

TKINTER MODÜLÜ

PYTHON GÖRSEL PROGRAMLAMA İLE

12 UYGULAMA

TKINTER MODÜLÜ PYTHON GÖRSEL PROGRAMLAMA İLE 12 UYGULAMA

İÇERİK

PYTHON GÖRSEL PROGRAMLAMAYA GİRİŞ.....	1
Python Programlama Dili	1
Programlama Dilleri ile Yapılabilecek Uygulama Alanları	1
Python Dilinin Özellikleri	1
Python Dilinde Kullanılan Bazı Modüller.....	1
Python Tkinter Form Araçları	3
BÖLÜM 1 HESAP MAKİNESİ UYGULAMASI.....	5
Proje Hakkında	5
Projede Kullanılan Araçlar	5
Tkinter ile Proje Tasarımı	5
Proje Kodları	6
BÖLÜM 2 FORM ÜZERİNDE İS VE DATE BİLGİSİ UYGULAMASI.....	8
Proje Hakkında	8
Projede Kullanılan Araçlar	8
Tkinter ile Proje Tasarımı	8
Proje Kodları	9
BÖLÜM 3 FORMDA LISTBOX İLE VERİ EKLE/SİL UYGULAMASI	10
Proje Hakkında	10
Projede Kullanılan Araçlar	10
Tkinter ile Proje Tasarımı	10
Proje Kodları	11
BÖLÜM 4 FORM ÜZERİNE RESİM EKLEME UYGULAMASI	12
Proje Hakkında	12
Projede Kullanılan Araçlar	12
Tkinter ile Proje Tasarımı	12

Proje Kodları	13
BÖLÜM 5 ŞİFRENİN GÜVENLİĞİNİ TEST EDEN UYGULAMA	14
Proje Hakkında	14
Projede Kullanılan Araçlar	14
Tkinter ile Proje Tasarımı	14
Proje Kodları	15
BÖLÜM 6 FORM ÜZERİNDE LISTBOX UYGULAMASI	16
Proje Hakkında	16
Projede Kullanılan Araçlar	16
Tkinter ile Proje Tasarımı	16
Proje Kodları	17
BÖLÜM 7 FORM ÇALIŞTIĞINDA ŞİFRE OLUŞTURAN UYGULAMA...	18
Projede Kullanılan Araçlar	18
Tkinter ile Proje Tasarımı	18
Proje Kodları	19
BÖLÜM 8 FORMDA BUTON İLE ŞİFRE OLUŞTURMA UYGULAMASI.	20
Proje Hakkında	20
Projede Kullanılan Araçlar	20
Tkinter ile Proje Tasarımı	20
Proje Kodları	21
BÖLÜM 9 GİRİLEN KARAKTERLERİ SAYAN UYGULAMA.....	22
Proje Hakkında	22
Projede Kullanılan Araçlar	22
Tkinter ile Proje Tasarımı	22
Proje Kodları	23
BÖLÜM 10 NOT HESAPLAMA UYGULAMASI.....	24
Proje Hakkında	24

Projede Kullanılan Araçlar	24
Tkinter ile Proje Tasarımı	24
Proje Kodları	25
BÖLÜM 11 FORMDA ZAMAN FONKSİYONLARI UYGULAMASI	27
Proje Hakkında	27
Projede Kullanılan Araçlar	27
Tkinter ile Proje Tasarımı	27
Proje Kodları	28
BÖLÜM 12 GÖRSEL UYGULAMALARI EXE YAPMA.....	30
EXE Hakkında.....	30
Exe Adımları.....	30
EXE Yapmanın Avantajı.....	30
EXE' ye Dönüştürme İşlemleri	30

Kitap Hakkında

Hızla gelişen yazılım teknolojileri arasında son yıllarda PYTHON programlama dili ilk sıralarda yer almaktadır. Python programlama dili açık kaynak yapıya sahip olması dolayısıyla birçok dile göre daha avantajlı hale getirmiştir. Bu çalışma python dilinde özellikle masaüstü yazılım geliştirmek isteyenler için iyi bir başlangıç olacağı düşünülmektedir. Bu çalışmanın diğer bir amacı yazılan kodları uygulama haline yani kullanılabilir hale getirmektir. Görsel uygulamaların artması ve proje ortamına, EXE, SETUP haline çevrilmesini sağlamaktır. Python programlama dili, masaüstü yazılım, web teknolojileri, siber güvenlik, mobil uygulama, gömülü sistemler, yapay zekâ uygulamaları, robotik sistemler, veri analizi uygulamalarında yaygın olarak kullanılmaktadır. Günümüz teknolojilerinde birçok yazılımcı python programlama ile uygulama geliştirmektedir. Eğitim sektöründe yer alan yazılımcılarında hızla python öğrendiği, uygulama geliştirdiği bilinmekte olup, kamu, vakıf, akademi veya özel eğitim kurumlarının python programlama dilinin öğrenmesi için birçok eğitim başlattığı da bilinmektedir. Bu büyük eğitim bilgi havuzunda bizimde küçük bir katkımız olduysa ne mutlu bize.

Yazar Hakkında

Bilgisayar serüvenim 2000 Yılında Afyon Kocatepe Üniversitesi Afyon MYO Bilgisayar Donanımı ile başlamıştır. Öğrenciliğim halen devam etmektedir. 😊 Birçok kurumda bilişim teknolojileri alanında eğitimler verdim. Bilgisayar kullanımı, ofis programları, web tasarımı ve web programlama eğitimleri. Son yıllarda robotik ve yazılım alanlarında yoğunlaştım. Bu dokümanı, derslerde kullanmak ve elimizdeki dokümanları sizlerle paylaşmak için küçük bir kitapçık haline getirmeye çalıştım. Umarım çorbada bizimde bir katkımız olur düşüncesiyle oluşturduğumuz küçük notlarımızı paylaşmaktan mutluluk duyarım. Umarım çok hatamız çıkmaz. Ayrıca eğitimlerime katılan tüm kursiyerlerime teşekkür ederim. Yeni yazılım notlarında görüşmek dileğiyle.

Kod yazmaya Devam...

İletişim Bilgileri

Mail: serkanozargin@gmail.com
Web: www.serkanozargin.com
Linkedin: <https://www.linkedin.com/in/serkan-ozargin-7a153716a/>
Youtube: <https://www.youtube.com/serkanozargin>
GitHub: <https://github.com/serkan2047>
SlidePlayer: <http://slideplayer.biz.tr/user/slides/>

Teşekkür

Ders esnasında sınıf ortamında çalışmalar yaptığımız hem öğrendiğimiz hem öğrettiğimiz, tüm öğrenci ve kursiyerlerime sonsuz teşekkürlerimi sunarım.

10.06.2024
Serkan Özargın

PYTHON GÖRSEL PROGRAMLAMAYA GİRİŞ

Python Programlama Dili

Son yıllarda yükselişe geçen Python Programlama Dili açık kaynak ve birçok platformda kullanılması, python dilini popüler hale getirmiştir. Python, masaüstü, web, mobil, siber güvenlik, gömülü sistemler, veri madenciliği, veri analizi gibi birçok alanda kullanılmaktadır. Python programlama dili yapay zekâ alanında da yoğun bir şekilde kullanıldığı görülmektedir. Günümüz teknolojilerinde yaygın olarak kullanılması birçok programlama dilini geride bırakmıştır.

Programlama Dilleri ile Yapılabilecek Uygulama Alanları

Uygulama Platformları	C	C++	C#	JAVA	PYTHON
Masaüstü	+	+	+	+	+
Web	-	-	+	+	+
Mobil	-	-	+	+	+
Gömülü sistemler	+	+	-	-	+
Siber Güvenlik	-	-	-	-	+
Veri Analizi	-	-	-	-	+
Veri Madenciliği	-	-	-	-	-

Python Dilinin Özellikleri

Python programlama dili; veri bilimi, veri madenciliği, makine öğrenimi, yapay zekâ, web API geliştirme, web kazıma uygulamaları, oyun geliştirme, gömülü sistemler, siber güvenlik araçlarının yazılması gibi birçok alanda çok yaygın olarak kullanılan bir programlama dilidir. Bu nedenle en popüler programlama dilleri arasında yer almaktadır.

Python Dilinde Kullanılan Bazı Modüller

Python Programlama Dili içinde kullanılan modüller, çeşitli görevleri yerine getirmek için kullanılmaktadır. Python dilinde kullanılan bu modüller aslında birçok fonksiyonu içinde barındırmaktadır. Bu fonksiyonlar ise kod satırlarından oluşmaktadır. Kullanılan modüllerde birden fazla kod satırı yer almaktadır. Modüller python dili içinde en önemli

özelliklerinden birisidir. Python dilinde çok sayıda modül yer almaktadır. Eklenmek istenen modül import ederek kullanılmaktadır. Modüller içinde çalışan fonksiyonlar belirli görevleri yerine getirir. Python içinde modülleri kullanmak için, import edilmesi gerekmektedir. İmport etmek modülü içe aktarmak anlamına da gelmektedir. Modüller iki başlık altında incelenmektedir. Bunlar hazır modüller ve sonradan yazılan modüllerdir. Hazır modüller de kendi içinde ikiye ayrılmaktadır. Bunlar Standart modüller ve üçüncü parti modülleridir. Bu dokümanda hazır modüller daha çok kullanılmıştır. Bu çalışmada kullanılan bazı modül ve sınıfları aşağıda inceleyelim.

Date Modülü

Zaman, tarih ve saat bilgilerini kullanmamızı sağlayan modüldür. Datetime modülünde 3 farklı sınıf içinde yer almaktadır. Bu sınıflar date, datetime, time olarak kullanılmaktadır.

- **Date sınıfı:** Tarih işlemlerini yapmamızı sağlayan fonksiyonlardan oluşan sınıftır.
- **Datetime sınıfı:** Saat, dakika, saniye ve zaman dilimi gibi zaman bilgisi verir.
- **Time Sınıfı:** Zaman işlemlerini yapmamızı sağlayan fonksiyonlardan oluşan sınıftır.

```
>>> dir(datetime)

['__add__', '__class__', '__delattr__', '__dir__',
 '__doc__', '__eq__',
 '__format__', '__ge__', '__getattribute__', '__gt__',
 '__hash__',
 '__init__', '__le__', '__lt__', '__ne__', '__new__',
 '__radd__',
 '__reduce__', '__reduce_ex__', '__repr__', '__rsub__',
 '__setattr__',
 '__sizeof__', '__str__', '__sub__', '__subclasshook__',
 'astimezone',
 'combine', 'ctime', 'date', 'day', 'dst', 'fromordinal',
 'fromtimestamp',
 'hour', 'isocalendar', 'isoformat', 'isoweekday', 'max',
 'microsecond',
 'min', 'minute', 'month', 'now', 'replace', 'resolution',
 'second',
 'strftime', 'strptime', 'time', 'timestamp', 'timetuple',
 'timetz', 'today',
 'toordinal', 'tzinfo', 'tzname', 'utcfromtimestamp',
 'utcnow', 'utcoffset',
 'utctimetuple', 'weekday', 'year']
```

%a: hafta gününün kısaltılmış adı
%A: hafta gününün tam adı
%b: ayın kısaltılmış adı
%B: ayın tam adı
%c: tam tarih, saat ve zaman bilgisi
%d: sayı değerli bir karakter dizisi olarak gün
%j: belli bir tarihin, yılın kaçınıcı gününe denk geldiğini gösteren 1-366 arası bir sayı
%m: sayı değerli bir karakter dizisi olarak ay
%U: belli bir tarihin yılın kaçınıcı haftasına geldiğini gösteren 0-53 arası bir sayı
%y: yılın son iki rakamı
%Y: yılın dört haneli tam hali
%x: tam tarih bilgisi
%X: tam saat bilgisi

OS Modülü

Sistemin işletim sistemi bilgisini verir. Windows tabanlı işletim sistemi NT, linux işletim sisteminde posix değeri döner. İşletim sistemi modülü içinde birçok fonksiyon yer almaktadır. Windows ortamında \ ve \\ kullanılırken linux ortamında / işareti dönmektedir. OS modülü Python'da hazır olarak gelen bir modüldür. İşletim sistemi dosya ve izin işlemlerinde kullanılmaktadır.

