

EĞİTİMİN AMACI

Bu eğitimin amacı, Python programlama dilinin temel kavramlarını öğrenmek ve bu bilgilerle basit projeler geliştirebilecek düzeye gelmektir. Eğitim, katılımcılara Python'un gücünü ve esnekliğini tanıtmak, onları daha ileri seviyedeki programlama ve veri bilimi gibi konulara yönlendirmek için bir temel oluşturmayı hedefler. Ayrıca, programlama mantığını anlamak, algoritmalar geliştirmek ve problemleri çözmek için gerekli olan becerileri kazandırmak eğitimdeki ana odak noktalarından biridir.

EĞİTİM PLANI

Hafta	Konu Başlıkları
1. Hafta: Python'a Giriş	<ul style="list-style-type: none">- Python Nedir? Kullanım Alanları- Python Kurulumu ve IDE'ler- Python Çalışma Mantığı: İfadeler, Yorumlayıcı- <code>print()</code> ve <code>input()</code> Fonksiyonları- Veri Türleri (<code>int</code>, <code>float</code>, <code>string</code>, <code>bool</code>)- Yorum Satırları- Basit Matematiksel İşlemler- Casting (Tür Dönüşümü)- Koşullu İfadeler (<code>if</code> , <code>else</code> , <code>elif</code>)
2. Hafta: Döngüler ve Veri Yapıları	<ul style="list-style-type: none">- Döngüler: <code>for</code> , <code>while</code>- Döngülerde <code>break</code> , <code>continue</code> , <code>range()</code>- Listeler (<code>list</code>)- Demetler (<code>tuple</code>)- Sözlükler (<code>dictionary</code>)
3. Hafta: Fonksiyonlar ve Değişkenler	<ul style="list-style-type: none">- Fonksiyonlar ve Fonksiyon Tanımlama- <code>return</code> İfadesi- Global ve Yerel Değişkenler- Fonksiyonlarda Parametreler
4. Hafta: Dosya İşlemleri	<ul style="list-style-type: none">- Dosya Açma, Okuma, Yazma- Dosya Kapatma- Dosya Modları (<code>r</code> , <code>w</code> , <code>a</code>)- Hata Yönetimi ve Dosya İşlemleri

PYTHON NEDİR?



Python, günümüzde en popüler ve çok yönlü programlama dillerinden biridir. Hem yeni başlayanlar için anlaşılır bir sözdizimine sahip hem de profesyoneller tarafından dev projelerde kullanılıyor. Google, NASA, Netflix gibi dev firmalar Python'u veri analizi, yapay zeka ve web geliştirme gibi birçok alanda tercih ediyor. Python'un esnekliği, hızlı öğrenilebilirliği ve güçlü kütüphaneleri sayesinde yazılım dünyasına adım atmak isteyenler için harika bir başlangıç noktasıdır.

NEDEN PYTHON?

- **Dinamik olarak türlendirilmiş**
Değişken tanımlarken tür belirtmemize gerek yoktur. Python, değişkenin türünü otomatik olarak belirler ve çalışma zamanında bu türü atar.

Python, yorumlayıcı bir dildir. Bu, kodu satır satır okuyup çalıştırdığı anlamına gelir. Önce hataları kontrol eder, eğer hata yoksa kodu çalıştırır.











- C ve C++ gibi ayrı derleme ve çalıştırma adımları yoktur.
- Kaynak kodu doğrudan çalıştırabilirsiniz.
- Python, kodu ara bir form olan bytecode'a çevirir ve bu da bilgisayarın diline dönüştürülerek çalıştırılır.
- Kütüphanelerle bağlantı ve yükleme işlemleriyle uğraşmanıza gerek yoktur.

