

PYTHON ile VERİ ANALİZİ

Hafta 01

PYTHON ile VERİ ANALİZİ

- PYTHON programlama dilinin ortaya çıkışının amacı:

Yazılım üzerine çalışan Mühendisleri ve Bilgisayar Programcılarını **daha verimli ve daha üretken hale getirmek için**, yeni bir programlama dili ortaya çıkarmak amacı ile oluşturulmuştur.

PYTHON ile VERİ ANALİZİ

- Avantajları:
- Daha az program kodlama süresi: Python kodları ile programlama, genel olarak C / C ++, C# ve Java gibi programlama dillerine göre **bir kaç kat daha kısa** sürede hazırlanabilir.
- Program Kodlarının Okunma Kolaylığı: Python program kodları daha kısa sürede okunup anlaşılabilir.
- Daha az öğrenme süresi: Diğer programlama dillerine göre, öğrenme süresi daha kısa zaman alabilmektedir.

PYTHON ile VERİ ANALİZİ

- PYTHON Programlama Kodlarını ile çalışma yapmak için:
- 'Anaconda' platformu – eko sistemi kurulur.
- <https://www.anaconda.com/products/distribution>

PYTHON ile VERİ ANALİZİ

- Bütünleşik Geliştirme Ortamı(IDE) olarak
- Spyder Python yorumlayıcısı kurulur.
- veya
- Jupyter Notebook kurulur.

PYTHON ile VERİ ANALİZİ

Anaconda | Anaconda Distributio x +

anaconda.com/products/distribution

Google Chrome varsayılan tarayıcınız değil Varsayılan olarak ayarla


ANACONDA Products Pricing Solutions Resources Partners Blog Company Contact Sales

Individual Edition is now

ANACONDA DISTRIBUTION




The world's most popular open-source Python distribution platform

Anaconda Distribution

Download 

For Windows
Python 3.9 • 64-Bit Graphical Installer • 621 MB

Get Additional Installers

We use cookies

By clicking "Accept all", you agree to the storing of cookies on your device for functional, analytics, and advertising purposes.

Accept all

This website uses cookies

More choices See Our Privacy Policy

Hey! 🐍 Welcome to Anaconda. I'm here to help. What are you looking for today?

01:39 28.02.2023

PYTHON ile VERİ ANALİZİ

The screenshot displays the Anaconda Navigator desktop application. The interface is divided into several sections:

- Top Bar:** Includes the Anaconda Navigator logo, a menu (File, Help), an "Upgrade Now" button, and a "Connect" button.
- Left Sidebar:** Contains navigation links for "Home", "Environments", "Learning", and "Community". At the bottom, there is a section for "Anaconda Assistant" with a "Get started now" button, and links to "Documentation" and "Anaconda Blog".
- Main Panel:** A grid of application tiles, each representing a different tool or environment. Each tile includes an icon, a name, a version number, a brief description, and a button to either "Install" or "Launch".

Application	Version	Description	Action
DataSpell		DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.	Install
CMD.exe Prompt	0.1.1	Run a cmd.exe terminal with your current environment from Navigator activated	Launch
JupyterLab	3.4.4	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	Launch
Jupyter Notebook	6.4.12	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	Launch
Powershell Prompt	0.0.1	Run a Powershell terminal with your current environment from Navigator activated	Launch
Qt Console	5.2.2	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	Launch
Spyder	5.2.2	Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features	Launch
Datalore		Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.	Launch
Deepnote		Deepnote is a new kind of data notebook build for collaboration - Jupyter compatible, in the cloud and sharing is easy as sending a link	Launch
IBM Watson Studio Cloud		IBM Watson Studio Cloud provides you the tools to analyse and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.	Launch
ORACLE Cloud Infrastructure		OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools	Launch
console_shortcut_miniconda	0.1.1		Install
Glueviz	1.2.4	Multidimensional data visualization across files. Explore relationships within and among related datasets.	Launch
Orange 3	3.34.0	Component-based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.	Launch
powershell_shortcut_miniconda	0.0.1		Launch
PyCharm Professional		A Full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.	Launch
RStudio	1.1.456	A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.	Launch

The bottom of the image shows a Windows taskbar with several open applications, including Python-Tutorial-Co..., How does machine..., SAKARYA ÜNİVERSİ..., Gelen Kutusu (194) ..., and Anaconda Navigator. The system clock indicates 17:07 on 18.02.2023.