Random Modülü

0.0 ile 1.0 arasında rasgele rasyonel sayı üretmek için kullanılır. Random altında birçok fonksiyon kullanılmaktadır. Bu fonksiyonlar içinde randint fonksiyonu bir tam sayı üretmek için kullanılmaktadır.

Locale Modülü

Türkçe karakterleri eklemek için kullanıldığı gibi, para birimi finansal ifadeleri biçimlendirmek için kullanılmaktadır.

Form Üzerinde Kullanılan Bazı Araçları

Label Aracı: Label aracı etiket anlamına gelmektedir. Kullanıcıya bilgi vermek için kullanılmaktadır.

Entry Aracı: Kullanıcıdan bilgi almak için kullanılan bir araçtır. Input veya textbox olarak ta bilinmektedir.

Button Aracı: Button aracı, belirli bir görevi fonksiyonu veya kod satırını çalıştırmak için kullanılmaktadır. Fare ile tıklandığında çalışmasını istediğiniz kod bloklarını harekete geçirir.

Text Aracı: Klavyeden birden fazla satır veri girişi sağlayan araçtır.

ListBox Aracı: Form üzerinde çoklu satır verilerini ekrana yazdırmak için kullanılmaktadır. Kullanıcının müdahalesi söz konusu değildir. Text aracında kullanıcı veri ekleyebilir. ListBox aracında kullanıcı klavyeden veri ekleyemez.

PhotoImage Sınıfı

Resim eklemek için kullanılan bir modüldür. JPG, PNG, GIF formatındaki resimleri okuyabilmektedir. Pencere ortamında görselleri kullanmak için bir değişken oluşturarak görselleri kullanabilirsiniz.

BÖLÜM 1 HESAP MAKİNESİ UYGULAMASI

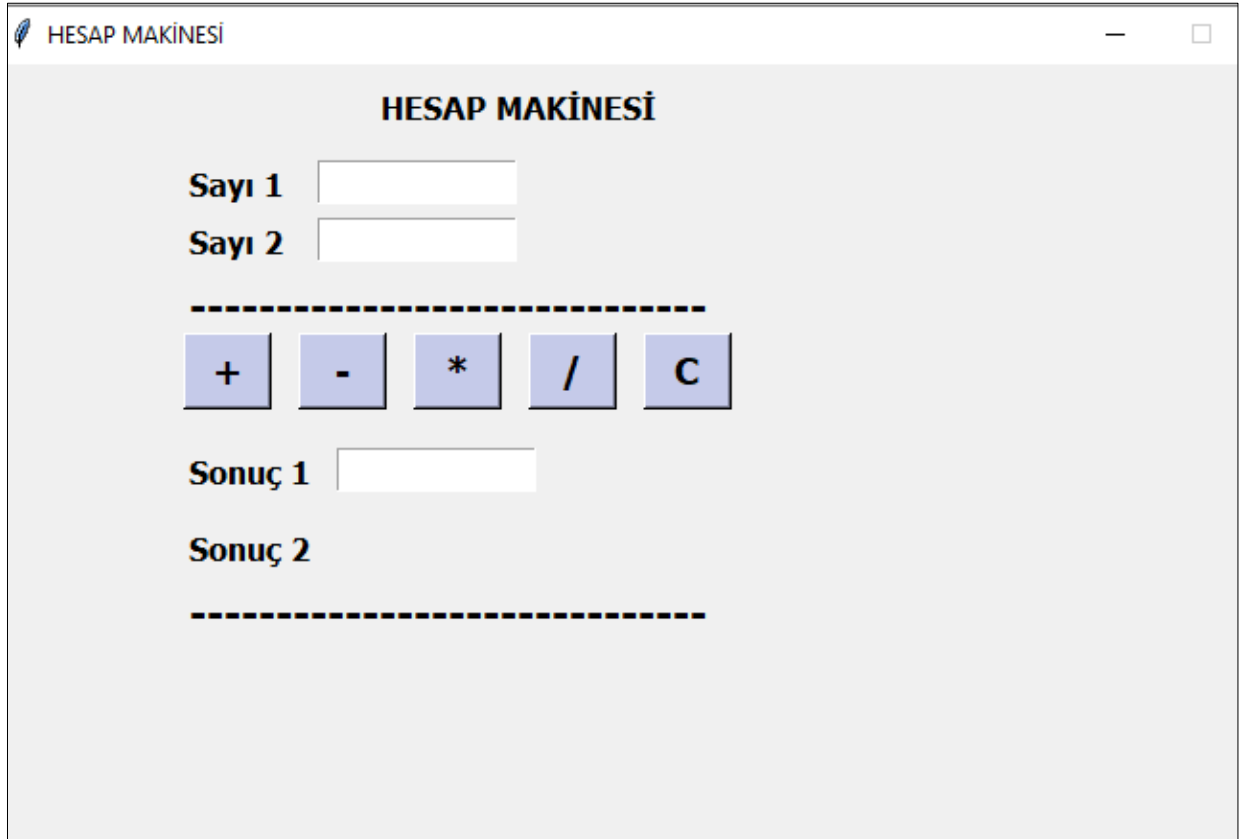
Proje Hakkında

Bu bölümde yer alan uygulamamızda basit bir hesap makinesi yer almaktadır. Bu hesap makinesi uygulamasında en az dört matematiksel işlem yer almaktadır. Görsel olarak tasarladığımız hesap makinesi uygulamasında, matematiksel işlemlerin yapılması için fonksiyonlar yazılmıştır. Sonraki süreçte bu fonksiyonlar butonlara görev olarak atanmıştır. Entry araçlarına yazılan sayılar butonların tıklanmasıyla sonuçları yer alan diğer entry ve label aracına yazdırmaktadır. Hesap makinesi uygulamasında hesaplanan sonuç Entry ve label araçlarına aynı anda yazılmaktadır. Burada önemli olan sonucun Entry aracına veya Label aracına yazdırmaktır. Şimdi aşağıda yer alan hesap makinesi görselini tasarlayalım. Daha sonraki süreçte kodları yazıp çalıştırmayı deneyelim.

Projede Kullanılan Araçlar

- Entry Aracı
- Label Aracı
- Button Aracı
- Tkinter modülü

Tkinter ile Proje Tasarımı



Proje Kodları

Hesap makinesi uygulamasında yaklaşık 64 satır kod yer almaktadır. Tkinter modülü kullanılarak, beraberinde Label ve Entry aracı kullanılmıştır.

```
1  import tkinter as tk
2  #-----
3  pencere=tk.Tk()
4  pencere.title("HESAP MAKİNESİ ")
5  pencere.geometry("400x250")
6  pencere.resizable(width="FALSE", height="FALSE")
7  #-----
8  screen_width = pencere.winfo_screenwidth()
9  screen_height = pencere.winfo_screenheight()
10 pencere.geometry(f"700x500+{(screen_width//2)-300}+{(screen_height//2)-300}")
11 #-----
12 def toplama():
13     say1 = int(s1.get())
14     say2 = int(s2.get())
15     top=say1+say2
16     s3.insert(tk.END, str(top))
17     m6["text"] = "" + str(top)
18     m6.place(x=180, y=236)
19 def temizle():
20     m6["text"] = "" + str("0")
21     #yazi=0
22     #b2.config(text=yazi)
23     #b2.config(text="0")
24     #b2=tk.Label(text=str(son))
25     #b2.place(x=100, y=250)
26     s1.delete(0, tk.END)
27     s2.delete(0, tk.END)
28     s3.delete(0, tk.END)
29 #-----
30 b1=tk.Label(text='HESAP MAKİNESİ',font = ("tahoma", 12, "bold"))
31 b1.place(x=200, y=10)
```

```

33 m1=tk.Label(text='Sayı 1',font = ("tahoma", 12, "bold"))
34 m1.place(x=100, y=50)
35 s1 = tk.Entry(font = ("tahoma", 12, "bold"), width=10)
36 s1.place(x=170, y=50)
37 m2=tk.Label(text='Sayı 2',font = ("tahoma", 12, "bold"))
38 m2.place(x=100, y=80)
39 s2 = tk.Entry(font = ("tahoma", 12, "bold"), width=10)
40 s2.place(x=170, y=80)
41 b1=tk.Label(text= '-----' ,font = ("tahoma", 16, "bold"))
42 b1.place(x=100, y=110)
43 m3=tk.Label(text='Sonuç 1',font = ("tahoma", 12, "bold"))
44 m3.place(x=100, y=200)
45 s3 = tk.Entry(font = ("tahoma", 12, "bold"), width=10)
46 s3.place(x=180, y=200)
47 m5=tk.Label(text='Sonuç 2',font = ("tahoma", 12, "bold"))
48 m5.place(x=100, y=240)
49 m6=tk.Label(text=" ",font = ("tahoma", 14, "bold"))
50 m6.place(x=300, y=260)
51 dugme1 = tk.Button(text="+", command=topla,font = ("tahoma", 14, "bold"), width=3, bg="#C5CAE9")
52 dugme1.place(x=100, y=140)
53 dugme1 = tk.Button(text="-", command=topla,font = ("tahoma", 14, "bold"), width=3, bg="#C5CAE9")
54 dugme1.place(x=160, y=140)
55 dugme1 = tk.Button(text="*", command=topla,font = ("tahoma", 14, "bold"), width=3, bg="#C5CAE9")
56 dugme1.place(x=220, y=140)
57 dugme1 = tk.Button(text="/", command=topla,font = ("tahoma", 14, "bold"), width=3, bg="#C5CAE9")
58 dugme1.place(x=280, y=140)
59 temizle = tk.Button(text="C", command=temizle,font = ("tahoma", 14, "bold"), width=3, bg="#C5CAE9")
60 temizle.place(x=340, y=140)
61 b1=tk.Label(text= '-----' ,font = ("tahoma", 16, "bold"))
62 b1.place(x=100, y=270)
63 #-----
64 pencere.mainloop()

```


BÖLÜM 2 FORM ÜZERİNDE IS VE DATE BİLGİSİ UYGULAMASI

Proje Hakkında

Python programlama dili kullanarak oluşturduğumuz proje uygulamasında görsel modül olarak Tkinter kullanılmıştır. Tasarlanan form ekranı üzerinde işletim sistemi bilgisi, tarih saat gibi bilgiler yer almaktadır. Amaç form üzerine, zaman ve işletim sistemi bilgilerinin çekilmesidir. Form ekranına farklı bilgilerde çekilebilir. Bu çekilen bilgiler Label araçları kullanılarak yazdırılmıştır. Form ekranı üzerinde işletim sistemi bilgilerini çekmek için OS modülü, tarih bilgilerinin çekilmesi için DATE modülü eklenmiştir. Bu bağlamda görsel form ekranı kullanımı için Tkinter modülü nünde import edilmesi gerekmektedir.