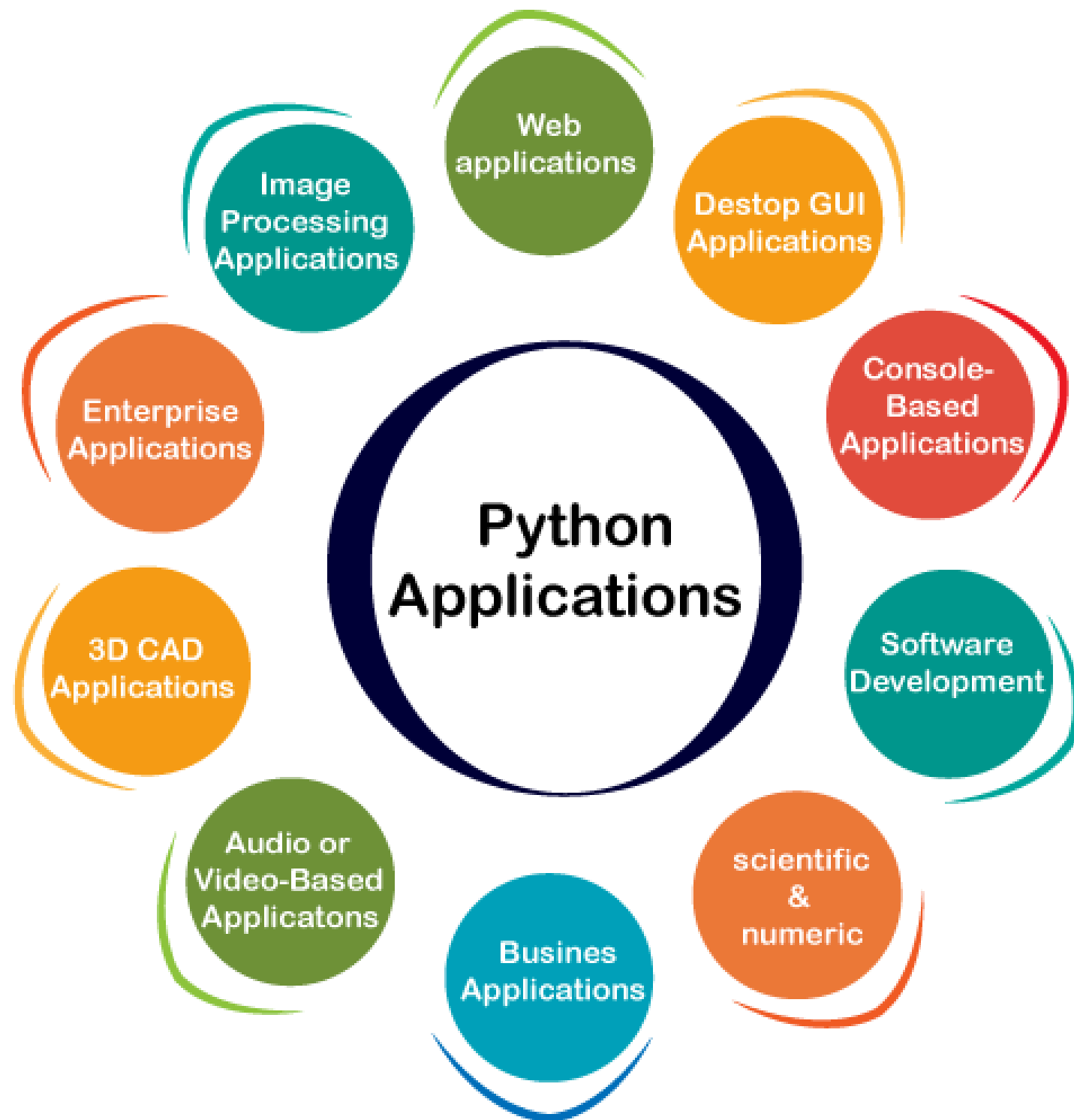
NEDEN PYTHON?

