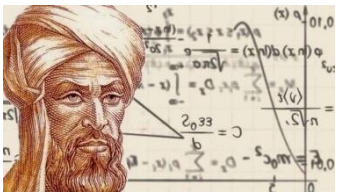
TEMEL SEVİYE PYTHON PROGRAMLAMA

1. GİRİŞ / ALGORİTMA

A. Algoritma Nedir?



El Harizmi(780-850)

» 700-800'lü yıllarda El Harizmi tarafından ortaya atılan algoritma kavramı, problemlerin adım adım çözüm yolunu gösteren süreçler veya işlemlerdir.

» Günlük hayatta plan kavramı ile tanımlanabilir.

B. Algoritma Kullanımı ve Hazırlanması

» Algoritma, yapılacak işlemleri sırasıyla gösteren maddeleri ifade eden basit bir planlama şeklidir.

» Problemin nasıl bir yol çizilerek çözüleceğini sade bir biçimde gösterir.

» Akış şemalarıyla birlikte görselleştirilebilir.

» Projeye bütüncül ve yalın olarak bakmayı sağlar.

» Bir algoritma hazırlanırken aşağıdaki süreçler sırasıyla ve dikkatle izlenir.

Analiz yapılır → Çözüm yolu seçilir → Algoritma belirlenir → Basit İfadeler → BAŞLA ile başlar → BİTİR ile biter → GİT ile akış komutu → Test

» Bir algoritma 3 farklı yolla ifade edilebilir:

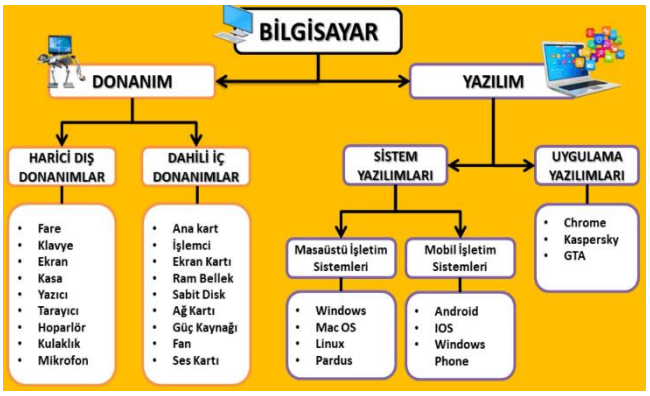
» 1. Düz metin ifadeleri: Günlük konuşma diliyle yapılan algoritma ifadeleridir.

» 2. Kaba-kod (Pseudo-code): Konuşma mantığıyla yazılıp daha çok komutlarla ifade edilen algoritma yazım şeklidir.

» 3. Akış şemaları (Flow charts): Çizimlerle algoritma ifade edilmesidir.