Projede Kullanılan Modül ve Araçlar

- Label aracı
- TKINTER modülü
- OS modülü
- DATETIME modülü
- LOCALE modülü

Tkinter ile Proje Tasarımı



Proje Kodları

📄 tarih is yazdırmak.py ✕

📄 tarih is yazdırmak.py > ...

```
1  import tkinter as tk
2  import datetime
3  import os
4  import locale
5  locale.setlocale(locale.LC_ALL, '')
6  #-----
7  pencere=tk.Tk()
8  pencere.title("PYTHON GÖRSELLEŞTİRME")
9  pencere.geometry("500x400")
10 pencere.configure(background="#BBDEFB")
11 #-----
12 lbl1=tk.Label(text="PYTHON ARAYÜZ GÖRSELLEŞTİRME",fg="#0D47A1",bg="yellow",font="Times 14 bold")
13 lbl1.place(x=50,y=10)
14 #-----
15 gun=datetime.datetime.now()
16 lbl2=tk.Label(text=gun.year,font="Times 12 bold")
17 lbl2.place(x=140,y=50)
18 gun=datetime.datetime.now()
19 lbl3=tk.Label(text=gun.month,font="Times 12 bold")
20 lbl3.place(x=40,y=50)
21 gun=datetime.datetime.now()
22 lbl4=tk.Label(text=gun.day,font="Times 12 bold")
23 lbl4.place(x=10,y=50)
24 an = datetime.datetime.now()
25 tarih = datetime.datetime.strftime(an, '%B')
26 #gun=datetime.datetime.now()
27 lbl4=tk.Label(text=tarih,font="Times 12 bold")
28 lbl4.place(x=70,y=50)
29 #-----
30 lbl5=tk.Label(text=os.name.upper(),font="Times 12 bold")
31 lbl5.place(x=150,y=80)
32 #https://python-istihza.yazbel.com/standart\_moduller/datetime.html
33 #locale.setlocale(locale.LC_ALL, '')
34 #'Turkish_Turkey.1254' tarih bilgisi TR oldu.
35 pencere.mainloop()
```

BÖLÜM 3 FORMDA LISTBOX ARAÇINA VERİ EKLE/SİL UYGULAMASI

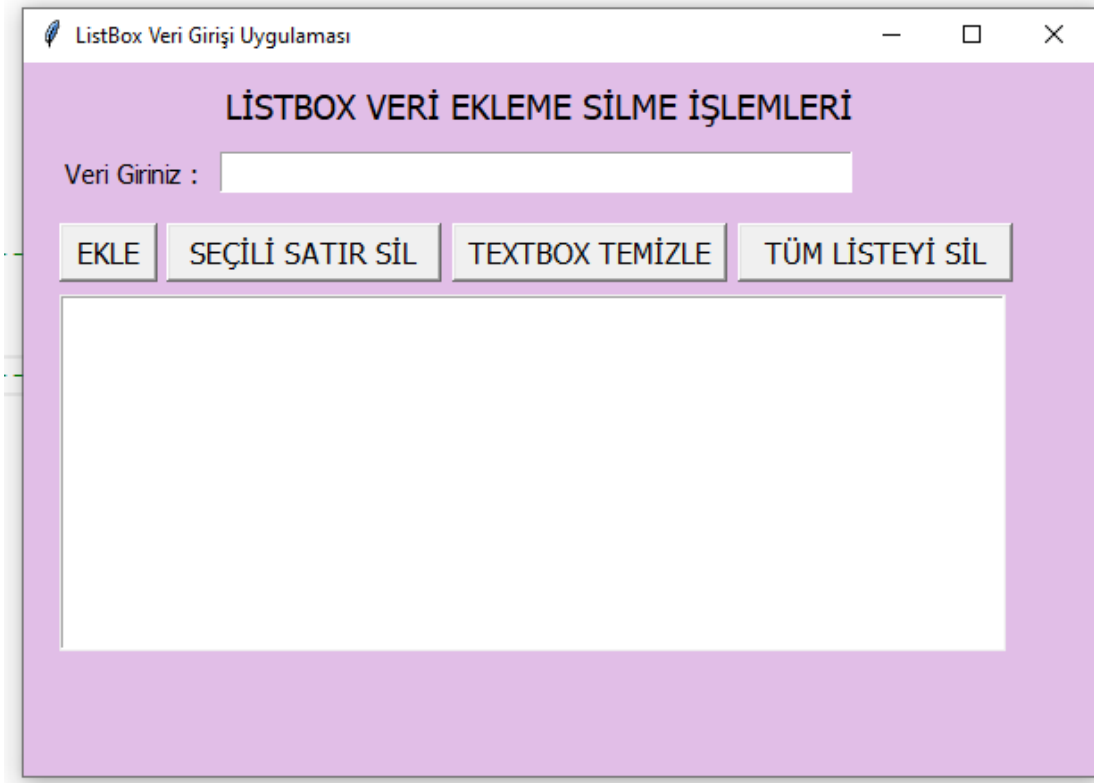
Proje Hakkında

Python uygulamalarında genel olarak Entry ve Label araçlarını yoğun olarak kullandık. Bu uygulamada verileri çoklu satır özelliği olan ListBox aracına eklemeye çalışacağız. Uygulama kodlarını incelediğimizde ekle ve silme işlemleri yer almaktadır. Bu işlemler fonksiyon yapısı kullanılarak yazılmıştır. Daha sonraki süreçte butonlara bu görevler atanmıştır. Bu butonlar tıklandığında çalışması gereken kod blokları çalıştırılmıştır. Entry aracına klavyeden girilen bilgiler ListBox aracına eklenmiştir. Bu uygulamada uygulama da silme ve temizleme işlemleri de yer almaktadır.

Projede Kullanılan Modül ve Araçlar

- Entry Aracı
- Label aracı
- Button aracı
- ListBox aracı
- Tkinter modülü

Tkinter ile Proje Tasarımı



Proje Kodları

listeye veri ekleme.py > ...

```
1  import tkinter as tk
2  pencere=tk.Tk()
3  pencere.title("ListBox Veri Girişi Uygulaması ")
4  pencere.geometry("600x400")
5  pencere.configure(background="#E1BEE7")
6  #-----
7  veriler=[]
8  def ekle():
9      s1 = ent1.get()
10     list1.insert(tk.END,s1)
11 def sil():
12     list1.delete(tk.ACTIVE)
13 def temizle():
14     #list1.delete(0,tk.END)
15     ent1.delete(0,tk.END)
16 #-----
17 for i in veriler:
18     list1.insert(tk.END,i)
19 #-----
20 baslik=tk.Label(text="LISTBOX VERİ EKLEME SİLME İŞLEMLERİ", font = "tahoma 14", bg="#E1BEE7")
21 baslik.place(x=110, y=10)
22 e1=tk.Label(text="Veri Giriniz :", font = "tahoma 11", bg="#E1BEE7")
23 e1.place(x=20, y=50)
24 ent1=tk.Entry(font = ("tahoma", 12, "bold"), width=35)
25 ent1.place(x=110, y=50)
26 b1=tk.Button(text="EKLE",command=ekle,width=5,height=1, font = "tahoma 12")
27 b1.place(x=20, y=90)
28 b2=tk.Button(text="SEÇİLİ SATIR SİL",command=sil,width=16,height=1, font = "tahoma 12")
29 b2.place(x=80, y=90)
30 b3=tk.Button(text="TEXTBOX TEMİZLE",command=temizle,width=16,height=1, font = "tahoma 12")
31 b3.place(x=240, y=90)
32 b4=tk.Button(text="TÜM LİSTEYİ SİL",command=temizle,width=16,height=1, font = "tahoma 12")
33 b4.place(x=400, y=90)
34 list1=tk.Listbox(font = "tahoma 12 bold")
35 list1.place(x=20,y=130,width=530,height=200)
36 #-----
37 pencere.mainloop()
38 #-----
```

BÖLÜM 4 FORM ÜZERİNE RESİM EKLEME UYGULAMASI

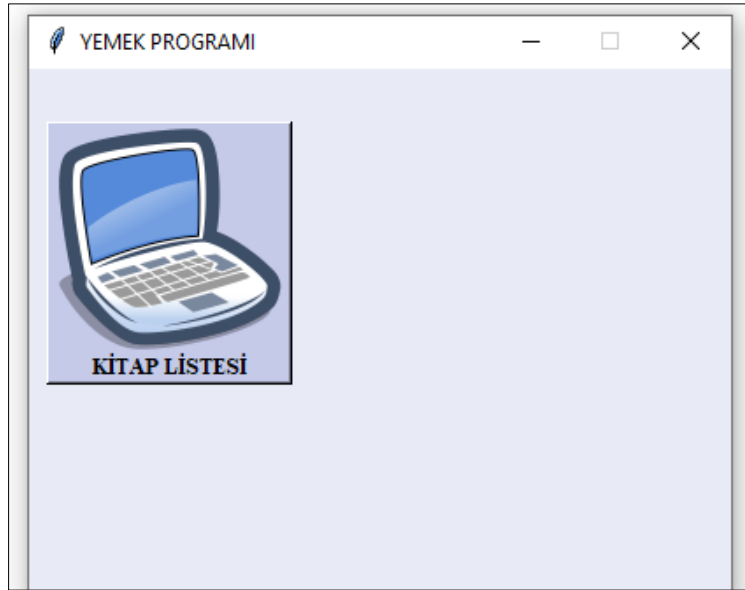
Proje Hakkında

Görsel programlama çalışmalarında resimler vazgeçilmez görsellerdir. Form üzerine eklenen görseller sayesinde programlar daha anlaşılır hale gelmektedir. Bu görseller form ekranlarını daha görsel ve daha dinamik kılmaktadır. Formlar üzerine resimler ekleyerek uygulamaların daha canlı görünmesi sağlanır. Eklene bu görseller PNG veya JPG formatlarında olabilir. Diğer resim uzantıları da kullanılabilir. Aşağıdaki uygulamada resmi buton içine eklenmiştir. Yapılan uygulamaların form ekranına resim eklenerek daha görsel hale getirilmesi sağlanır. Ayrıca resim altında metin yer almaktadır. Bu metin de font kodu ile biçimlendirilmiştir. Böylece Button içine hem görsel hem metin eklenmiş olunur.

Projede Kullanılan Modül ve Araçlar

- Button Aracı
- Tkinter modülü
- PhotoImage modülü

Tkinter ile Proje Tasarımı



 res1.png

 resim ekle 2.py

Uygulamada resim ile python dosyası aynı klasör içinde yer almaktadır. Resim başka klasörlerden dizinlerden de eklenebilir.

Proje Kodları

 resim ekle 2.py > ...

```
1 import tkinter as tk
2 from PIL.ImageTk import PhotoImage
3
4 pencere=tk.Tk()
5 pencere.title("YEMEK PROGRAMI")
6 pencere.geometry("400x300")
7 pencere.configure(background="#E8EAF6")
8 pencere.resizable(width="FALSE", height="FALSE")
9
10 resim1 = PhotoImage(file="res1.png")
11 dugme1=tk.Button(text="KİTAP LİSTESİ", image=resim1,compound="top",font="Times 10 bold",fg="black", bg="#C5CAE9")
12 dugme1.place(x=10,y=30,width=140,height=150)
13
14 pencere.mainloop()
```

BÖLÜM 5 GİRİLEN ŞİFRENİN GÜVENLİĞİNİ TEST EDEN UYGULAMA

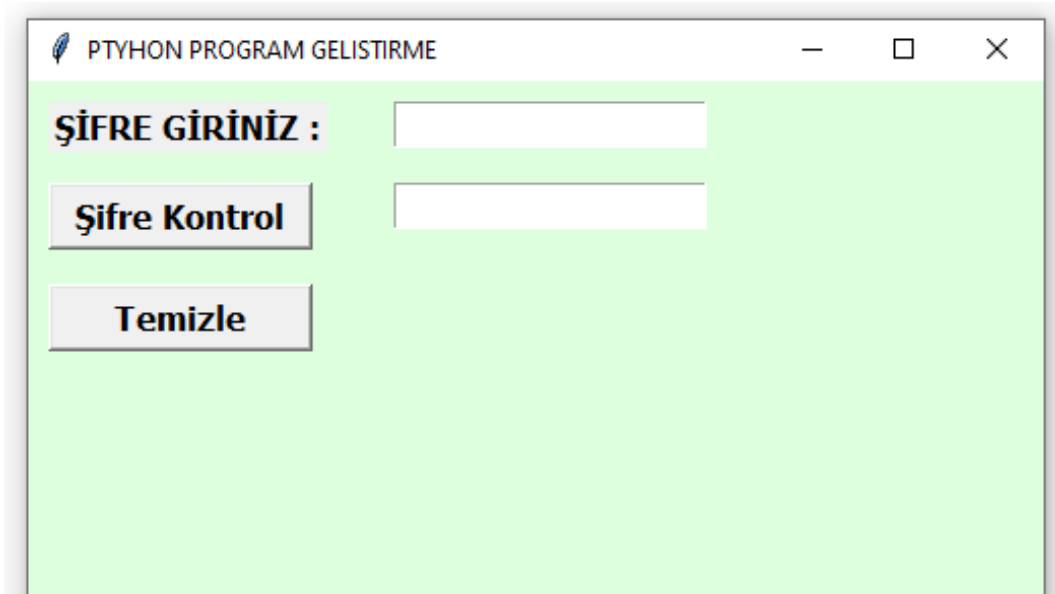
Proje Hakkında

Python uygulamalarımız incelendiğinde birçok uygulama CONSOL kısmında yapılmıştır. Yapılan bu uygulamaları aslında görsele taşımak daha faydalı olacağı düşünülmüştür. Bu uygulamamızda klavyeden yazılan 4 haneli şifrenin güvenli veya güvensiz olduğunu tespit eden bir python uygulaması yer almaktadır. Bu uygulama daha geniş kod yapısı içinde yazılabilir. Aynı zamanda karakter sayısı klavyeden de yazılabilir.

