
tırnak ile tanımlanır.

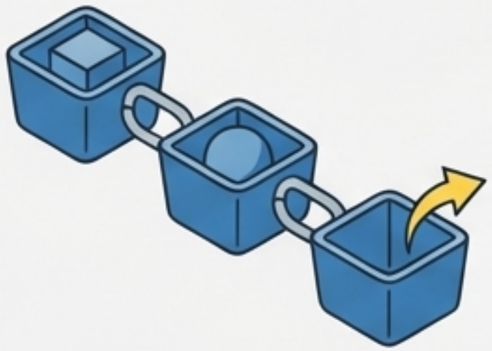
```
mesaj = "Merhaba Dünya"
```

Boolean
Mantıksal ifadeler için sadece iki değer alır.

```
dogru_mu = True
yanlis_mi = False
```

Veri Koleksiyonları: Bilgiyi Düzenleme Sanatı

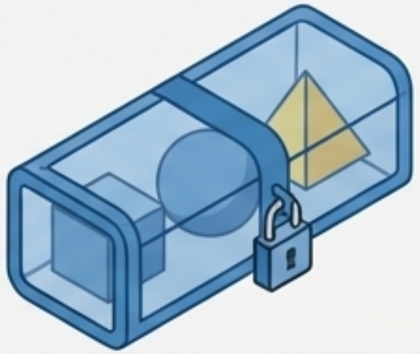
Python, birden fazla değeri bir arada depolamak için kullanılan, her biri farklı özelliklere sahip güçlü koleksiyon yapıları sunar.



Listeler (List)

Sıralı ve **değiştirilebilir**. Farklı veri türlerini içerebilir.

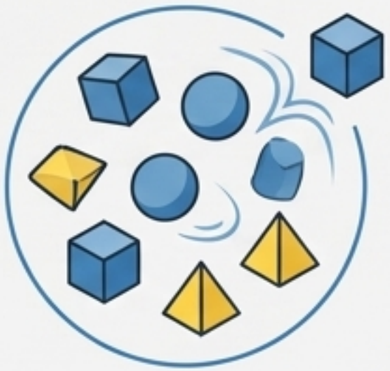
```
meyveler = [1, "elma", 3.14]
meyveler.append("kiraz")
```



Demetler (Tuple)

Sıralı ancak **değiştirilemez**. Tanımlandıktan sonra elemanları değiştirilemez.

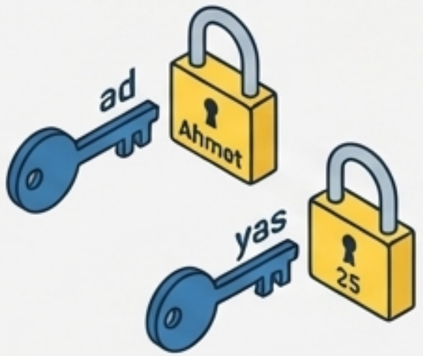
```
koordinatlar = (10, 20, "y")
```



Kümeler (Set)

Sırasız ve **tekrarsız** elemanlardan oluşur.

```
# Sonuç: {1, 2, 3, 4}
rakamlar = {1, 2, 3, 3, 4}
```



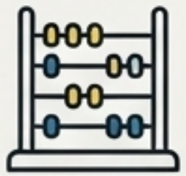
Sözlükler (Dictionary)

Anahtar-değer çiftlerinden oluşan, sırasız ve değiştirilebilir bir yapıdır.

```
kisi = {"ad": "Ahmet", "yas": 25}
```

Program Akışını Kontrol Etmek: Mantık ve Döngüler

Operatörler - Verilerle İşlem Yapma

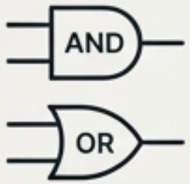


Aritmetik

`+, -, *, /, %, **, //`

Karşılaştırma

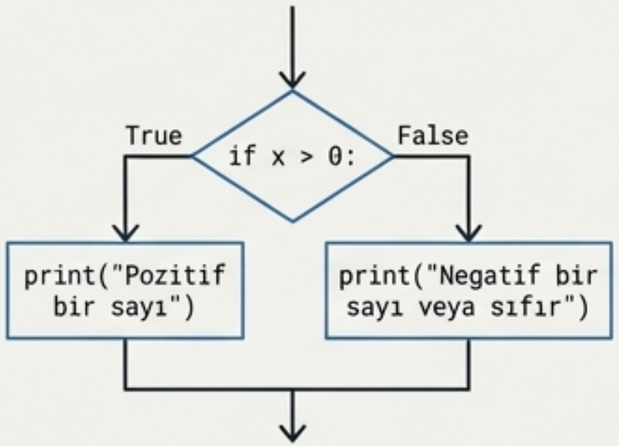
== (Eşit mi?),
!= (Eşit değil mi?), >, <