2. BİLGİSAYAR VE PYTHON

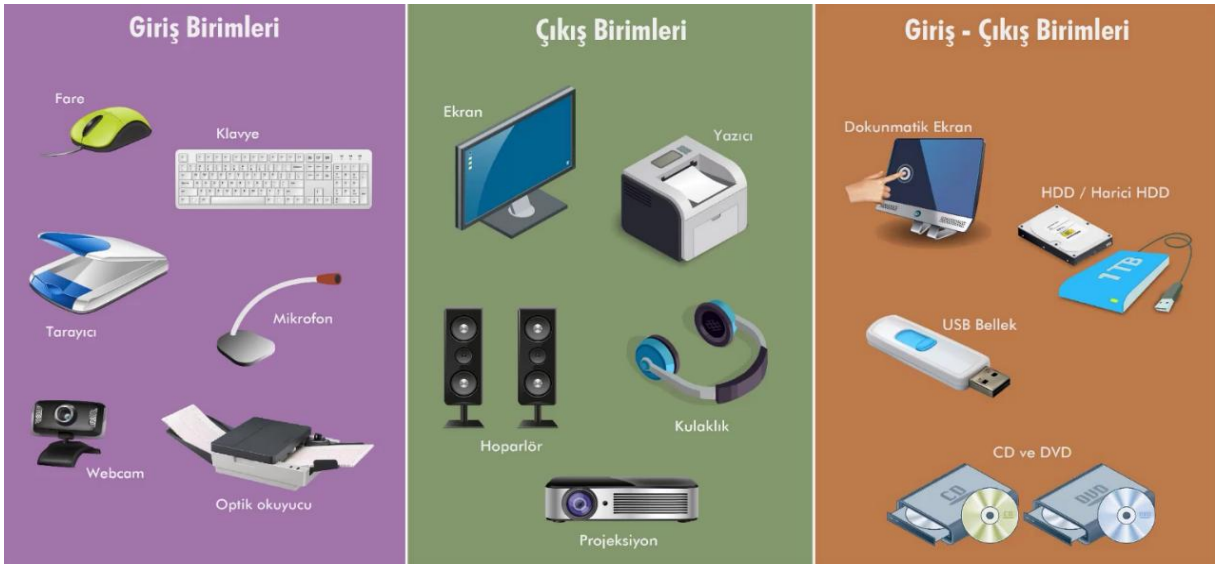
A) Bilgisayar ve Bileşenleri



» Dışarıdan aldığı verilerle mantıksal ve matematiksel işlemler yapan, yaptığı işlemlerin sonucunu saklayabilen; sakladığı bilgilere istenildiğinde ulaşılabilen ve geri verebilen makineye bilgisayar denir.

» Bilgisyarlar, donanım(hardware) ve yazılım(software) bileşenlerinden oluşur.

» Bilgisayarın gözle görülür ve elle tutulabilir her türlü fiziksel parçası bir donanım birimidir.



» Donanım birimleri; giriş birimleri, çıkış birimleri, bellek birimleri, depolama birimleri ve bütün donanım birimleri yöneten mikroişlemcilerden (CPU) oluşur.

» Yazılım birimleri ise donanım elemanlarını çalıştıran her türlü programlardır. Kullanıcıdan aldığı girdileri, donanım birimlerine iletip aldığı dönütle çıktıyı ekrana verir.

B) Program ve Programlama Dili

» Bilgisayara belirli bir işlemi veya işlemleri gerçekleştirmesi için verilen komutlara program denir.

» Programlar, bir programla dili (Python, java, c++) ile yazılır.

» Programlama dilleri, günümüzde daha fonksiyonel hale getirilerek daha çok proje üretimi sağlarlar.

C) Makine Dili

» Bilgisayarların ana yapısında (işlemci, hafıza kartı, ...) transistör adı verilen bir bileşen kullanılır.

» Transistörler, iki konumlu anahtar gibi davranarak elektriği geçirdiğinde '1' değerini geçirmediğinde '0' değerini döndürerek cihaz içinde veri üretir.

» Makine dili ise yukarıdaki verilerin birleştirilmesiyle oluşturulmuş bir alfabe niteliği taşır.

» İkili sayı sistemi (1-0) kullanarak bilgisayarlarla iletişim sağlanabilir ancak pratikte uygulanması imkansıza yakındır.

- Bu yüzden bilgisayarlarla kullanıcı arasında ara programlar (derleyici, yorumlayıcı, ...) kullanılarak bu iletişim sağlanır.

D) Derleyici ve Yorumlayıcı

- Derleyici ve yorumlayıcı, bir programlama dilindeki komutları makine diline çeviren ara programlardır.
- Derleyiciler, bir kaynak kodunu bütün olarak alır ve hataları toplu bir şekilde gösterir.
- Derleyici kullanan bazı diller: C ailesi, Fortran, Pascal, ...
- Yorumlayıcılar ise anlık olarak hata takibi yapar ve anında dönüt verir.
- Yorumlayıcı kullanan bazı diller: Python, Java, Ruby, PHP, ...

E) Kaynak Program (Source Code)

- Bir programlama dili ile yazılıp henüz derlenmemiş olan programlar kaynak kod olarak adlandırılır.
- Kaynak kodlar yazıldıkları programlama dillerine özgü uzantılar alır.
- C++ (.cpp), Python (.py), Java (.java), ...
- Kaynak kod üzerinde değişiklik ve düzenleme yapmak, olası hataları fark etmek ve denetlemek için bir editör programı kullanılır.
- Python dilinin bazı editörleri şunlardır: Jupyter, IDLE, VsCode, Sublime Text, Atom, ...