Projede Kullanılan Modül ve Araçlar

- Entry Aracı
- Label Aracı
- Button Aracı
- Tkinter modülü

Tkinter ile Proje Tasarımı



Proje Kodları

```
gorsel sifre 5.py > islem
1  import tkinter as tk
2  pencere=tk.Tk()
3  pencere.title("PYTHON PROGRAM GELISTIRME")
4  pencere.geometry("500x400")
5  pencere.configure(background="#ddffdd")
6  #-----
7  lbl1=tk.Label(text='ŞİFRE GİRİNİZ :',font = ("tahoma", 12, "bold"))
8  lbl1.place(x=10, y=10)
9  t1 = tk.Entry(width="15",justify='center',font = ("tahoma", 12, "bold"))
10 t1.place(x=180, y=10)
11 t3 = tk.Entry(width="15",justify='center',font = ("tahoma", 12, "bold"))
12 t3.place(x=180, y=50)
13 #-----
14 def islem():
15     kharfler = "qwertyuıopğüasdfghjklşızxcvbnmöç"
16     bharfler = "QWERTYUIOPĞÜASDFGHJKLŞİZXCVBNMÖÇ"
17     rakamlar = "0123456789"
18     karakter = "+-*/|"
19     rakam_s = 0
20     kharf_s = 0
21     bharf_s = 0
22     karakter_s = 0.
23     sifre = t1.get()
24
25     for harf in sifre:
26         if harf in rakamlar:
27             rakam_s = rakam_s + 1
28         if harf in kharfler:
29             kharf_s = kharf_s + 1
30         if harf in bharfler:
31             bharf_s = bharf_s + 1
32         if harf in karakter:
33             karakter_s = karakter_s + 1
34
35     if(kharf_s==0 or bharf_s==0 or rakam_s==0 or karakter_s==0):
36         t3.insert(0, str("GÜVENLİDEĞİL"))
37     else:
38         t3.insert(0, str("GÜVENLİ"))
39
40 def temizle():
41     t1.delete(0,tk.END)
42     t3.delete(0,tk.END)
43 #-----
44 olustur=tk.Button(text='Şifre Kontrol',width="12",font = ("tahoma", 12, "bold"), justify='center', command=islem)
45 olustur.place(x=10, y=50)
46 sil=tk.Button(text='Temizle',width="12",font = ("tahoma", 12, "bold"), justify='center', command=temizle)
47 sil.place(x=10, y=100)
48 #-----
49 pencere.mainloop()
```


BÖLÜM 6 FORM ÜZERİNDE LISTBOX UYGULAMASI

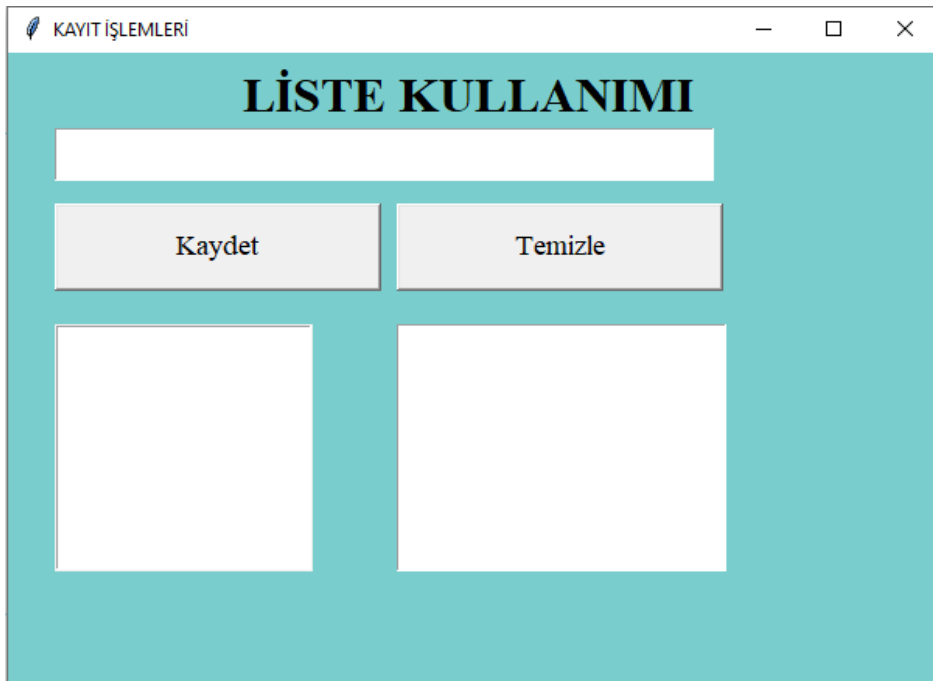
Proje Hakkında

Bu uygulamada kullanılan ekle sil butonlarıyla ListBox ve Text araçlarına veri ekleme silme işlemleri yapılmaktadır. Uygulamada ListBox ve Text araçlarına ekle butonuyla veri eklenmiş, sil butonlarıyla silinmesi sağlanmıştır. Bu uygulama sayesinde ListBox ve Text araçlarının arasındaki birkaç farkı görmüş olmaktadır. Diğer projelerde olduğu gibi yapılması istenen görevler fonksiyonlar oluşturularak yazılmıştır. Sonrasında istenilen butona eklenmiştir. Burada yer alan butonlar tıklanıldığında çalışmaktadır. Yapılması planlanan uygulamaya göre kullanılacak araçlar belirlenmektedir.

Projede Kullanılan Modül ve Araçlar

- Entry Aracı
- Label Aracı
- Text aracı
- ListBox aracı
- Button Aracı
- Tkinter modülü

Tkinter ile Proje Tasarımı



Proje Kodları

list ve text aracı kullanımı.py •

list ve text aracı kullanımı.py > ...

```
1  import tkinter as tk
2  pencere=tk.Tk()
3  pencere.title("KAYIT İŞLEMLERİ")
4  pencere.geometry("800x700")
5  pencere.configure(background="#79cdcd")
6  # PENCERE EKRANDA ORTALAMA
7  screen_width = pencere.winfo_screenwidth()
8  screen_height = pencere.winfo_screenheight()
9  pencere.geometry(f"800x700+{(screen_width//2)-400}+{(screen_height//2)-400}")
10 #-----
11 def ekle():
12     s1 = ent1.get()
13     liste.insert(tk.END,s1)
14     ent1.delete(0,tk.END)
15     metin1.insert(tk.END,s1)
16     ent1.delete(0,tk.END)
17 def sil():
18     s1.focus()
19     liste.delete(tk.ACTIVE)
20     metin1.delete(tk.ACTIVE)
21 def temizle():
22     liste.delete(0,tk.END)
23     metin1.delete(0.,tk.END)
24 #-----
25 baslik = tk.Label(text=" LİSTE KULLANIMI ",font="Times 24 bold", bg="#79cdcd")
26 baslik.place(x=200, y=10)
27 ent1= tk.Entry(width=30, font="Times 20 bold")
28 ent1.place(x=30, y=50)
29 kaydet = tk.Button(text="Kaydet", width=20, height=2, font="Times 14", command=ekle)
30 kaydet.place(x=30, y=100)
31 sil = tk.Button(text="Temizle", width=20, height=2, font="Times 14",command=temizle)
32 sil.place(x=250, y=100)
33 liste = tk.Listbox(width=27, height=10)
34 liste.place(x=30, y=180)
35 metin1 = tk.Text(width=26, height=10)
36 metin1.place(x=250, y=180)
37 #-----
38 pencere.mainloop()
```

BÖLÜM 7 FORM ÇALIŞTIĞINDA ŞİFRE OLUŞTURAN UYGULAMA

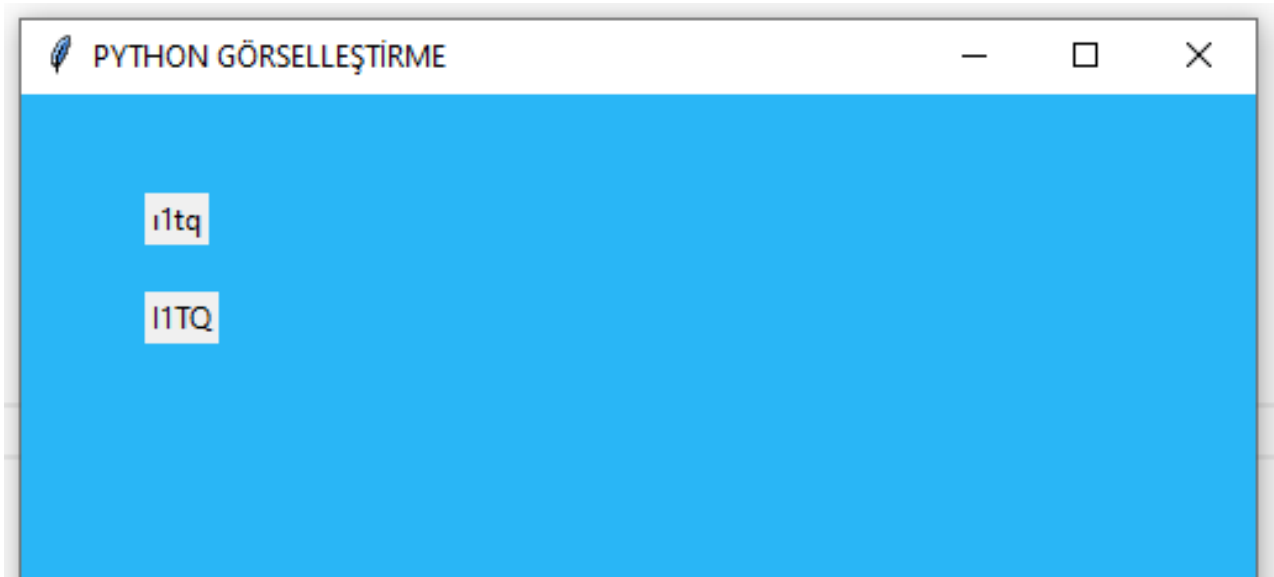
Proje Hakkında

Python programlama dili kullanarak görsel form ekranında otomatik olarak oluşan ve label araçlarına form çalıştığı anda yazdıran bir şifre oluşturma uygulamasıdır. Bu uygulama 4 haneli şifre oluşturmak için kullanılmaktadır. Üste yer alan ilk label aracında şifre karakterleri küçük harfli, alta yer alan Label aracında UPPER fonksiyonu kullanılarak tüm karakterler ekrana yazılmadan önce büyük harfe çevrilmiştir.