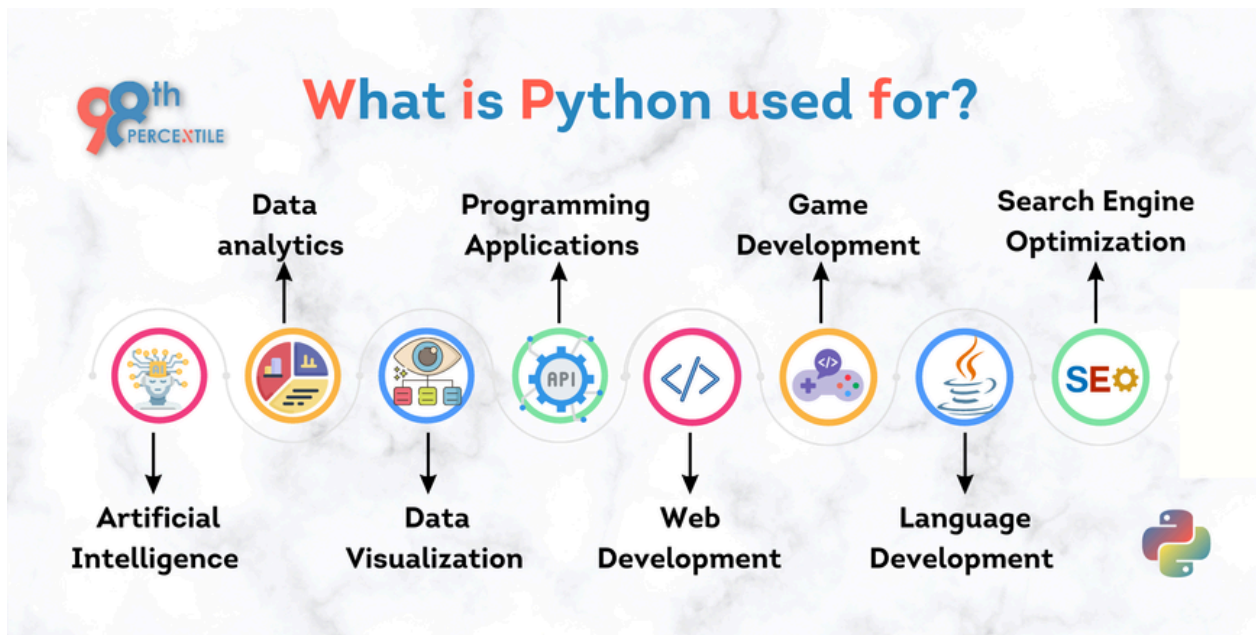
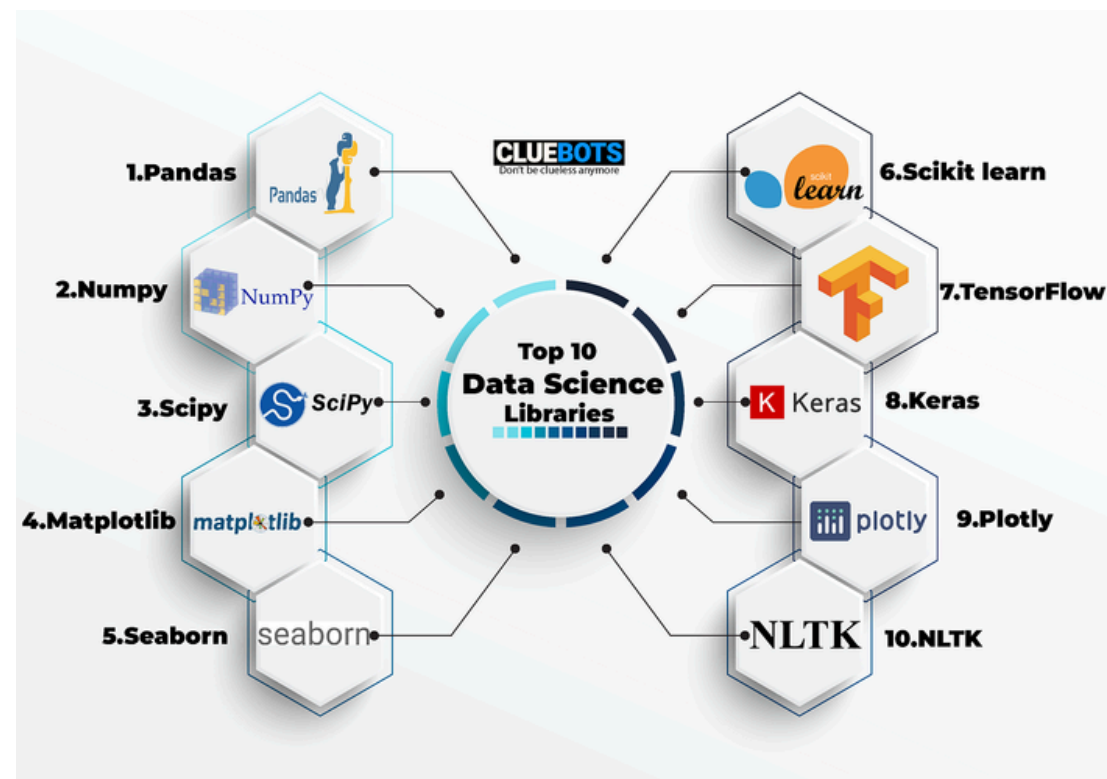
Kütüphaneler ve Frameworkler Python'un popülerliği sayesinde yüzlerce farklı kütüphane ve framework vardır. Bu araçlar, geliştirme sürecinizi hızlandırır ve bazı işleri manuel yapmanıza gerek kalmadan kolayca çözüm sağlar.