F) Çalıştırılabilir Dosya

- Yukarıdaki süreçlerden sonra artık çalıştırılmaya hazır olan dosyalar verilen isimdir.
- Windows tabanlı işletim sistemlerinde çalıştırılabilir dosyaların uzantısı .exe dir.
- Dosyayı çalıştırmak için 'run' komutunun verilmesi yeterlidir.

G) Python

Guido van Rossum

» Hollandalı bir bilgisayar programcısı olan Guido van Rossum tarafından 1991'de oluşturulmuştur.

» Python; nesne tabanlı (OOP), yorumlamalı ve yüksek seviyeli bir programla dilidir.

» Girintilere dayalı basit bir sözdizimine (syntax) sahiptir.

» Python, kolay kavranılabilir sözdizimi ile insan diline yakın yüksek seviyeli ve açık kaynak kodlu bir programlama dilidir.

» Python ile oyun tasarımı, veri analizi, robotik uygulamalar gibi birçok proje yapılabilir.

3. DEĞİŞKENLER

A) Değişken Nedir?

» İçinde veri saklanan yeri veya alanı ifade eden bellek alanlarına değişken denir.

» Değişkenler, sayısal (matematiksel) veya sözel (metinsel) veriler tutar.

» Bir değişkene değer ataması '=' operatörü ile yapılır.

» '=' operatörünün sağındaki kısım değer (value), solundaki kısım ise değişken olarak isimlendirilir

» (A = 1923), parantez içindeki kısımda A değişkenine 1923 sayısı matematiksel olarak atanmıştır

» (B = '1923'), burada ise B değişkenine '1923' ifadesi sözel olarak atanmıştır.

B) Değişken İsimlendirme

» Bir değişken isimlendirilirken ya bir harfle ya da alt çizgi (_) ile başlamalıdır.

» Alt çizgi haricinde boşluk da dahil olmak üzere hiçbir özel karakter kullanılmamalıdır.

» Herhangi bir Python komutu (keywords) bir değişkene isim olarak verilemez.

- Python komutlarına, ‘import **keyword**’ ardından ‘**keyword.kwlist**’ yazılarak erişilebilinir.
- Her ne kadar Python kullanırken Türkçe karakterler sorun çıkarmasa da kullanılmaması tavsiye edilir.
- Python büyük-küçük harf duyarlılığına (case sensitive) sahip bir dildir.