Projede Kullanılan Modül ve Araçlar

- Label Aracı
- Button Aracı
- Tkinter modülü
- Random modülü

Tkinter ile Proje Tasarımı



Proje Kodları

gorsel sifre 7.py ×

gorsel sifre 6.py

gorsel sifre 7.py > [x]

```
1  import tkinter as tk
2  import random
3  #-----
4  pencere=tk.Tk()
5  pencere.title("PYTHON GÖRSELLEŞTİRME")
6  pencere.geometry("500x400")
7  pencere.configure(background="#29B6F6")
8  #-----
9  liste1=[1,2,3,4,5,6,7,8,9,0]
10 liste2=["a","b","c","d","e","f"]
11 #liste4=["qwertyuiopğü1234567890"]
12 liste4="qwertyuiopğü1234567890"
13 #-----
14 s = 4
15 for x in range(0,1):|
16     password=""
17     for x in range(0,s):
18         password_char = random.choice(liste4)
19         password      = password + password_char
20     print("Rastgele Oluşan Şifreniz : ",password)
21 #-----
22 lbl1=tk.Label(text=password)
23 lbl1.place(x=50,y=40)
24 lbl2=tk.Label(text=password.upper())
25 lbl2.place(x=50,y=80)
26 #-----
27 pencere.mainloop()
28 #-----
```

BÖLÜM 8 FORMDA BUTON İLE ŞİFRE OLUŞTURMA UYGULAMASI

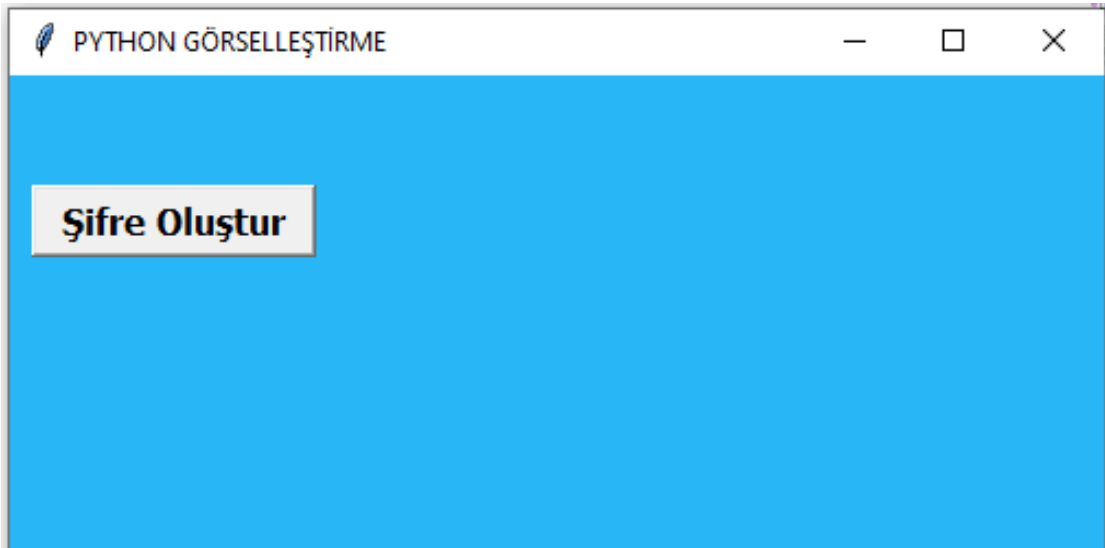
Proje Hakkında

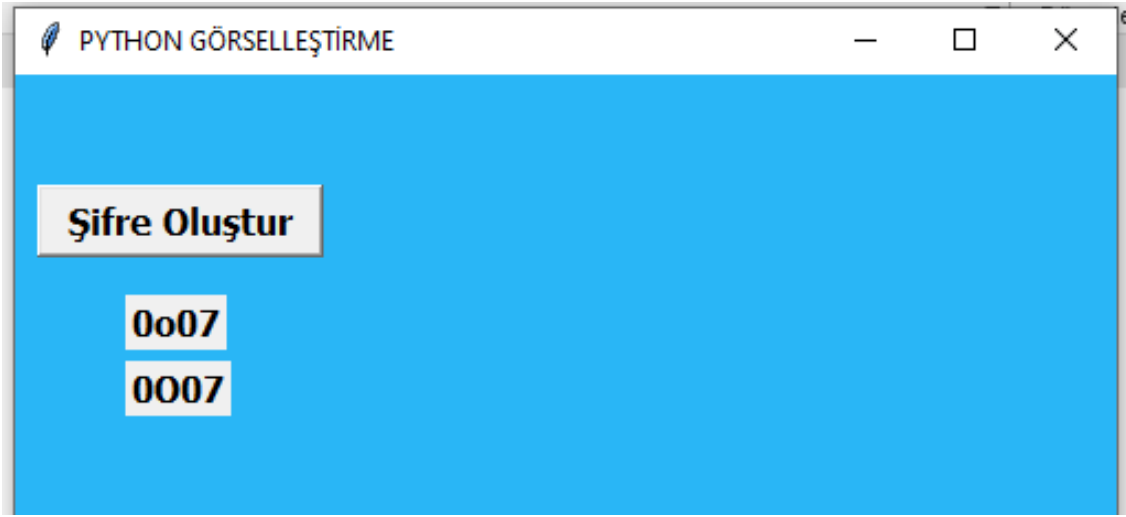
Görsel olarak çalışılan python uygulamasında butona tıklanıldığında, form ekranında yer alan iki adet Label araçlarına aynı şifre yazılmaktadır. Aynı şekilde ikinci Label aracında şifre büyük harfe çevrilerek yazılmıştır. Form ekranında şifre otomatik olarak değil, buton tıklanıldığında şifre oluşturulmaktadır. Bu uygulama admin panel ekranlarında KEY oluşturmak için kullanılabilir. Projede, iki modül kullanılmıştır. Görselleştirmek için TKINTER modülü kullanılmaktadır. Şifre karakterlerinin oluşturulması için RANDOM modülü kullanılmıştır. Yaptığımız proje ve uygulama içinde kullanılacak özelliklere göre modüller kullanılabilir. Ayrıca şifre oluşturmak için bir dizi karakter kullanılmıştır. Bu karakterler liste içinde yer almaktadır. Proje kodlarında yer alan fonksiyon, şifre oluşturmak için kullanılmaktadır.

Projede Kullanılan Modül ve Araçlar

- Label Aracı
- Button Aracı
- Tkinter modülü

Tkinter ile Proje Tasarımı





Proje Kodları

gorsel_sifre_6.py > sifre

```
1 import tkinter as tk
2 import random
3 #-----
4 pencere=tk.Tk()
5 pencere.title("PYTHON GÖRSELLEŞTİRME")
6 pencere.geometry("500x500")
7 pencere.configure(background="#29B6F6")
8 #-----
9 liste1=[1,2,3,4,5,6,7,8,9,0]
10 liste2=["a","b","c","d","e","f"]
11 liste3=["Quaresma","Yasin Sülün","Pablo Escobar","Pushkin"]
12 #liste4=["qwertyuiopğü1234567890"]
13 liste4="qwertyuiopğü1234567890"
14 #-----
15 def olustur():
16     s = 4
17     for x in range(0,1):
18         password=""
19         for x in range(0,s):
20             password_char = random.choice(liste4)
21             password = password + password_char
22             lbl1=tk.Label(text=password,font = ("tahoma", 12, "bold"))
23             lbl1.place(x=50,y=100)
24             lbl2=tk.Label(text=password.upper(),font = ("tahoma", 12, "bold"))
25             lbl2.place(x=50,y=130)
26 #-----
27 sifre=tk.Button(text='Şifre Oluştur',width="12", justify='center',command=olustur,font = ("tahoma", 12, "bold"))
28 sifre.place(x=10, y=50)
29 #-----
30 pencere.mainloop()
31 #-----
```

BÖLÜM 9 KLAVYEDEN GİRİLEN KARAKTERLERİ SAYAN UYGULAMA

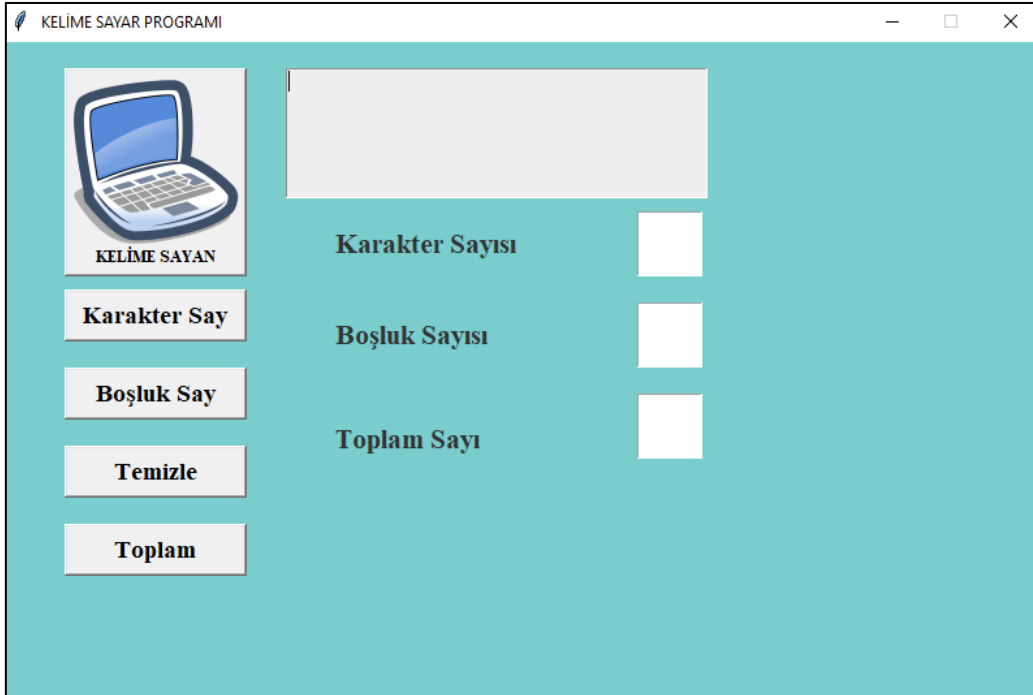
Proje Hakkında

Python programlama dili kullanılarak karakterleri sayan bir uygulama yazılmıştır. Bu karakterler arasında boşluk ta yer almaktadır. Projede bir adet resim (Button içinde kullanılmıştır), dört adet Button, bir Text aracı, üç adet Entry aracı, üç adet Label aracı kullanılmıştır. Text aracına yazılan karakterlerin sayısını bulan bir uygulamadır. Oluşturulan bu projeler önce tasarım ekranı oluşturulmuştur. Akabinde uygulamanın ne yapacağı hangi kodların nasıl çalıştırılacağı planlandıktan sonra yazılmıştır. Uygulamanın ekran tasarımı yapıldıktan sonra, uygulamanın yapacağı görevler belirlenerek programın amacına uygun kod satırları yazılarak uygulama tamamlanır.