PYTHON ile VERİ ANALİZİ



PYTHON ile VERİ ANALİZİ

- veya
- Google Colaboratory(Colab)'ta, Jupyter Notebook için hesap açılabilir.

PYTHON ile VERİ ANALİZİ

- Google Colaboratory(Colab), Python Programlama için kurulum ihtiyacı olmayan, GPU'lar ve TPU'lar dahil olmak üzere kaynaklara, ücretsiz bir şekilde erişim sağlayan bir Jupyter Notebook platformudur.

PYTHON ile VERİ ANALİZİ

- Google Colab, özellikle makine öğrenimi, ve veri bilimi için kullanımı uygundur.
- Drive ile birlikte kullanılabilir.
- Github ile birlikte kullanılabilir.

PYTHON ile VERİ ANALİZİ

- GPU'lar, grafikleri işlemek için tasarlanmıştır.
- GPU'lar, paralel işleme konusundaki güçleri ile, Yapay zeka çalışmalarında, performansları öne çıkan işlemcilerle dönüştürülmüştür.

PYTHON ile VERİ ANALİZİ

- Google tarafından geliştirilen TPU(Tensor Processing Units)'lar ise, makine öğrenimi projeleri için özel olarak tasarlanmıştır.
- Performansları artırılarak, Yapay zeka hesaplamaları için optimize edilmiştir.

PYTHON ile VERİ ANALİZİ

- `"""`
- docstring
- paragraf halinde yorumlar için kullanılır
- `"""`

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı
- '\ndocstring\nparağraf halinde yorumlar için kullanılır\n'

PYTHON ile VERİ ANALİZİ

- # print() Fonksiyonu
- # print() Fonksiyonu bilginin veya mesajın ekranda gösterilmesi için kullanılır.
- # Python'ın built-in fonksiyonudur

PYTHON ile VERİ ANALİZİ

- kelime = 'Merhaba'
- print(kelime)

PYTHON ile VERİ ANALİZİ

- Merhaba

PYTHON ile VERİ ANALİZİ

- kelime = 'Merhaba'
- print('kelime')

PYTHON ile VERİ ANALİZİ

- kelime

PYTHON ile VERİ ANALİZİ

- `print("Bilişim Sistemleri Mühendisliği Bölümü!")`
- Ekran Çıktısı:
- Bilişim Sistemleri Mühendisliği Bölümü!

PYTHON ile VERİ ANALİZİ

- Programlama Dillerindeki Temel Kontrol Yapıları (main control structures):

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- **Sıralı Yapı**– sequential control structure

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- Karar Verme Yapısı - Selection Structure(Decision Making)
 - If Deyimi
 - Switch Deyimi – base python'da switch deyimi mevcut değildir.

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- Tekrarlı Yapı – Looping Structure
 - for ve while deyimleri

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- `sayi = 1`
- `while sayi < 10:`
- `print (sayi)`
- `sayi +=1 # satır silinirse sonsuz döngü`

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- Ekran Çıktısı:
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- `sayi = 1`
- `while sayi < 10:`
- `print (sayi)`
- `sayi =sayi +1 # artış`

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- `for sayi in range(1,10):` # 1'den 9'a kadar olan sayıların listesi
- `print (sayi)`

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- `for sayi in range(10):` # 0'den 9'a kadar olan sayıların listesi
- `print (sayi)`

PYTHON ile VERİ ANALİZİ

- Programlama Dillerindeki Temel Kontrol Yapıları:
- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

PYTHON ile VERİ ANALİZİ

- Programlama Dillerindeki Temel Kontrol Yapıları:
- `for sayi in range(10,1,-2):`
- `print(sayi)`

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
 - Ekran Çıktısı:
-
- 10
 - 8
 - 6
 - 4
 - 2