Mantıksal

and (ve), or (veya),
not (değil)

Kontrol Yapıları - Karar Verme ve Tekrarlama



Koşullu İfadeler (if-elif-else)

```
x = 10
if x > 0:
    print("Pozitif bir sayı")
else:
    print("Negatif bir sayı veya sıfır")
```

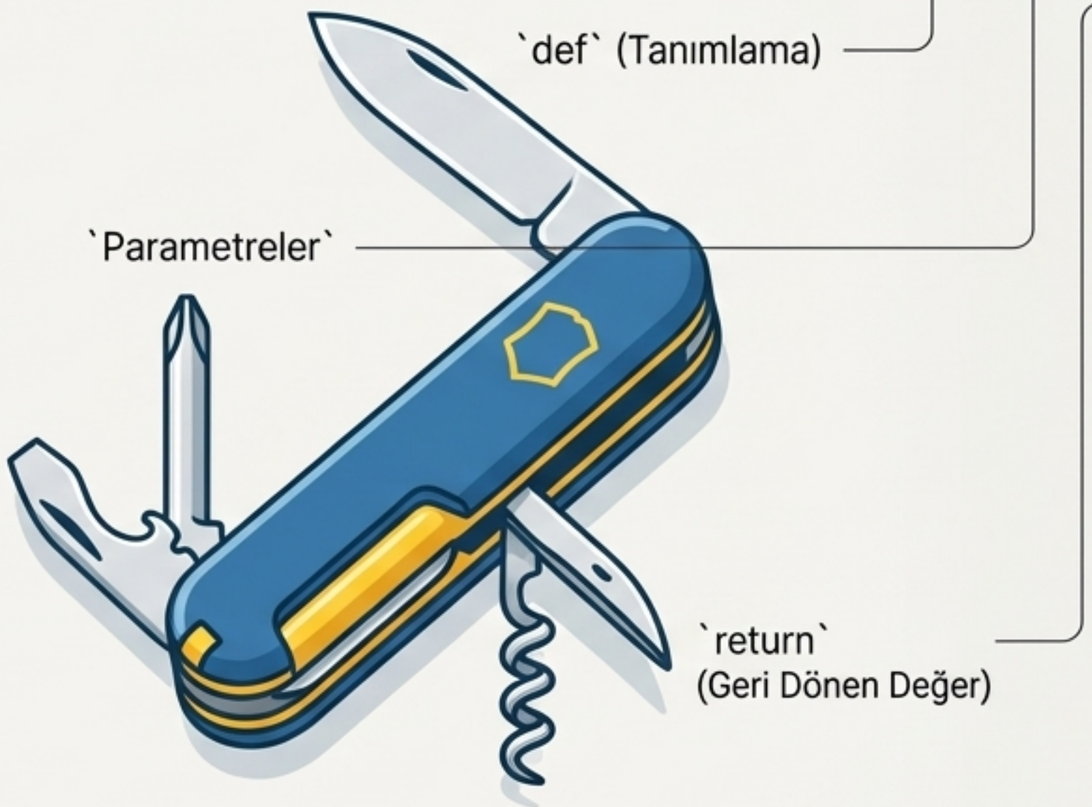
Döngüler (for, while)

```
meyveler = ["elma", "armut", "kiraz"]
for meyve in meyveler:
    print(meyve)
```

> Python'da blok yapıları girintilerle belirlenir. 4 boşluk veya **1 tab** girintisi zorunludur. ····

Kodun İsviçre Çakısı: Yeniden Kullanılabilir Fonksiyonlar

Fonksiyonlar, belirli bir görevi yerine getiren, yeniden kullanılabilir kod bloklarıdır. Programınızı daha düzenli ve yönetilebilir hale getirirler.