Projede Kullanılan Modül ve Araçlar

- Entry Aracı
- ListBox aracı
- Label Aracı
- Button Aracı
- Tkinter modülü
- PhotoImage modülü

Tkinter ile Proje Tasarımı



Proje Kodları

TAM GORSEL HARF SAYAN 3.py > ...

```
1  import tkinter as tk
2  from tkinter import *
3  from PIL.ImageTk import PhotoImage
4  pencere=tk.Tk()
5  pencere.title(" KELİME SAYAR PROGRAMI ")
6  #pencere.geometry("600x500")
7  screen_width = pencere.winfo_screenwidth()
8  screen_height = pencere.winfo_screenheight()
9  pencere.geometry(f"800x700+{(screen_width//2)-400}+{(screen_height//2)-400}")
10 pencere.configure(background="#79cdcd")
11 pencere.resizable(width="FALSE", height="FALSE")
12 resim1 = PhotoImage(file="res1.png")
13 dugme1=tk.Button(text="KELİME SAYAN", image=resim1,compound="top",font="Times 10 bold",fg="black")
14 dugme1.place(x=50,y=20,width=140,height=160)
15 #-----
16 sesli_harf = "aeiıouöü"
17 sessiz_harf = "wrtpğsdfghjklşzxcvbnmqç"
18 bos_harf= " "
19 sz_sayac=0
20 bos_sayac=0
21 k1 = tk.Text(pencere, height = 5, width = 40,bg = "#eeeeee" , font="Times 12 bold",)
22 k1.place(x=220, y=15)
23 kar_sayi = tk.Label(text=" Karakter Sayı[sı]", font="Times 16 bold", fg="#333333", bg="#79cdcd")
24 kar_sayi.place(x=250,y=140)
25 bos_sayi = tk.Label(text=" Boşluk Sayı[sı]", font="Times 16 bold", fg="#333333", bg="#79cdcd")
26 bos_sayi.place(x=250,y=210)
27 kar_top = tk.Label(text=" Toplam Sayı", font="Times 16 bold", fg="#333333", bg="#79cdcd")
28 kar_top.place(x=250,y=290)
29 t2=tk.Entry(font="Times 16 bold")
30 t2.place(x=490,y=130,width=50,height=50)
31 t3=tk.Entry(font="Times 16 bold")
32 t3.place(x=490,y=200,width=50,height=50)
33 t4=tk.Entry(font="Times 16 bold")
34 t4.place(x=490,y=270,width=50,height=50)
35 #-----
36 def sesli():
37     for harf in kelime:
38         if sesli(harf):
39             sayac += 1
40     t2.insert(tk.END, str(sayac))
41 def dolu_say():
42     kelime = k1.get(1.0,tk.END)
43     sayac = 0
44     for x in kelime:
45         if x != " ":
46             sayac += 1
47     t2.insert(tk.END, str(sayac))
48 def bosluk():
49     kelime=k1.get(1.0,tk.END)
50     b_say = 0
51     for x in kelime:
52         if x == " ":
53             b_say += 1
54     t3.insert(tk.END, str(b_say))
55 def toplama():
56     sayi1 = int(t2.get())
57     sayi2 = int(t3.get())
58     t4.insert(0,str(sayi1+sayi2))
59 def tem():
60     k1.delete(1.0,tk.END)
61     k1.focus() #imleci buraya yerleştirir.
62     t2.delete(0, 'end')
63     t3.delete(0, 'end')
64     t4.delete(0, 'end')
65 #-----
66 dugme2 = tk.Button(text="Karakter Say", font="Times 14 bold", command=dolu_say)
67 dugme2.place(x=50,y=190,width=140,height=40)
68 bosluk = tk.Button(text="Boşluk Say", font="Times 14 bold", command=bosluk)
69 bosluk.place(x=50,y=250,width=140,height=40)
70 temizle = tk.Button(text="Temizle", font="Times 14 bold", command=tem)
71 temizle.place(x=50,y=310,width=140,height=40)
72 toplam = tk.Button(text="Toplam", font="Times 14 bold", command=topla)
73 toplam.place(x=50,y=370,width=140,height=40)
74 #-----
75 pencere.mainloop()
```


BÖLÜM 10 NOT HESAPLAMA UYGULAMASI

Proje Hakkında

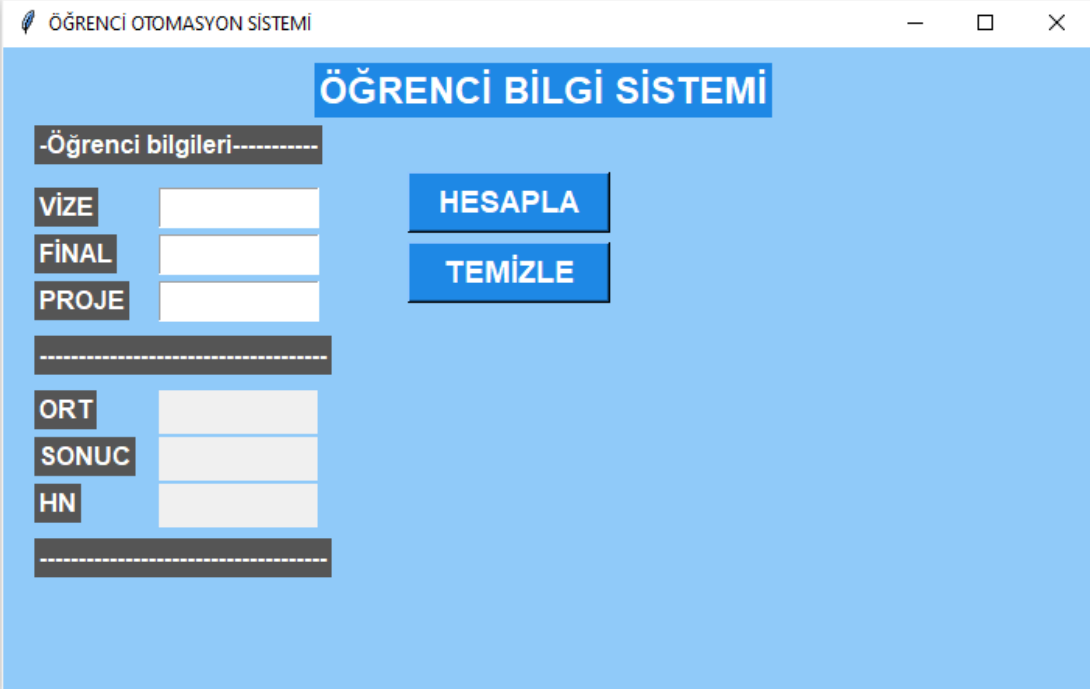
Bu bölümde öğrenci otomasyon sistemi tasarımı yer almaktadır. Bir hesap makinesi gibi çalışan uygulama içinde birden fazla fonksiyon kullanılmıştır. Üç adet klavyeden girilen notların ortalamasını, sonucunu ve harf notunu hesaplayan görsel bir uygulamadır. Python görsel programlamada ekran tasarımı kod yazarak oluşmaktadır. Tasarım ve kodlama aynı zamanda gerçekleşmektedir.

Python programlama dili ile masaüstü birçok uygulama geliştirilebilir. Bu bölümde yer alan öğrenci otomasyon uygulamasıdır. Bu uygulamada klavyeden girilen vize, final, proje notlarının aritmetik ortalamasını alarak, geçti/kaldı ve harf notunu hesaplayan bir uygulama yazılmıştır. Uygulamada iki adet buton bulunmaktadır. Bu butonlar hesaplama ve temizleme işlemlerini yapmaktadır. Hesaplama ve temizleme işlemlerinin yapılmasını sağlayan her iki işlem için fonksiyon tanımlanmıştır. Aynı şekilde not girişi yapıldıktan sonra butonların tıklanmasıyla işlemler yapılarak gerekli araçlara yazdırılması sağlanır.

Projede Kullanılan Modül ve Araçlar

- Entry Aracı
- Label Aracı
- Button Aracı
- Tkinter modülü

Tkinter ile Proje Tasarımı



Proje Kodları

```
not hesaplama.py > ...
1  import tkinter as tk
2  pencere=tk.Tk()
3  pencere.title("ÖĞRENCİ OTOMASYON SİSTEMİ")
4  pencere.geometry("700x500")
5  pencere.configure(background="#999999")
6  #-----
7  def hesapla():
8      sayi1=int(s1.get())
9      sayi2=int(s2.get())
10     sayi3=int(s3.get())
11     hesap=(sayi1+sayi2+sayi3)//3
12     s4.config(text=hesap)
13     if hesap>=50:
14         s5.config(text=("GEÇTİ"))
15     else:
16         s5.config(text=("KALDI"))
17     if hesap<50:
18         s6.config(text=("FF"))
19     elif hesap<60:
20         s6.config(text=("DD"))
21     elif hesap<70:
22         s6.config(text=("CC"))
23     elif hesap<85:
24         s6.config(text=("BB"))
25     elif hesap<101:
26         s6.config(text=("AA"))
27 def temizle():
28     s1.focus()
29     s1.delete(0,'end')
30     s2.delete(0,'end')
31     s3.delete(0,'end')
32     s4.config(text="")
33     s5.config(text="")
34     s6.config(text="")
35 #-----
36 baslik=tk.Label(text="ÖĞRENCİ OTOMASYONU", font=("arial",18,"bold"),bg="#999999",fg="#FFFFFF")
37 baslik.place(x=200,y=10)
38 ogrencibilgileri=tk.Label(text="-Öğrenci bilgileri-----", font=("arial",12,"bold"),bg="#555555",fg="#FFFFFF")
39 ogrencibilgileri.place(x=20,y=50)
40 #-----
```

```

40 #-----
41 vize=tk.Label(text="VİZE", font=("arial",12,"bold"),bg="#555555",fg="FFFFFF")
42 vize.place(x=20,y=90)
43 s1=tk.Entry(width=9, font=("arial",14,"bold"),justify="center")
44 s1.place(x=100,y=90)
45 hesap=tk.Button(text="HESAPLA",font=("arial",14,"bold"),bg="#555555",fg="FFFFFF",width=10,command=hesapla)
46 hesap.place(x=260,y=80)
47 temizle=tk.Button(text="TEMİZLE",font=("arial",14,"bold"),bg="#555555",fg="FFFFFF",width=10,command=temizle)
48 temizle.place(x=260,y=125)
49 final=tk.Label(text="FİNAL", font=("arial",12,"bold"),bg="#555555",fg="FFFFFF")
50 final.place(x=20,y=120)
51 s2=tk.Entry(width=9, font=("arial",14,"bold"),justify="center")
52 s2.place(x=100,y=120)
53 proje=tk.Label(text="PROJE", font=("arial",12,"bold"),bg="#555555",fg="FFFFFF")
54 proje.place(x=20,y=150)
55 s3=tk.Entry(width=9, font=("arial",14,"bold"),justify="center")
56 s3.place(x=100,y=150)
57 cizgi=tk.Label(text="-----", font=("arial",12,"bold"),bg="#555555",fg="FFFFFF")
58 cizgi.place(x=20,y=185)
59 ort=tk.Label(text="ORT", font=("arial",12,"bold"),bg="#555555",fg="FFFFFF")
60 ort.place(x=20,y=220)
61 s4=tk.Label(width=8, font=("arial",14,"bold"),justify="center")
62 s4.place(x=100,y=220)
63 #-----
64 sonuc=tk.Label(text="SONUC", font=("arial",12,"bold"),bg="#555555",fg="FFFFFF")
65 sonuc.place(x=20,y=250)
66 s5=tk.Label(width=8, font=("arial",14,"bold"),justify="center")
67 s5.place(x=100,y=250)
68 hn=tk.Label(text="HN", font=("arial",12,"bold"),bg="#555555",fg="FFFFFF")
69 hn.place(x=20,y=280)
70 s6=tk.Label(width=8, font=("arial",14,"bold"),justify="center")
71 s6.place(x=100,y=280)
72 cizgi=tk.Label(text="-----", font=("arial",12,"bold"),bg="#555555",fg="FFFFFF")
73 cizgi.place(x=20,y=315)
74 #-----
75 pencere.mainloop()

```

BÖLÜM 11 FORM ÜZERİNDE ZAMAN FONKSİYONLARI UYGULAMASI

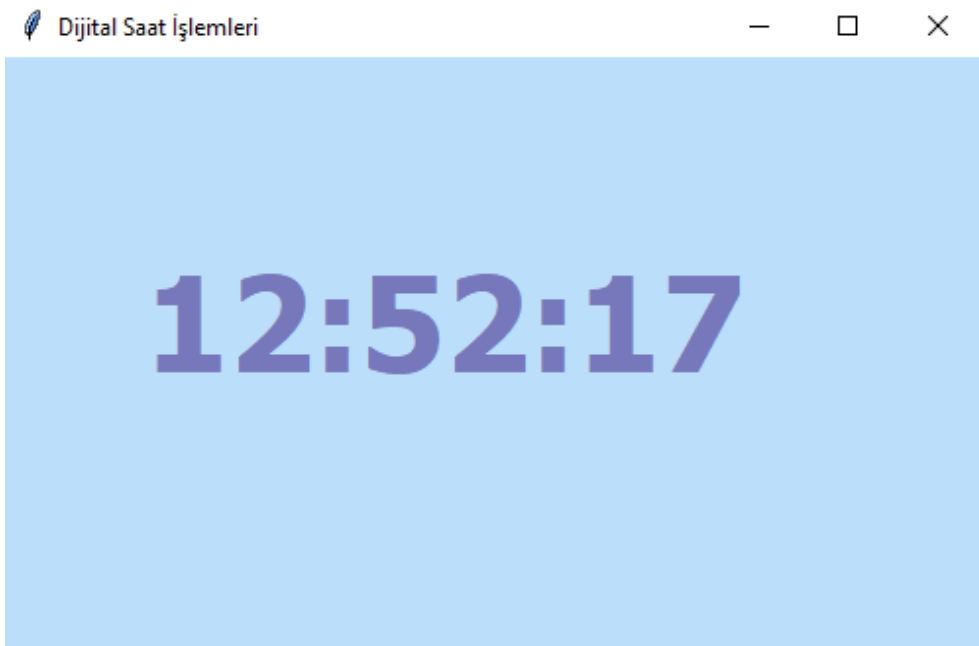
Proje Hakkında

Bir önceki uygulamada tarih fonksiyonlarını kullanmıştık. Bu uygulamada saat fonksiyonlarını kullanacağız. Buradaki amaç saati sürekli canlı tutmak ve anlık olarak ekrana basmaktır. Bu uygulamada sistem üzerinden çekilen saat sürekli kendini güncellemektedir. Daha önceki uygulamalarda formun açıp kapatılması gerekmekteydi. Form ekranda çalıştığı sürece saat kendini sürekli güncelleyecek anlık olarak sistem üzerinden saat dakika saniye değerlerini sürekli aktif olarak çekecektir. Saati ekrana sürekli olarak çekmek için bir fonksiyon yazılmıştır.