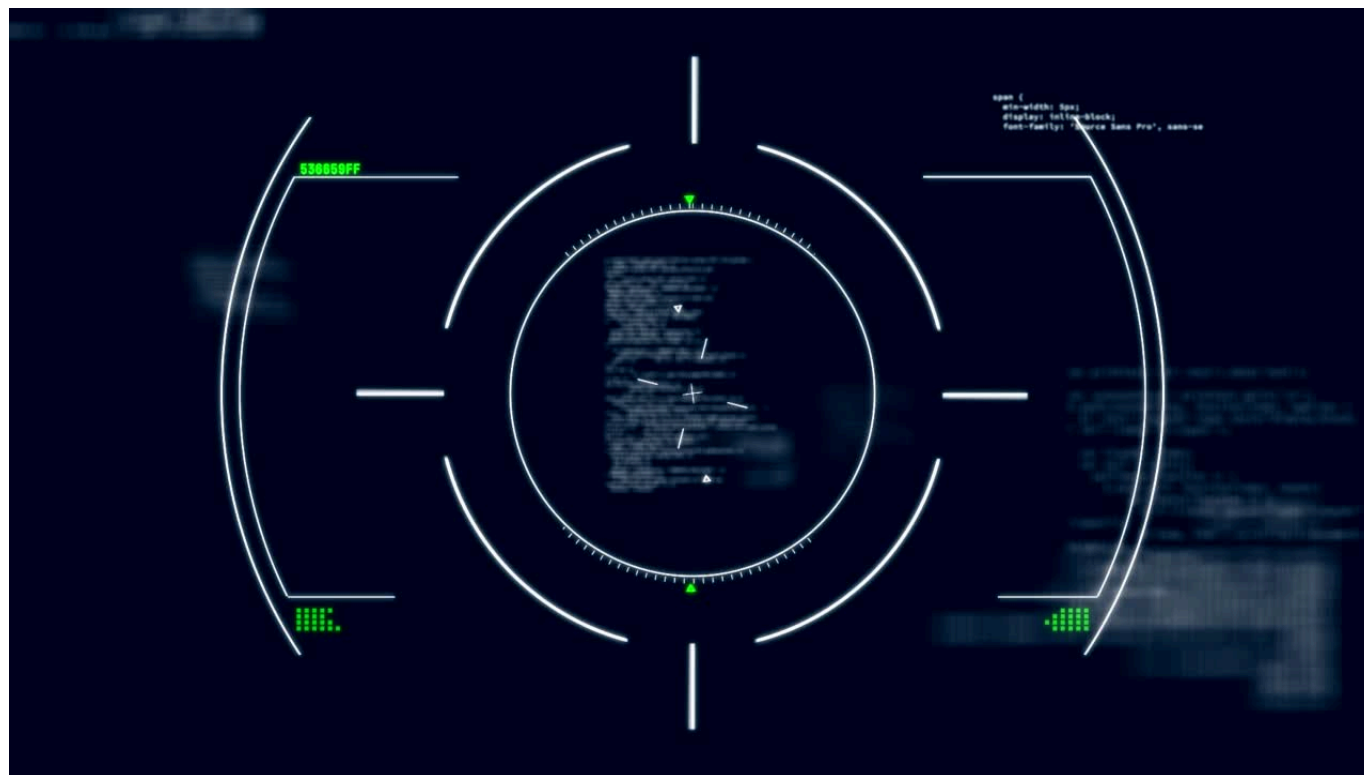
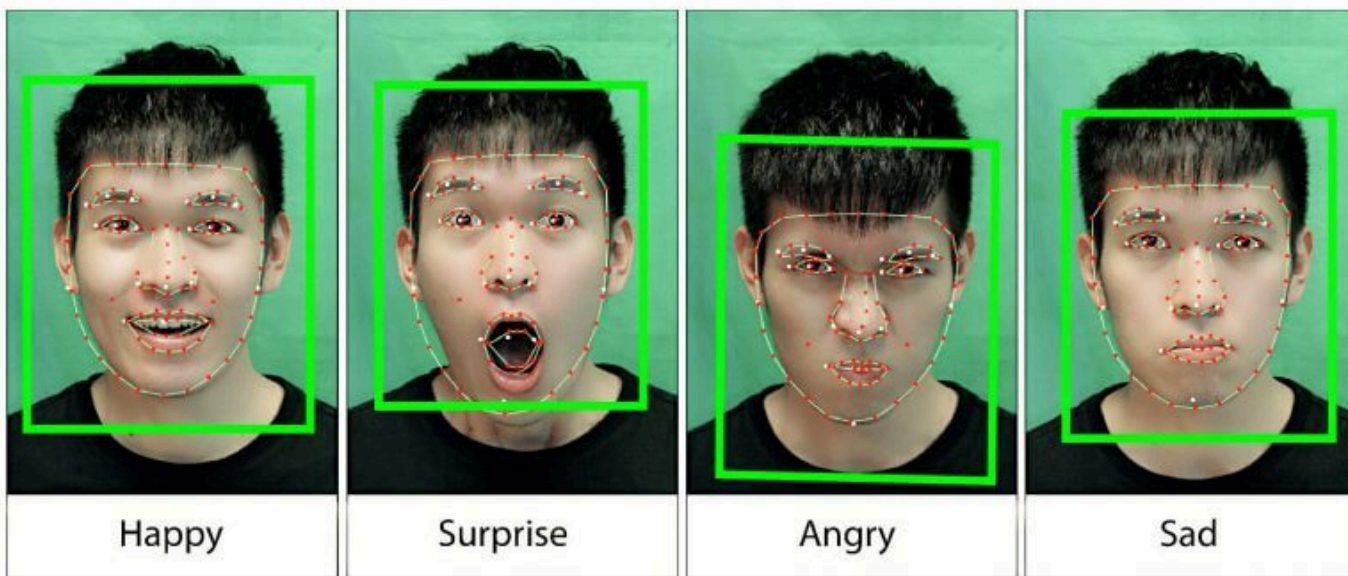
Kütüphane: Daha önce yazılmış, tekrar kullanılabilir kod parçalarından oluşan bir paket. Belirli bir işlevi hızlıca gerçekleştirmek için kullanılır (örneğin, grafik çizmek).