Fonksiyon Anatomisi

```
def selamla(isim):
```

`isim` (Parametreler: Dışarıdan veri almak için kullanılır)

`return f"Merhaba, {isim}"` (Geri Dönen Değer: İşlem sonucunu döndürür)

Örnek Kod

```
# Fonksiyon tanımı
def topla(a, b):
    return a + b

# Fonksiyon çağrılması
sonuc = topla(5, 7)
print(sonuc)  # Çıktı: 12
```

İleri Seviye Araç: Lambda Fonksiyonları

Tek satırlık, isimsiz fonksiyonlar tanımlamak için kullanılır.

```
carpma = lambda x, y: x * y
print(carpma(3, 4)) # Çıktı: 12
```

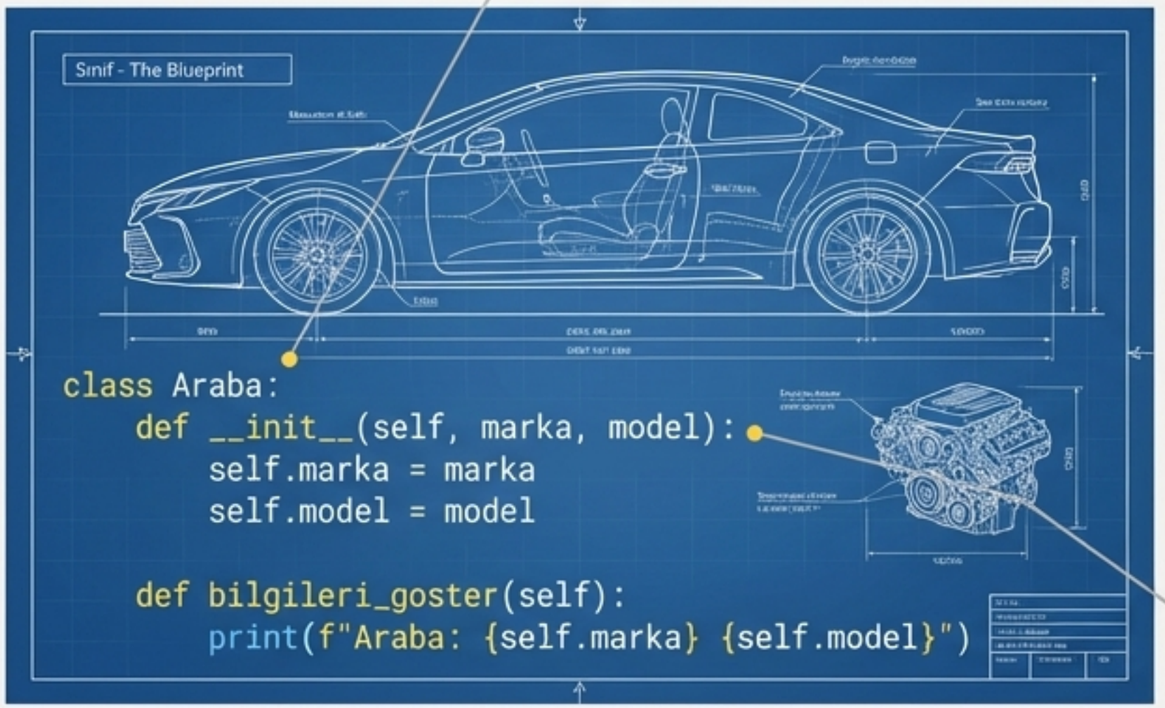

PYTHON PROGRAMLAMA - ISBN No: 978-625-00-2406-5

Nesne Yönelimli Düşünce: Kendi Veri Dünyanızı Yaratın

OOP, yazılımı birbiriyle etkileşen "nesneler" olarak modellemeye dayanan bir yaklaşımdır. Python, bu yaklaşımı `sınıflar` ve `nesneler` ile tam olarak destekler.

Sınıf - The Blueprint

Sınıf (Class): Nesnelerin özelliklerini ve davranışlarını tanımlayan bir kalıptır.



Yapıcı Metot (`__init__`): Nesne oluşturulduğunda otomatik olarak çağrılır.

Nesne - The Reality

```
# Nesne oluşturma
arac1 = Araba("Toyota", "Corolla")
arac1.bilgileri_goster() # Çıktı: Araba: Toyota Corolla
```

Nesne (Object): Bir sınıftan türetilmiş somut bir örnektir.