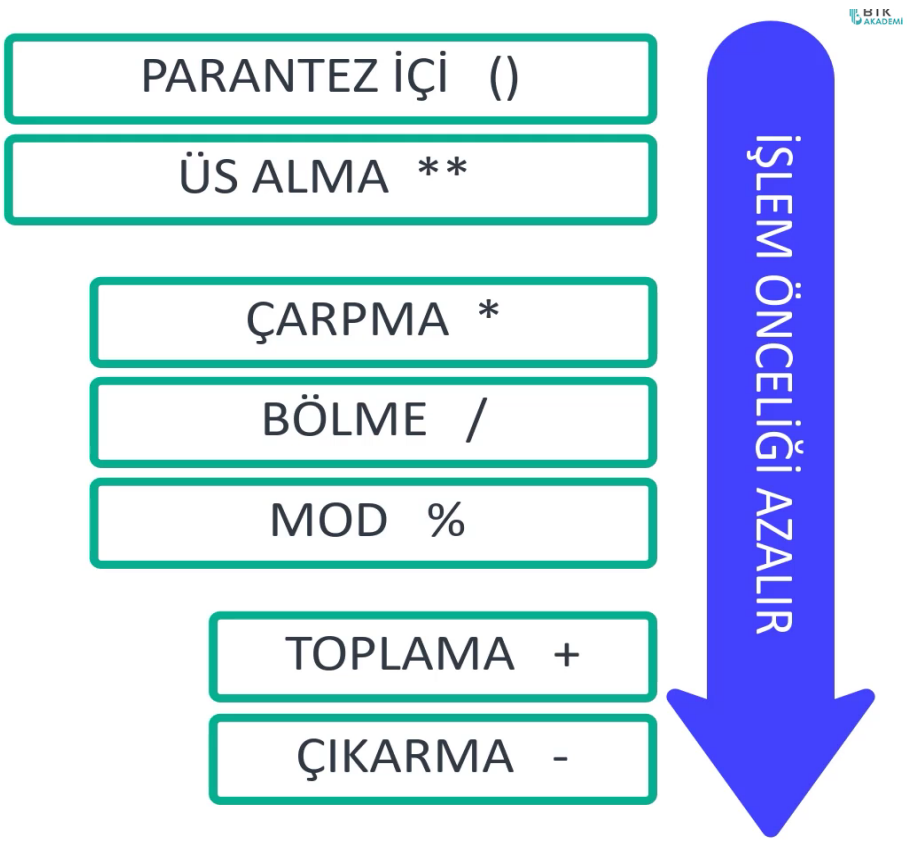
C) Değişkenleri Birbirine Atama

- İki değişkenin değerini birbirine atarken üçüncü bir değişken tanımlanarak bu işlem yapılabilir.
- Python’da bu işlem biraz daha basit bir şekilde yapılabilir
- A = 3 ve B = 8 değişkenleri A,B = B,A ile birbirinin değerini alabilir. Son durumda A’nın yeni değeri 8, B’nin yeni değeri ise 3 olmuştur.
- Matematiksel değer atamalarını kolayca yapmak için A = A + B yerine A += B yazılabilir.

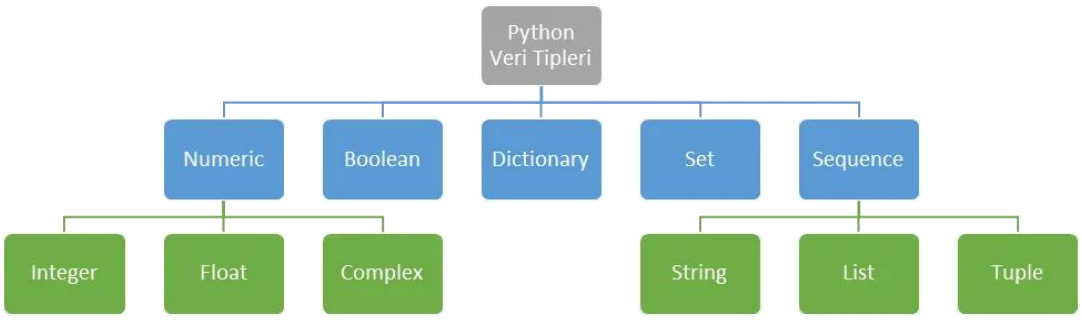
D) Operatörler

Operatör	Görevi
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Kalan
**	Üs Alma
//	Tam Sayı Bölme

E) İşlem Önceliği



F) Veri Tipleri



- Python'da veri tipi type(degiskenAdi) komutu ile öğrenilebilir.

G) Değişken Dönüsümleri

- int(8.9) ile ondalıklı kısım atılır ve yeni değer 8 olur.
- float(2) ile ondalıklı kısım eklenir ve yeni değer 2.0 olur.
- Matetiksel ifadeler string bir ifadeye çevrilebilir.
- str(7) ile yeni değer "7" olur ve bu şekilde gösterilir.
- Sözel ifadeleri '+' operatörü ile yan yana birleştirebiliriz.