Framework: Belirli bir alanda (örneğin web geliştirme) bir uygulama yapmanızı sağlayan, daha geniş ve yönlendirici bir yapı sunar. Kütüphaneleri ve diğer araçları bir araya getirir.

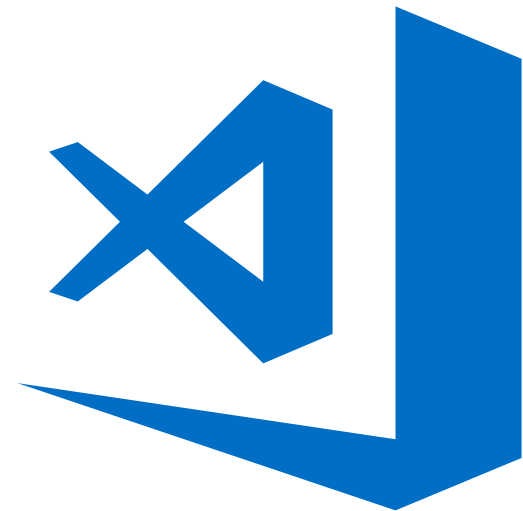
TIOBE Index						
TIOBE Index						
Oct 2024	Oct 2023	Change	Programming Language		Ratings	Change
1	1			Python	21.90%	+7.08%
2	3	^		C++	11.60%	+0.93%
3	4	^		Java	10.51%	+1.59%
4	2	v		C	8.38%	-3.70%
5	5			C#	5.62%	-2.09%
6	6			JavaScript	3.54%	+0.64%
7	7			Visual Basic	2.35%	+0.22%
8	11	^		Go	2.02%	+0.65%
9	16	^^		Fortran	1.80%	+0.78%
10	13	^		Delphi/Object Pascal	1.68%	+0.38%