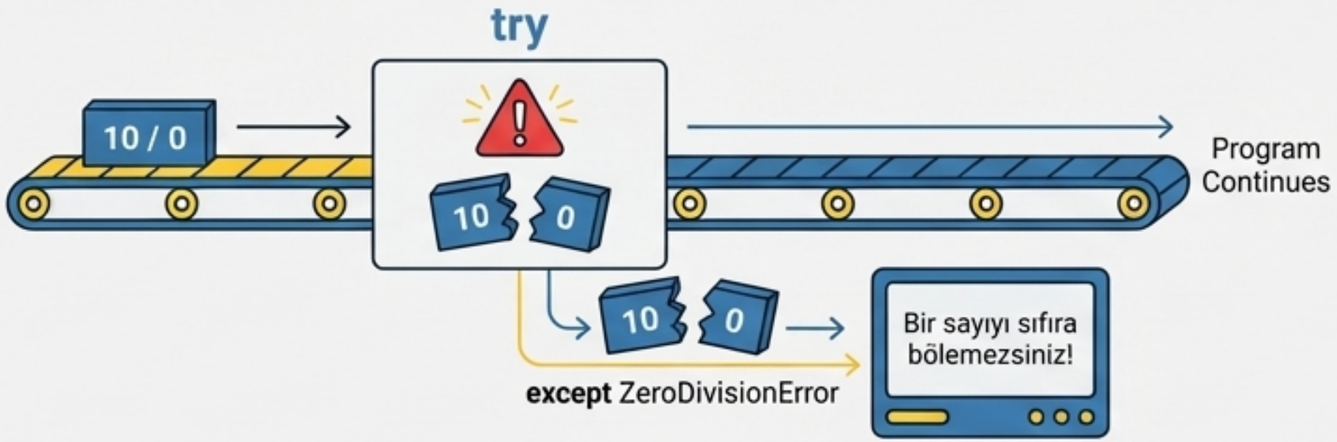
Diğer OOP Prensipleri

Miras Alma (Inheritance): Bir sınıfın özelliklerini başka bir sınıfa devretmesi.

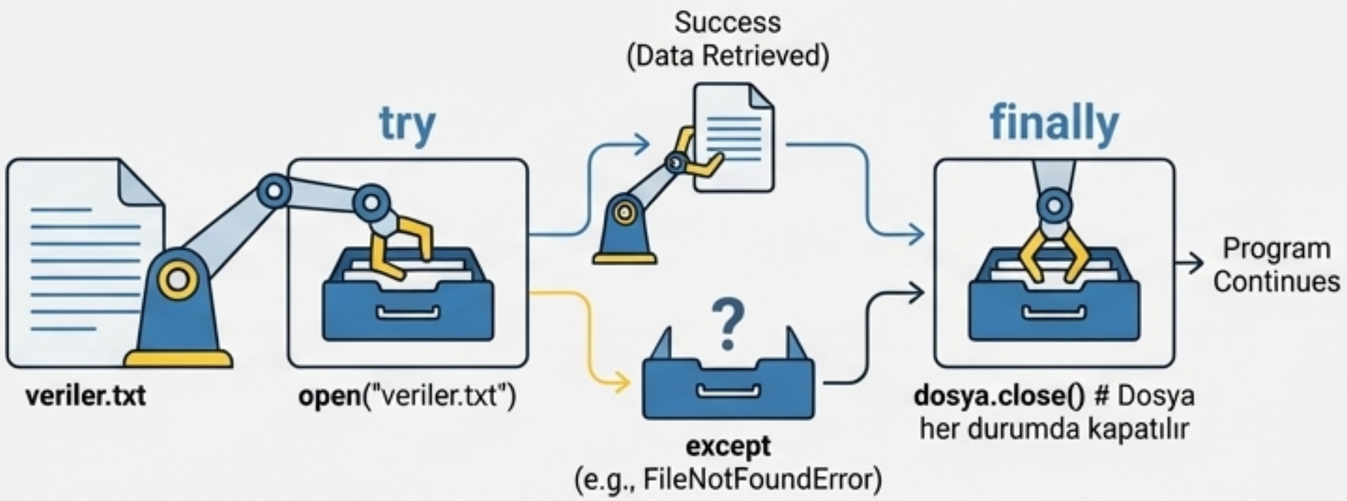
Kapsülleme (Encapsulation): Verileri dış erişime karşı koruma.