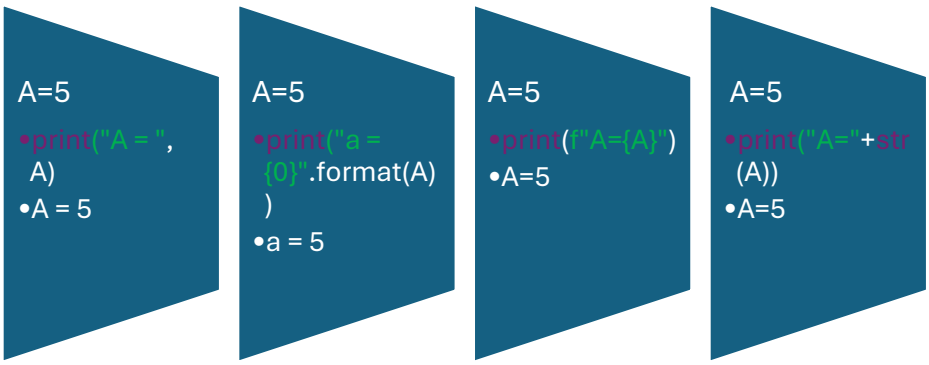
H) Yorum Satırı

- Kaynak kodun işleyişine herhangi bir değişiklik yapmaz.
- Diyez sembolü (#) ile yorum satırı eklenir.
- Python'da tek satırlık yoruma izin verilir.
- Karmaşık ve uzun kodlarda bir anımsatıcı olması bakımından çok önemli bir yere sahiptir.

4. VERİ İŞLEMLERİ

A) Veri Giriş ve Çıkış Komutları

- isim = input(“İsminiz:”) şeklinde bir komutla isim değişkenine kullanıcıdan alınan veriyle değer atanır.
- input() komutu önüne float(), int() gibi parametreler alarak kullanıcıdan alınan değerin tipi dönüştürülebilir.
- print(‘yusuf’) komutu ile ekrana yusuf yazdırılır.
- print() komutu içinde sep= parametresi kullanarak bazı kelime ayırma işlemleri yapılabilir.
- print(“y”, “u”, “s”, “u”, “f”, sep= ‘-’) komutu ekrana y-u-s-u-f yazdırır.
- print(“y”*3) komutu ile ekrana yyy yazdırılır.
- end= ‘ ’ parametresi ise bir alt satıra geçmeyip print ifadeiyle yazdırılan verinin hemen yanından devam eder.
- print(“Yusuf\nEmre”) komutu ile alt alta olacak şekilde ilk satıra Yusuf, alt satıra ise Emre yazdırılır.
- n yerine t yazarak yan yana olacak ve araya bir tab boşluk bırakılır (Yusuf Emre).
- print(‘\’) ifadesi hata verir ve bu durumu düzeltmek için iki tane \\ yazılmalıdır.
- Bir değişkenin değerini ekrana yazdırmak için farklı yollar kullanılabilir. A = 5 verisini ele alalım:



5. KOŞUL VE KARŞILAŞTIRMA OPERATÖRLERİ

A) Koşul İfadeleri

if : Tek koşula bağlı gerçekleşecek işlemler için

if else : Çift koşula bağlı gerçekleşecek işlemler için

if elif else : Çoklu koşula bağlı gerçekleşecek işlemler için

B) Karşılaştırma Operatörleri

Karşılaştırma Operatörü	Anlamı	Örnek
==	Eşittir	a == 20
!=	Eşit değildir	a != 20
>	Büyüktür	a > 20
<	Küçüktür	a < 20
>=	Büyük eşittir	a >= 20
<=	Küçük eşittir	a <= 20

C) Mantıksal Operatörler

MANTIKSAL OPERATÖRLER

İşlem	Operatör	Örnek	Ekran Çıktısı
Mantıksal VE	and	(3<5) and (7>3)	True
Mantıksal VEYA	or	(3<5) or (7<3)	True
DEĞİL	not	not (5<10)	False

6. DÖNGÜLER(LOOPS)

A) Döngüler

- Döngüler, yüzlerce satır aynı görevde olan komutları birkaç satırda daha pratik bir şekilde uygulamak için oluşturulmuş bir yapıdır.
- Python'da for ve while olmak üzere iki çeşit döngü bulunur.
- for döngüsünde, döngü sayısı önceden belli olup döngü sayısına varana kadar devam eder.
- while döngüsünde ise döngü sayısı belli olmayıp bir koşula bağlıdır ve koşul gerçekleşene kadar devam eder.

B) Listeler

- Birden fazla değişkeni içinde barındırabilen sıralı eleman dizisidir.
- Listeler farklı bir veri tipini temsil eder.
- İçerisinde hem string hem de sayısal ifadeler yer arabilir.
- Köşeli parantezlerle oluşturulur. Listenin elemanları birbirinden virgül kullanılarak ayrılır. Liste = ['ali', 2, 'veli', 3, 5] şeklinde liste oluşturulabilir.
- Bu kullanımda ekrana listenin elemanlarını yazdırmak istersek çıktı köşeli parantez içinde gelir.
- String ifadeler aslında bir liste belirtir. List = "yusuf" gibi.