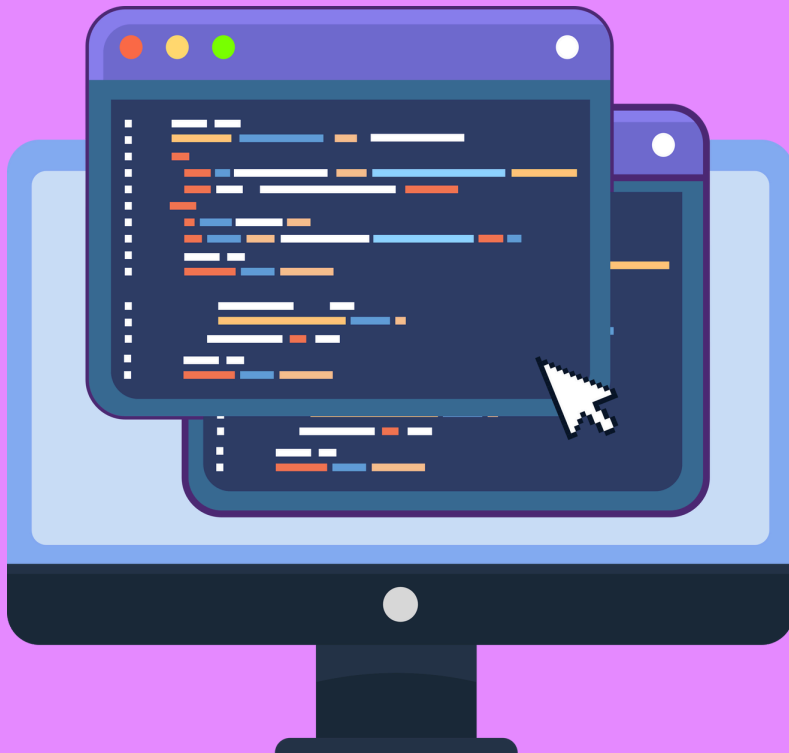




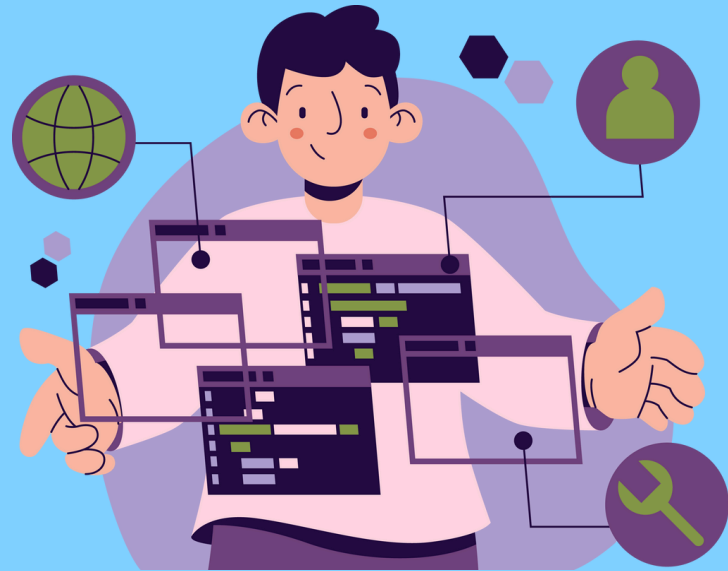
KURULUM VE IDE'LER



PROGRAMLAMA MANTIĞI



Programlama Mantığı



101010
101010
101010

DEĞİŞKENLER VE VERİ TÜRLERİ

Değişkenler, veri değerlerini depolamak için kullanılan yapılardır.

Python'da bir değişkeni tanımlamak için özel bir komut yoktur.

Bir değişken, ilk kez bir değeri atadığınız anda oluşturulur.

Değişkenlerin belirli bir tür ile tanımlanması gerekmez ve atandıktan sonra türü bile değiştirilebilir.

Girintileme, kod satırlarının başındaki boşluklara verilen isimdir.

Python'da girintileme, bir kod bloğunun başlangıcını ve bitişini belirtmek için kullanılır ve oldukça önemlidir. Girintileme doğru yapılmazsa kod hata verir.

Python'da değişken adı oluştururken aşağıdaki kurallara dikkat etmelisiniz:

- **Alfabetik karakterler ve alt çizgiyle başlamalıdır: Değişken adları bir harf (a-z, A-Z) veya alt çizgi (_) ile başlamalıdır. Sayılarla başlayamaz.**
- **Büyük-küçük harf duyarlılığı vardır: Python'da büyük harf ve küçük harf farklı kabul edilir, bu nedenle myVar ile myvar aynı değişken değildir.**


- **Sayılar, harflerden sonra gelebilir: Değişken adları harflerden veya alt çizgiden sonra sayı içerebilir.**
- **Özel karakterler kullanılamaz: Değişken adlarında sadece harfler, sayılar ve alt çizgi (_) kullanılabilir. Boşluk, tire (-), nokta (.) gibi karakterler kullanılamaz.**
- **Anahtar kelimeler kullanılamaz: Python'un rezerve ettiği anahtar kelimeler (örneğin: if, for, while, def, class) değişken adı olarak kullanılamaz.**

Veri Türleri:

Değişkenler, farklı türlerde veri depolayabilir. Python'da en yaygın veri türleri şunlardır:

1. **int** (Tam Sayı): Tam sayı değerlerini saklar. Örneğin: 5, -10.
2. **float** (Ondalık Sayı): Kesirli sayıları saklar. Örneğin: 3.14, -0.001.
3. **str** (Metin): Metin veya karakter dizilerini saklar. Örneğin: "Merhaba", "123".
4. **bool** (Boolean): Doğru (True) veya yanlış (False) değerlerini saklar.

Bir değişkenin veri türünü `type()` fonksiyonu ile öğrenebilirsiniz.

 main.py ×

```
1 isim = "IEEE"
2 yas = 20
3
4 print(type(isim))
5 print(type(yas))
6
```

Değişken isimleri büyük/küçük harf duyarlıdır.

```
isim = "IEEE"

İsim = 2024
```

PYTHON'DA YORUMLAR

```
# Birden fazla satırla açıklama yapılabilir.  
# Bu fonksiyon iki sayıyı toplar ve sonucu döner.  
def sayilari_topla(a, b):  
    return a + b # a ve b'nin toplamını döner.  
  
# Fonksiyonu kullanma örneği  
sonuc = sayilari_topla(5, 3) # Fonksiyona 5 ve 3'ü gönderiyoruz  
print("Sonuç:", sonuc) # Sonucu yazdırır  
  
"""  
Bu Python'da  
birçok satıra yayılan  
bir yorumdur.  
"""
```

Yorum satırları, geliştiricilerin kodu okuyacak kişilere kodun nasıl çalıştığını veya mantığını açıklamak için ekledikleri bilgilerdir. Yorumlar, kodun belli bir bölümünü anlamayı kolaylaştırır.