Hataları Kucaklamak: `try-except` ile Sağlam Kodlar

Programlar çalışırken beklenmedik durumlar ve hatalar oluşabilir. **`try-except`** yapısı, bu hataları yakalayarak programın çökmesini engeller ve hatayı kontrollü bir şekilde yönetmenizi sağlar.




```try:
    sonuc = 10 / 0
except ZeroDivisionError:
    print("Bir sayıyı sıfıra bölemezsiniz!")```




**\*\*Gelişmiş Kullanım: `finally`\*\***

Hata oluşsa da oluşmasa da **her zaman** çalışacak kod bloğudur. Genellikle dosya kapatma gibi temizlik işlemleri için kullanılır.

```try:
    dosya = open("veriler.txt", "r")
    # Dosya ile işlemler...
finally:
    dosya.close() # Dosya her durumda kapatılır```

# Gücünüzü Artırın: Modüller ve Kütüphaneler Ekosistemi

Python'un gücü, kodun yeniden kullanılabilirliğini sağlayan modüllerden ve belirli görevler için hazır çözümler sunan geniş kütüphane desteğinden gelir.

## Python Standart Kütüphanesi


Python ile birlikte gelen, ek kurulum gerektirmeyen hazır modüller.

```
import math
print(math.pi)       # 3.14159...
print(math.sqrt(16)) # 4.0
```

Üçüncü Parti Kütüphaneler

Topluluk tarafından geliştirilen ve `pip` ile yüklenen paketler.

****Yükleme:**

```
pip install requests
```

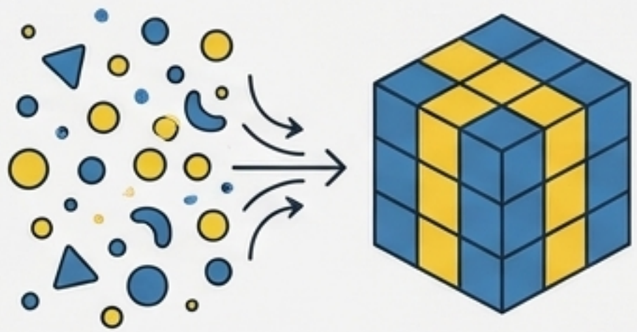
****Kullanım:**

```
import requests
response = requests.get("https://api.github.co
print(response.status_code) # Çıktı: 200
```

Verinin Gücünü Keşfedin: NumPy, Pandas & Matplotlib

Python, veri bilimi ve analizi için endüstri standardı haline gelmiştir. Bu üç kütüphane, veri işleme sürecinin temelini oluşturur.

NumPy

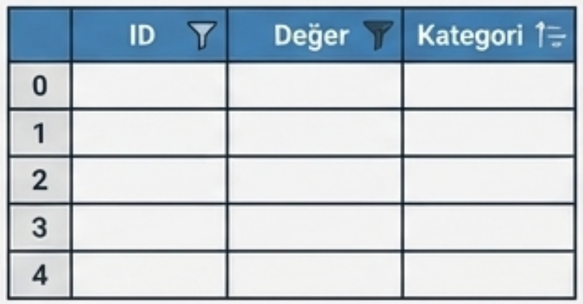


Ne Yapar?

Yüksek performanslı çok boyutlu diziler (array) ve matris işlemleri için kullanılır. Bilimsel hesaplamaların temelidir.

```
dizi = np.array([1, 2, 3, 4, 5])
```

pandas

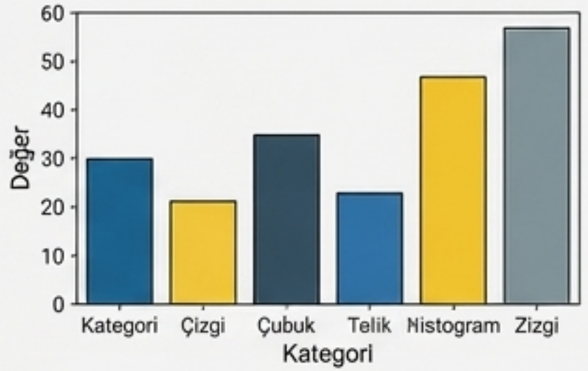


Ne Yapar?