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- `for sayi in range(1,10):`
- `print ("sayılar : ", sayi)`

PYTHON ile VERİ ANALİZİ

- Programlama Dillerindeki Temel Kontrol Yapıları:
- sayılar : 1
- sayılar : 2
- sayılar : 3
- sayılar : 4
- sayılar : 5
- sayılar : 6
- sayılar : 7
- sayılar : 8
- sayılar : 9

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- `for sayi in range(1,10):`
- `print ("sayılar : ", sayi + sayi)`

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- sayılar : 2
- sayılar : 4
- sayılar : 6
- sayılar : 8
- sayılar : 10
- sayılar : 12
- sayılar : 14
- sayılar : 16
- sayılar : 18

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- `başarıNotu = 90`
- `if başarıNotu >= 65:`
- `print("dersi geçmiştir")`
- `else:`
- `print("dersin tekrar alınması gerekmektedir")`

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- Ekran Çıktısı:
- dersi geçmiştir

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- `başarıNotu = 50`
- `if başarıNotu >= 65:`
- `print("dersi geçmiştir")`
- `else:`
- `print("dersin tekrar alınması gerekmektedir")`

PYTHON ile VERİ ANALİZİ

- Programlama dillerindeki Temel Kontrol Yapıları:
- Ekran Çıktısı:
- dersin tekrar alınması gerekmektedir

PYTHON ile VERİ ANALİZİ

- `print ('Sakarya Üniversitesi')`
- `print (100)`
- `print (50.0)`

- `print (type('Sakarya Üniversitesi'))`
- `print (type(100))`
- `print (type(50.0))`

PYTHON ile VERİ ANALİZİ

- Programın Ekran Çıktısı:
- Sakarya Üniversitesi
- 100
- 50.0
- <class 'str'>
- <class 'int'>
- <class 'float'>

PYTHON ile VERİ ANALİZİ

- `print("Sakarya Üniversitesi")` # çift tırnak
- `print('Sakarya Üniversitesi')` # tek tırnak

PYTHON ile VERİ ANALİZİ

- Programın Ekran Çıktısı:
- Sakarya Üniversitesi
- Sakarya Üniversitesi

PYTHON ile VERİ ANALİZİ

- # Tek satırlık Yorum satırı oluşturma
- `print("Bilişim Sistemleri Bölümü!")` #yorum satırı

PYTHON ile VERİ ANALİZİ

- Programın Ekran Çıktısı:
- Bilişim Sistemleri Bölümü!

PYTHON ile VERİ ANALİZİ

- '''
- Çok satırlı yorum
- Birden fazla satır yorum için kullanılır
- '''
- `vizeNotu = 100`
- `finalNotu= 90`
- `print ((vizeNotu + finalNotu)/2)`

PYTHON ile VERİ ANALİZİ

- Programın Ekran Çıktısı:
- 95.0

PYTHON ile VERİ ANALİZİ

- '''
- Çok satırlı yorum
- Birden fazla satır yorum için kullanılır
- '''
- `vizeNotu = 100`
- `finalNotu= 90`
- `sonuç = (vizeNotu + finalNotu)/2`
- `print ('başarı notu: ',sonuç)`

PYTHON ile VERİ ANALİZİ

- Programın Ekran Çıktısı:
- başarı notu: 95.0

PYTHON ile VERİ ANALİZİ

- `sayio1 = 100`
- `sayio2 = 550`
- `sayio1, sayio2 = sayio2, sayio1`
- `print (sayio1)`
- `print (sayio2)`

PYTHON ile VERİ ANALİZİ

- Programın Ekran Çıktısı:
- 550
- 100

PYTHON ile VERİ ANALİZİ

- `isim =input("isminizi klavyeden giriniz ")`
- `print(isim)`

PYTHON ile VERİ ANALİZİ

- Programın Ekran Çıktısı:
- isminizi klavyeden giriniz Kadir
- Kadir

PYTHON ile VERİ ANALİZİ

- `sayi1 = int(input("sayi 1: "))`
- `sayi2 = int(input("sayi 2: "))`
- `sayi3 = int(input("sayi 3: "))`
-
- `# ortalama hesabı`
- `ortalama = (sayi1 + sayi2 + sayi3) / 3`
-
- `# sonuçların ekrana yazdırılması`
- `print("sayıların ortalaması: ", ortalama)`