Python'da iki tür yorum vardır:

- **Tek satırlık yorumlar:** Python'da tek satırlık bir yorum **#** sembolüyle başlar ve bu sembolün ardından boşluk bırakmadan yazılır.
- **Çok satırlı yorumlar:** Python'da çok satırlı yorumlar, her iki tarafı da **"""** ile çevrelenmiş bir metin parçasıdır. Bu, birden fazla satıra yayılan bir yorum oluşturur.

Senaryo:

Basit Veri Türleri ile Süpermarket Otomasyonu

Bir süpermarkette çalışan bir yazılımcısın ve temel işlemler için küçük bir program yazman isteniyor. Süpermarket, ürünleri yönetmek, fiyatları hesaplamak ve müşteri bilgilerini düzenlemek için küçük bir Python programı kullanmak istiyor.

Görev:

- **Ürünün adı bir string olarak saklanmalı.**
- **Ürünün fiyatı bir float olarak saklanmalı.**
- **Satılan ürün sayısı bir integer olarak saklanmalı.**
- **Fonksiyon: Program, süpermarketin kasasında çeşitli işlemler yapmalı:**
- **Ürün adını ekrana yazdır.**
- **Ürünün fiyatını 10% indirimli olarak hesapla.**
- **Satılan ürün sayısını ekrana yazdır.**

Python'da input() Fonksiyonu

- **input()** fonksiyonu, kullanıcıdan girdi almak için kullanılır. Program çalışırken kullanıcıdan bir metin girmesini bekler.
- **Kullanım:** Genellikle kullanıcıdan bilgi almak için kullanılır; bu, programın dinamik olarak çalışmasını sağlar.

- **Parametre:** **input()** fonksiyonu, isteğe bağlı bir parametre alır. Bu parametre, kullanıcıya gösterilecek mesajdır.
- **Değer:** Kullanıcıdan alınan girdi, her zaman bir string (metin) olarak döner.

```
isim = input("Adınızı girin: ")  
print("Merhaba,", isim)
```

STRING'LER

String Nedir?

- String, bir karakter dizisi olarak tanımlanır ve metin verilerini temsil eder.
- Örneğin: "Merhaba Dünya"

Programlamadaki Önemi

- Stringler, kullanıcı girişi, veri saklama ve metin manipülasyonu gibi birçok alanda kullanılır.

String Yapısı

- Karakter Dizisi Tanımı
 - Bir veya daha fazla karakterin bir araya gelmesiyle oluşan dizidir.
 - Örnek: **mesaj = "Selam";**
- String Değişkenleri
 - Programda string verilerini saklamak için değişkenler kullanılır.

Python String Methods

Python has a set of built-in methods that you can use on strings.

Note: All string methods returns new values. They do not change the original string.

Method	Description
<u>capitalize()</u>	Converts the first character to upper case
<u>casefold()</u>	Converts string into lower case
<u>center()</u>	Returns a centered string
<u>count()</u>	Returns the number of times a specified value occurs in a string
<u>encode()</u>	Returns an encoded version of the string
<u>endswith()</u>	Returns true if the string ends with the specified value
<u>expandtabs()</u>	Sets the tab size of the string
<u>find()</u>	Searches the string for a specified value and returns the position of where it was found
<u>format()</u>	Formats specified values in a string
<u>format_map()</u>	Formats specified values in a string
<u>index()</u>	Searches the string for a specified value and returns the position of where it was found
<u>isalnum()</u>	Returns True if all characters in the string are alphanumeric
<u>isalpha()</u>	Returns True if all characters in the string are in the alphabet
<u>isascii()</u>	Returns True if all characters in the string are ascii characters
<u>isdecimal()</u>	Returns True if all characters in the string are decimals
<u>isdigit()</u>	Returns True if all characters in the string are digits
<u>isidentifier()</u>	Returns True if the string is an identifier

Görev :

Bir öğrencinin aşağıdaki bilgileri için gerekli değişkenleri oluşturunuz.

Öğrencinin Adı

Öğrencinin Soyadı

Öğrencinin Adı ve Soyadı

Öğrencinin Doğum Yılı

Öğrencinin Yaşı

Öğrencinin Tıttuđu Takım

Senaryo: Metin Analiz Aracı

Bu projede, kullanıcıdan bir metin alacağız ve bu metin üzerinde bazı basit string işlemleri gerçekleştireceğiz. Kullanıcıya metindeki karakter sayısı, kelime sayısı, büyük harf ve küçük harf dönüşümlerini göstereceğiz.

Görev:

- Kullanıcıdan bir metin al.
- Metindeki toplam karakter sayısını hesapla (boşluklar dahil).
- Metindeki toplam kelime sayısını hesapla.
- Metni büyük harfle ve küçük harfle yazdır.