Yapılandırılmış verileri (CSV, Excel) işlemek, analiz etmek ve manipüle etmek için kullanılır. Temel veri yapısı `DataFrame`'dir.

```
df = pd.read_csv("veriler.csv")
```

Matplotlib



Ne Yapar?

Verileri görselleştirmek için kullanılır. Çizgi, çubuk, histogram gibi çeşitli grafikler oluşturmayı sağlar.

```
plt.plot(x, y)
```

Mini Akış: Pandas ile veriyi oku, NumPy ile hesapla, Matplotlib ile görselleştir.

PYTHON PROGRAMLAMA - ISBN No: 978-625-00-2406-5

Öğrendiklerinizi Hayata Geçirin: İlham Veren Projeler

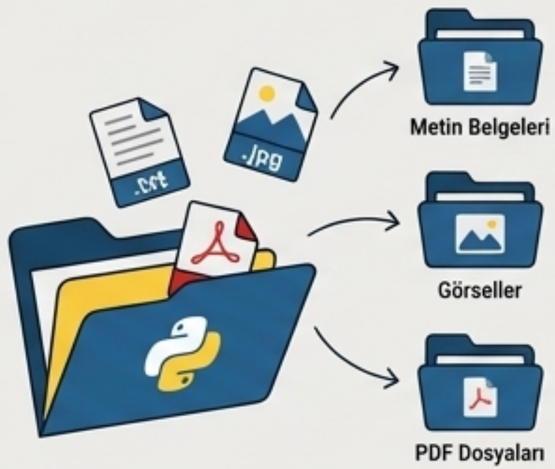
Teorik bilgiyi pekiştirmenin en iyi yolu pratik yapmaktır. İşte Python ile yapabileceğiniz birkaç başlangıç projesi.

Basit Hesap Makinesi

Kullanıcıdan iki sayı ve bir işlem alarak sonucu ekrana yazdıran temel bir konsol uygulaması.

Kullanılan Beceriler: Değişkenler, input(), if-elif-else.

```
sayi1 = float(input("Birinci sayıyı girin: "))
sayi2 = float(input("İkinci sayıyı girin: "))
if islem == '1':
    print(f"Sonuç: {sayi1 + sayi2}")
# ...
```



Dosya Yönetimi Otomasyonu

Belirli bir klasördeki dosyaları uzantılarına göre (.txt, .jpg vb.) otomatik olarak ilgili alt klasörlere taşıyan bir script.

Kullanılan Beceriler: `os` ve `shutil` modülleri, döngüler, dosya işlemleri.

Veri Görselleştirme Uygulaması

Bir CSV dosyasındaki verileri okuyup (örneğin, yaş dağılımı), Pandas ile analiz edip Matplotlib ile histogram grafiği oluşturan bir uygulama.

Kullanılan Beceriler: Pandas, Matplotlib, dosya okuma.

Ustalığa Giden Yol: Sırada Ne Var?

Python ile programlamaya giriş yolculuğunuzu tamamladınız. Ancak bu macera burada bitmiyor. Python'un geleceği parlak ve sizi bekleyen birçok ileri seviye konu var.

Gelecek Trendleri ve Alanlar

- **Veri Bilimi ve Yapay Zeka:** TensorFlow ve PyTorch gibi kütüphanelerle derin öğrenme modelleri geliştirin.
- **Web Geliştirme:** Django ve Flask gibi framework'lerle dinamik web siteleri ve API'ler oluşturun.
- **Otomasyon ve DevOps:** Sistem yönetimi ve bulut tabanlı görevleri otomatikleştiren script'ler yazın.

İleriye Yönelik Adımlar

- **İleri Düzey Python:** Asenkron programlama (asyncio), dekoratörler ve jeneratörler gibi konuları öğrenerek kodunuzu optimize edin.
- **Diğer Dillerle Entegrasyon:** Performans gerektiren yerlerde C/C++ entegrasyonu yapmayı veya Jython ile Java platformunda çalışmayı keşfedin.

Öğrenme sürekli bir yolculuktur. Projeler geliştirmeye, topluluklara katılmaya ve yeni kütüphaneleri