PYTHON ile VERİ ANALİZİ

- Programın Ekran Çıktısı:
- sayi 1: 20
- sayi 2: 30
- sayi 3: 40
- sayilarin ortalamasi: 30

PYTHON ile VERİ ANALİZİ

- `sayi_1 = int(input("Birinci Sayıyı giriniz: "))`
- `sayi_2 = int(input("İkinci Sayıyı giriniz: "))`
- `#dört işlemin sonucu`
- `sonuc = sayi_1 + sayi_2`
- `print('Sayıların Toplamı: ', sonuc)`
- `print('Sayıların Farkı : ', sayi_1 - sayi_2)`
- `print("Sayıların Çarpımı:", sayi_1 * sayi_2)`
- `print("Sayıların Bölümü :",sayi_1 / sayi_2)`

PYTHON ile VERİ ANALİZİ

- Programın Ekran Çıktısı:
- Birinci Sayıyı giriniz: 10
- İkinci Sayıyı giriniz: 2
- Sayıların Toplamı: 12
- Sayıların Farkı : 8
- Sayıların Çarpımı: 20
- Sayıların Bölümü : 5.0

PYTHON ile VERİ ANALİZİ

- `import os`
-
- `print ("Kullanıcı adı: ", os.environ["USERNAME"])`
- `print ("Bilgisayar adı: ", os.environ["COMPUTERNAME"])`
- `print ("Ev dizini: ", os.environ["HOMEPATH"])`
- `print ("İşlemci: ",
os.environ["PROCESSOR_IDENTIFIER"])`
- `print ("İşlemci sayısı: ",
os.environ["NUMBER_OF_PROCESSORS"])`
- `print ("İşletim sistemi:", os.environ["OS"])`

PYTHON ile VERİ ANALİZİ

- Kullanıcı adı: cedim
- Bilgisayar adı: DESKTOP-IM5V43G
- Ev dizini: \Users\cedim
- İşlemci: Intel64 Family 6 Model 69 Stepping 1, GenuineIntel
- İşlemci sayısı: 4
- İşletim sistemi: Windows_NT

PYTHON ile VERİ ANALİZİ

- # harflerin toplamı
- from collections import Counter
- print (Counter('Bilisim Sistemleri Bölümü'))

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- Counter({'i': 5, 'l': 3, 'm': 3, 'B': 2, 's': 2, ' ': 2, 'e': 2, 'ü': 2, 'S': 1, 't': 1, 'r': 1, 'ö': 1})

PYTHON ile VERİ ANALİZİ

- # harflerin toplamı
- import collections
- print (Counter('Bilisim Sistemleri Bölümü'))

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- Counter({'i': 5, 'l': 3, 'm': 3, 'B': 2, 's': 2, ' ': 2, 'e': 2, 'ü': 2, 'S': 1, 't': 1, 'r': 1, 'ö': 1})

PYTHON ile VERİ ANALİZİ

- `import sys` # sys modülü veya kütüphanesi
- `print (sys.version)`

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]

PYTHON ile VERİ ANALİZİ

- `import sys` # sys modülü veya kütüphanesi
- `sys.version`

PYTHON ile VERİ ANALİZİ

- Ekran Çıktısı:
- 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]

PYTHON ile VERİ ANALİZİ

- `import sys # Jupyter Notebook`
- `print (sys.version)`
- Ekran Çıktısı:
- 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]

PYTHON ile VERİ ANALİZİ

Kaynaklar:

<https://www.anaconda.com/distribution/>

Beginning Programming with Python, John Paul Mueller, 2014.

<https://realpython.com/jupyter-notebook-introduction/>

<https://github.com/rasbt/python-machine-learning-book/blob/master/code/cho1/README.md>

<https://www.datacamp.com/blog/tpu-vs-gpu-ai>https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html

<https://www.dataquest.io/blog/jupyter-notebook-tutorial/>