KOŞULLU İFADELER

Python'da koşullu ifadeler, programın belirli bir koşulun doğru veya yanlış olmasına göre farklı yollar izlemesine olanak tanır. En yaygın koşullu ifade, if ifadesidir.

2. else ifadesi: if koşulu yanlışsa alternatif bir kod bloğu çalıştırır.

```
sıcaklık = 20

if sıcaklık > 25:
    print("Hava sıcak! Dışarı çıkabilirsiniz.")
else:
    print("Hava soğuk! Evde kalmalısınız.")
```

1. if ifadesi: Koşul doğruysa belirli bir kod bloğunu çalıştırır.

```
sıcaklık = 30

if sıcaklık > 25:
    print("Hava sıcak! Dışarı çıkabilirsiniz.")
```

3. elif ifadesi: Birden fazla koşulu kontrol etmemizi sağlar.

```
sıcaklık = 15

if sıcaklık > 25:
    print("Hava sıcak! Dışarı çıkabilirsiniz.")
elif sıcaklık > 15:
    print("Hava ılıman, dışarıda vakit geçirebilirsiniz.")
else:
    print("Hava soğuk! Evde kalmalısınız.")
```

Senaryo: Not Hesaplama Programı

Proje Tanımı:

- Kullanıcıdan bir sınav notu alınır.
- Alınan nota göre, aşağıdaki harf notları belirlenir:
- Harf notu ekrana yazdırılır.

HARF NOTLARI

95-100: A1

90-94: A2

85-89: A3

80-84: B1

75-79: B2

70-74: B3

65-69: C1

60-64: C2

55-59: C3

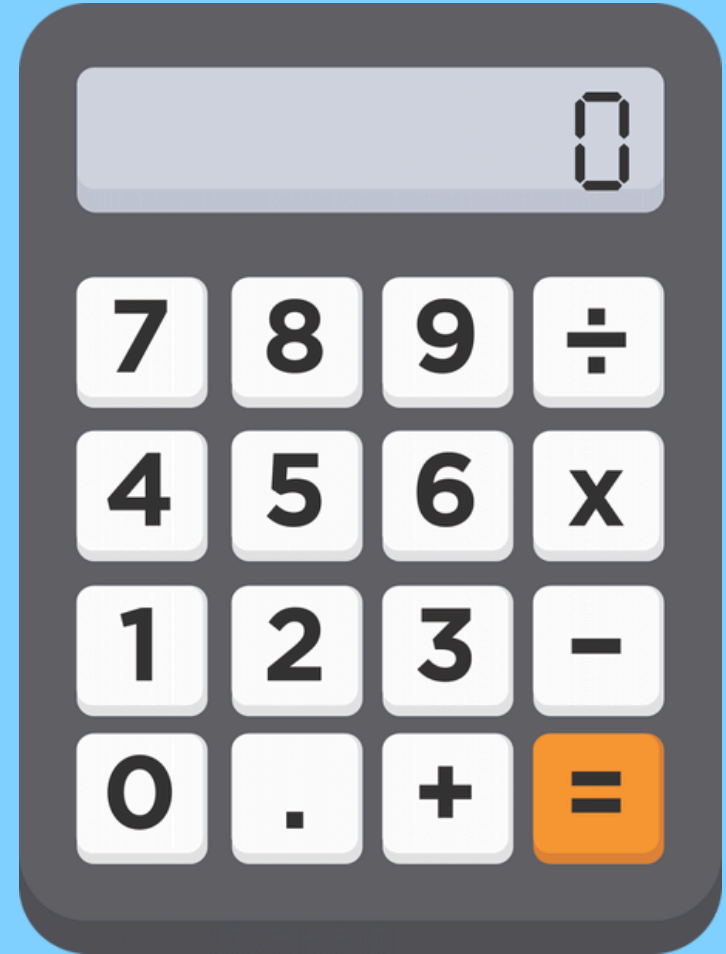
50-54: D

0-49: F

Senaryo: Hesap Makinesi

Proje Tanımı:

- Kullanıcıdan iki sayı ve bir matematiksel operatör (+, -, *, /) alarak, işlemi gerçekleştiren ve sonucu ekrana yazdıran bir hesap makinesi uygulaması oluşturularak işlem yapılacaktır.



Python'da Match-Case Yapısı

```
match deger:
    case durum1:
        # durum1 için yapılacaklar
    case durum2:
        # durum2 için yapılacaklar
    case _:
        # hiçbir durum eşleşmezse yapılacaklar
```

Belirli bir değeri kontrol etmek ve o değere göre farklı durumları ele almak için kullanılır.

```
renk = "kırmızı"

match renk:
    case "kırmızı":
        print("Renk kırmızı!")
    case "mavi":
        print("Renk mavi!")
    case "yeşil":
        print("Renk yeşil!")
    case _:
        print("Renk tanınmıyor!")
```