Projede Kullanılan Modül ve Araçlar

- Label aracı
- Datetime modülü
- Time modülü
- Locale Modülü
- Tkinter modülü

Tkinter ile Proje Tasarımı



12:41:53

Proje Kodları

gorsel_dj_saat.py ×

gorsel_dj_saat.py > ...

```
1  import datetime
2  import tkinter as tk
3  import time
4  import locale
5  locale.setlocale(locale.LC_ALL, 'Turkish')
6  'Turkish_Turkey.1254'
7
8  pencere = tk.Tk()
9  pencere.title("Dijital Saat İşlemleri")
10 pencere.geometry("500x300")
11 pencere.configure(background="#BBDEFB")
12
13 def saat():
14     saaat = time.strftime("%H:%M:%S")
15     s1 = tk.Label(text = saaat, font=("tahoma", 50, "bold"), fg="#7777bb", bg="#BBDEFB")
16     s1.place(x=70, y=90)
17     #s1.after(1000, saat)
18 saat()
19
20 pencere.mainloop()
```

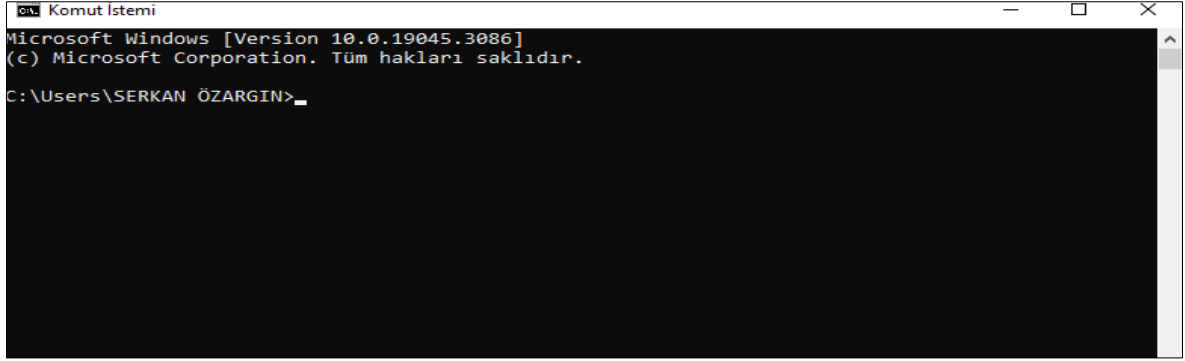
```
1  import datetime
2  import tkinter as tk
3  import time
4  import locale
5  locale.setlocale(locale.LC_ALL, 'Turkish')
6  'Turkish_Turkey.1254'
7
8  pencere = tk.Tk()
9  pencere.title("Dijital Saat İşlemleri")
10 pencere.geometry("500x300")
11 pencere.configure(background="#BBDEFB")
12
13 def saat():
14     saat = time.strftime("%H:%M:%S")
15     s1 = tk.Label(text = saat, font=("tahoma", 50, "bold"), fg="#7777bb", bg="#BBDEFB")
16     s1.place(x=70, y=90)
17     s1.after(1000, saat)
18 saat()
19
20 pencere.mainloop()
```

17. Satırda yer alan kodu not yorum satırı olarak aldığımızda form ekranında saat en son haliyle ekrana yazılmaktadır. Form her seferinde kapatılıp açıldığında saat kendini form üzerinde güncelleyecek ama sürekli olarak güncellemeyecektir. 17. Satırda yer alan kod yorum satırından çıkartılırsa bu fonksiyon sürekli olarak her saniyede çağırılarak çalıştırılacaktır. Böylece saat form üzerinde kendini sürekli güncelleyecektir.

BÖLÜM 12 GÖRSEL UYGULAMALARI EXE YAPMA

EXE Hakkında

Python programlama dilini kullanarak çeşitli uygulamalar yapılmıştır. Bu uygulamaları kod yazdığımız editörlerden çalıştırmak yerine projenin exe dosyası üzerine gelip tıklanarak çalıştırılmasıdır. Proje kodları üzerinde belli işlemler yapıldıktan sonra EXE işlemleri yapılır. EXE ye dönüştürme işlemlerinin yapılması için komut istemi üzerinde birtakım işlemlerin yapılması gerekmektedir. Bu işlemler IDE üzerinde yer alan consol ekranında da yapılabilir.



Exe Adımları

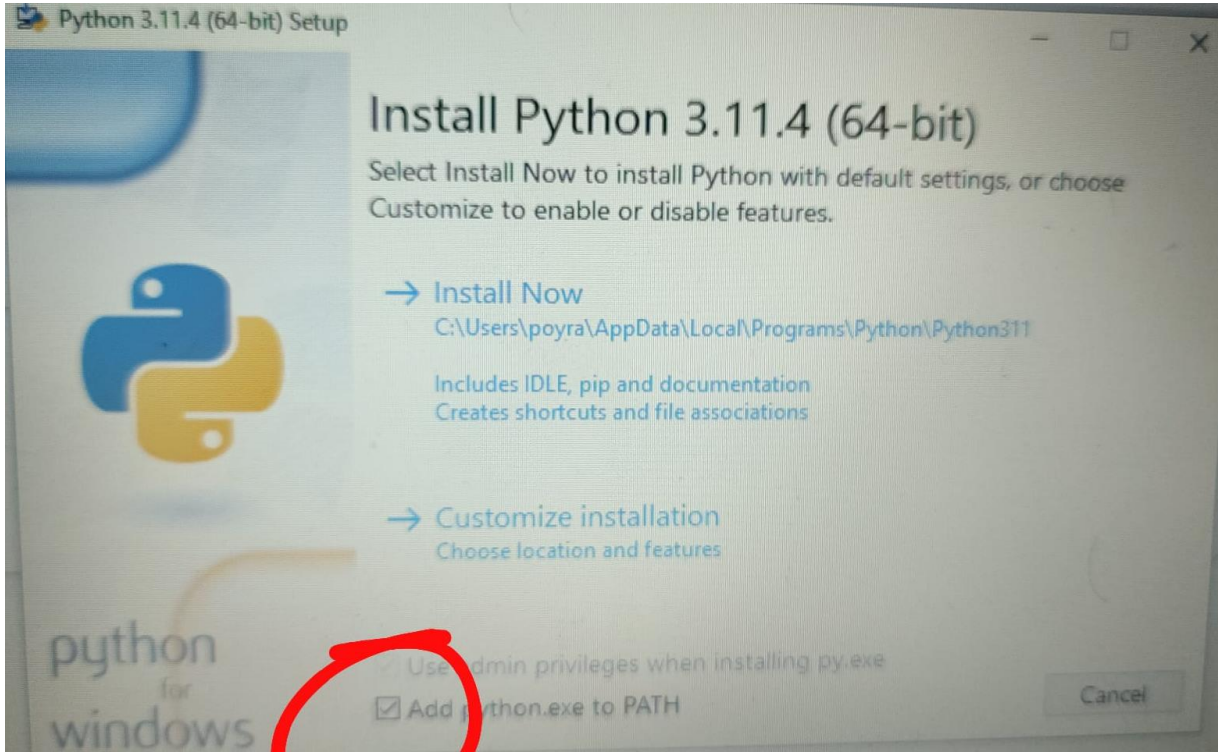
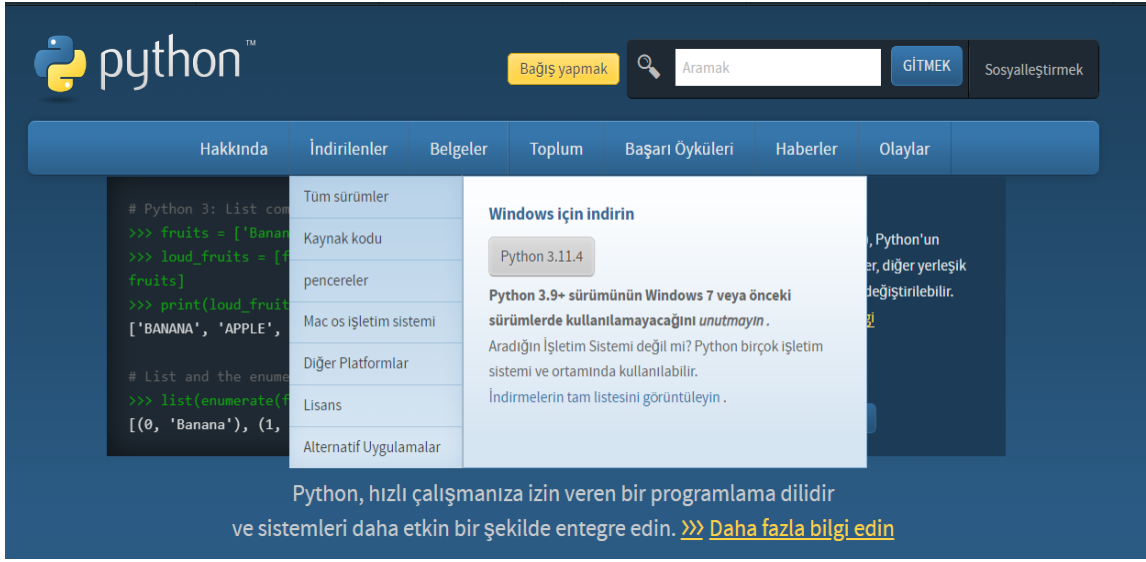
- Python 3.11.4 Versiyonunu python.org adresinden indirip kuralım.
- Komut istemini açıp python versiyonun yüklü olduğunu test edelim.
- Pip install pyinstaller komutlarını kullanarak pip paketini import edelim.
- Komut istemi üzerinde pyinstall komutunu kullanarak exe haline getirelim.

EXE Yapmanın Avantajı

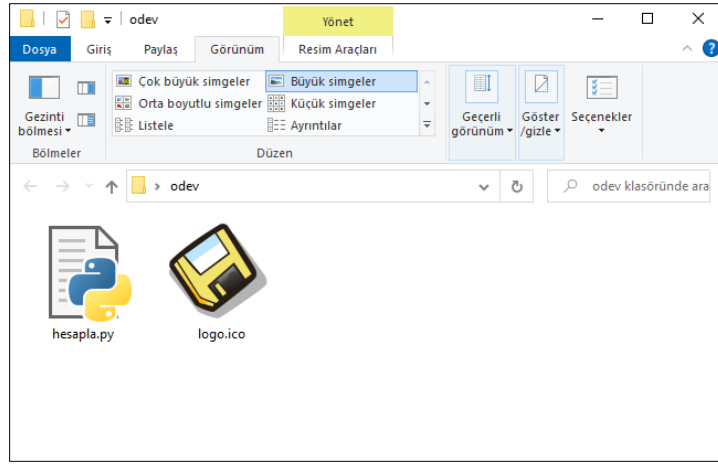
Python kodlarını proje haline getirebilmek için .EXE yada .SETUP uzantılarına çevirmek gerekmektedir. Tabi bu uzantılara çevirmek için kod uygulamasını bir dizi işlemten geçirmek gerekmektedir. Projelerin exe uzantısına çevrilmesi, exe uzantılı dosyanın tıklanarak çalıştırılmasına olanak sağlamaktadır. Aynı zamanda kodların editör tarafından açılmadan kodların görünme sininde önüne geçilmiş olunur. Böylece kodlar editörde açılmadan kodlar görünmeden uygulama exe ile çalıştırılmış olunur.

EXE' ye Dönüştürme İşlemleri

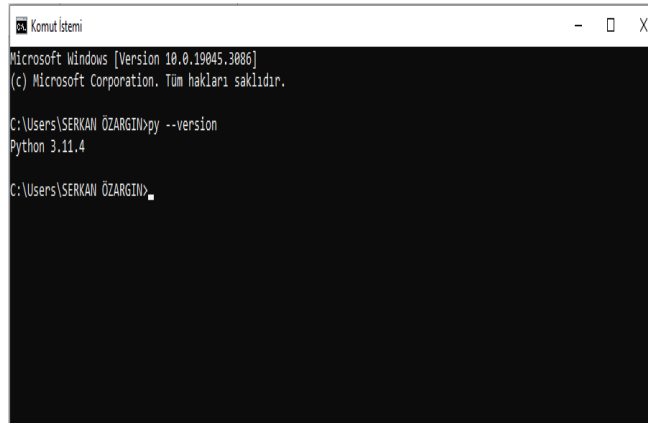
Yukarıdaki görseli incelediğimizde PYTHON konsol versiyonunu indirelim. EXE işleminin yapılması için bunu kuralım.



Yukarıdaki görseli incelediğimizde PYTHON konsol versiyonunu kurarken PATH seçeneğini işaretleyip kurma işlemine devam edelim. Kurulum işlemi bitirdikten sonra exe işlemine geçebiliriz.



Yukarıdaki görseli incelediğimizde python dosyası ile ico uzantılı küçük görsel aynı klasörde yer almaktadır. Kurulum bittikten sonra EXE dosyası üzerinde bu ico görseli yer alacaktır. Aslında bu EXE dosyasının simgesi olacak şekilde düzenlenmiştir.



Yukarıdaki görsel incelendiğinde kurulumunu gerçekleştirdiğimiz python konsol versiyonunu komut istemi üzerinden kontrol ediyoruz. Bu esnada komut istemine py versiyon kontrolü için yukarıdaki kodu yazarak diğer adıma gecemiz.



Py versiyon kontrol işlemi bittikten sonra yukarıdaki görseli gerçekleştirmek için masaüstü dizinine geçilmiştir. Bunu nedeni EXE yapacağımız dosya masaüstünde yer almaktadır. Asıl amaç EXE yapacağımız dosya dizinine geçilmesi gerekmektedir.

```
Komut İstemi

C:\Users\SERKAN ÖZARGIN\Desktop>cd odev
C:\Users\SERKAN ÖZARGIN\Desktop\odev>
```

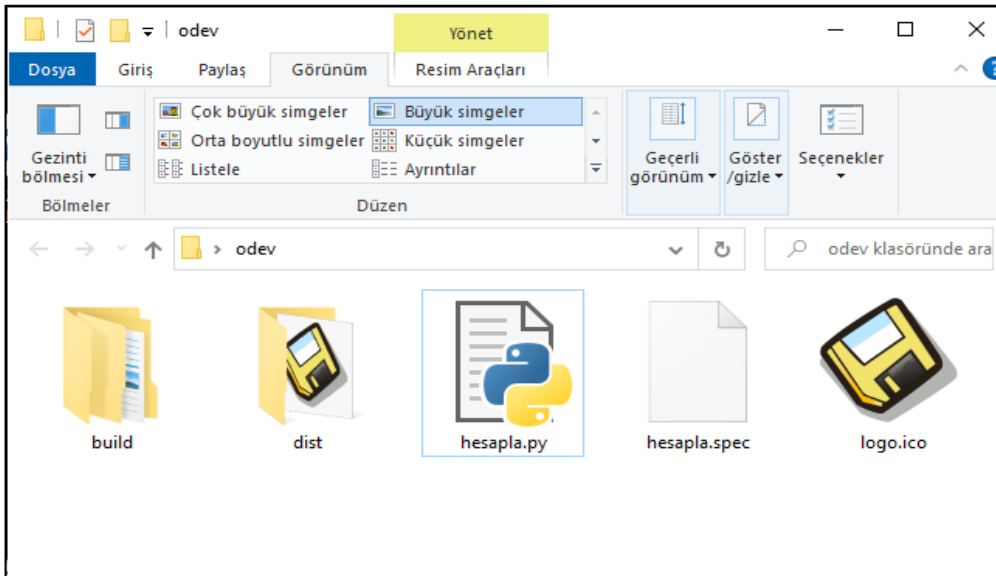
Artık EXE yapacağımız python dosyasına ulaşmanın zamanı geldi. EXE yapacağımız dosya, odev klasörünün içindedir. Bu klasör masaüstünde yer almaktadır. Yukarıdaki kod yardımıyla python dosyasının yer aldığı klasör içine ulaşılmaktadır. Böylece işlemlerimizin tamamlanmasına az kalmıştır. Artık EXE işlemini yapmak için hazırız.

```
Komut İstemi

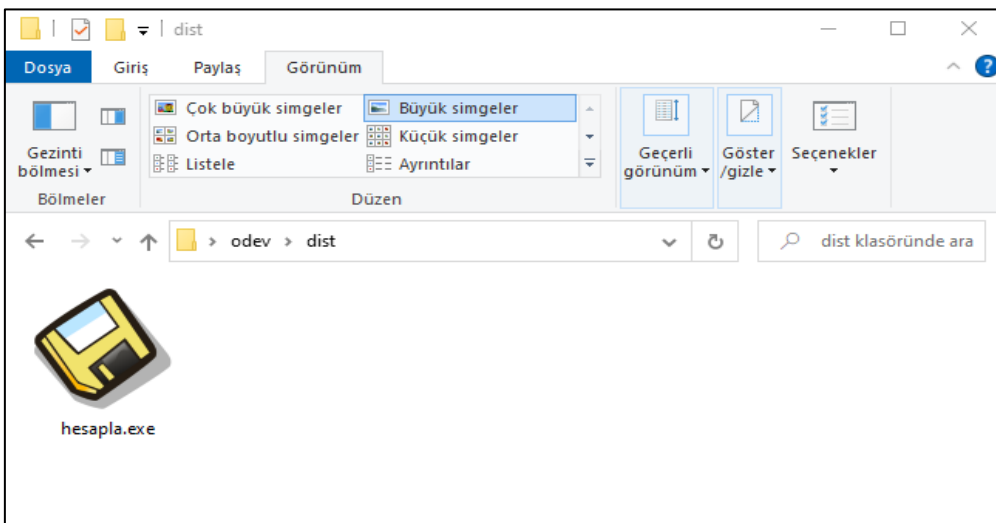
C:\Users\SERKAN ÖZARGIN\Desktop\odev>pyinstaller --onefile --noconsole --ico logo.ico hesapla.py
113 INFO: PyInstaller: 5.1
113 INFO: Python: 3.10.5
126 INFO: Platform: Windows-10-10.0.19045-SP0
127 INFO: wrote C:\Users\SERKAN ÖZARGIN\Desktop\odev\hesapla.spec
163 INFO: UPX is not available.
166 INFO: Extending PYTHONPATH with paths
['C:\Users\SERKAN ÖZARGIN\Desktop\odev']
688 INFO: checking Analysis
688 INFO: Building Analysis because Analysis-00.toc is non existent
688 INFO: Initializing module dependency graph...
691 INFO: Caching module graph hooks...
746 INFO: Analyzing base_library.zip ...
9095 INFO: Processing pre-find module path hook distutils from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks\pre_find_module_path\hook-distutils.py'.
9097 INFO: distutils: retargeting to non-venv dir 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib'
13983 INFO: Caching module dependency graph...
14152 INFO: running Analysis Analysis-00.toc
14157 INFO: Adding Microsoft.Windows.Common-Controls to dependent assemblies of final executable
required by C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\python.exe
14236 INFO: Analyzing C:\Users\SERKAN ÖZARGIN\Desktop\odev\hesapla.py
14251 INFO: Processing module hooks...
14251 INFO: Loading module hook 'hook-difflib.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14254 INFO: Loading module hook 'hook-distutils.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14255 INFO: Loading module hook 'hook-distutils.util.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14257 INFO: Loading module hook 'hook-encodings.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14352 INFO: Loading module hook 'hook-heapq.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14355 INFO: Loading module hook 'hook-lib2to3.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14410 INFO: Loading module hook 'hook-multiprocessing.util.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14413 INFO: Loading module hook 'hook-pickle.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14414 INFO: Loading module hook 'hook-platform.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14416 INFO: Loading module hook 'hook-sysconfig.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14419 INFO: Loading module hook 'hook-xml.etree.cElementTree.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14420 INFO: Loading module hook 'hook-xml.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
14567 INFO: Loading module hook 'hook-tkinter.py' from 'C:\Users\SERKAN ÖZARGIN\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks'...
29314 INFO: Copying icon to EXE
29314 INFO: Copying icons from ['C:\Users\SERKAN ÖZARGIN\Desktop\odev\logo.ico']
29428 INFO: Writing RT_GROUP_ICON 0 resource with 146 bytes
29428 INFO: Writing RT_ICON 1 resource with 57724 bytes
29430 INFO: Writing RT_ICON 2 resource with 25877 bytes
29435 INFO: Writing RT_ICON 3 resource with 67624 bytes
29436 INFO: Writing RT_ICON 4 resource with 38056 bytes
29436 INFO: Writing RT_ICON 5 resource with 21640 bytes
29437 INFO: Writing RT_ICON 6 resource with 16936 bytes
29438 INFO: Writing RT_ICON 7 resource with 9640 bytes
29439 INFO: Writing RT_ICON 8 resource with 4264 bytes
29439 INFO: Writing RT_ICON 9 resource with 2440 bytes
29440 INFO: Writing RT_ICON 10 resource with 1128 bytes
29445 INFO: Copying 0 resources to EXE
29445 INFO: Embedding manifest in EXE
29447 INFO: Updating manifest in C:\Users\SERKAN ÖZARGIN\Desktop\odev\dist\hesapla.exe.notanexecutable
29497 INFO: Updating resource type 24 name 1 language 0
29502 INFO: Appending PKG archive to EXE
32341 INFO: Building EXE from EXE-00.toc completed successfully.
```

Komut istemi üzerinde yapılan işlemler ile artık exe yapmaya hazır hale getirilmiştir. Yukarıdaki görsele yazdığımız kodu incelediğimizde, exe yapmak istediğimiz python uzantılı dosyayı yukarıdaki kod satırları ile EXE dosyasına dönüştürülmüştür.

pyinstaller --onefile --noconsole --ico logo.ico hesapla.py bu kodları incelediğimizde --onefile kodu ile sadece exe yapacağınız dosya işleme alınması sağlanır. Exe dosyası tıklandığında consol ekranı açılmasını engellemek için noconsole kodu ile exe çalıştığında programın consoldan açılması engellenmiştir.



EXE işlemi bittiğinde yukarıdaki görsel ortaya çıkmaktadır. Python dosyasının EXE dosyaları yukarıdaki gibidir. Bu dosya içinde dist dosyası içinde aşağıdaki görseldeki gibi yer almaktadır.



Yukarıdaki görselde EXE dosyasının son hali görülmektedir. Bu dosya dist dosyası içinde yer almaktadır. EXE dosyasının görseli seçilen ico uzantılı görsel olarak oluşturulmuştur. Ayrıca istenilen ico uzantılı görseller exe dosyasının görseline yukarıdaki gibi eklenebilir.