C) in / not in komutları

- Bir liste içinde bir elemanın olup olmadığını gösteren komutlardır.
- Bool bir değer (True, False) döndürürler.

» Print("elemanAdi" in listeAdi) komutu ile ekrana listeAdi listesinde elemanAdi isimli elemanın olup olmadığını gösteren bool değeri yazılır.

D) range() Fonksiyonu

» Sayı üretmek, istenilen aralıktaki sayıları üretmek ve for döngüsünde döngü sayısını belirlemek için kullanılan fonksiyondur.
» İlk değer atamasının yapılması zorunludur.
» range(0,5) komutunda ilk değer 0'dan başlar ve 5 dahil olmayacak şekilde 5'e kadar olan sayıları birer birer arttırarak değerleri üretir. 0,1,2,3,4.
» Artış miktarı default olarak birer birer gerçekleşir.
» range(5) komutunda ise ilk değer default olarak 0'dan başlar. Burada da sayılar birer birer artarak 5'e kadar (5 yine dahil değil) üretilir. 0,1,2,3,4.
» range(10,1,-2) komutunda ise 10'dan başlayarak ikişer ikişer azalarak 1'e kadar (1 dahil değil) olan sayılar oluşturulur. 10, 8, 6, 4,2.
» for a in range (1,9,3): gibi bir komutla for içinde kullanılır. 1, 4, 7.
» list () komutunun içinde kullanılarak bir liste oluşturulabilir. list (range(0,5)) komutuyla [0, 1, 2, 3, 4] oluşturulur.

E) for Döngüsü

» Örnek bir for döngüsü oluşturma:
» for a in range(0,5):
        print (a)
» 'a' bir döngü değişkenidir ve genelde tekrar sayısı için kullanılır.
» İlgili ifadeler de yazıldıktan sonra ':' ile for döngüsü başlar.
» Döngü içerisindeki işlemler bir tab boşluğu bırakılarak yapılmalıdır.
» Döngünün çıktısı:
0
1
2
3
4

F) while Döngüsü ve break ile continue Komutları

- Döngü bir şartı gerçekleşene kadar devam eder.
- while True ile sonsuz döngü oluşturulur.
- Sonsuz döngüden çıkmak için ise break komutu kullanılır.
- Döngüden çıkmak istemediğimiz ve döngünün bazı anlarda çalışmamasını istedğimiz durumlarda ise continue komutu kullanılır.

7. FONKSİYONLAR

A) Fonksiyon Nedir ?

- Fonksiyon, bir programda belli bir görevi yerine getiren bir kod parçasıdır. Bir nevi legoya benzetilebilir.
- Uzun kod satırlarını daha basite indirgemeyi ve programın daha sade görünmesini sağlar.
- Kod tekrarını önlemeyi ve olası hata takibini yapmada işlevseldir.

B) Fonksiyon Tanımlama ve Parametre Alma

- Daha önce gördüğümüz input() ve print() gibi parantezli keyword'ler aslında bir fonksiyonu ifade eder.
- Python'da yüzlerce yerleşik fonksiyon mevcuttur.
- def keywordü ile kendi fonksiyonlarımızı tanımlayabiliriz.
- def fonkAdi(parametre1): komutu ile fonkAdi isimli bir fonksiyon oluştu ve bu fonksiyon bir parametre alır.
- Fonksiyon parametresine gidecek değer parametre ile aynı isme sahip olmak zorunda değildir.
- Fonksiyon parametreleri varsayılan bir değer alabilir. Varsayılan değer, fonksiyon herhangi bir parametreyle almadan çağrılırsa devreye girer.
- Fonksiyonun çağrıldığı yere bir çıktı verebilmesi için bir değeri return() etmelidir.

C) Fonksiyonda Global ve Yerel Değişkenler

- Ana programda tanımlanıp daha sonra her yerde kullanılabilen değişken, global değişken olarak adlandırılır.
- Bir fonksiyon içinde tanımlanan ve sadece burada değişiklik veya düzenlemeye tabi olan değişkene yerel değişken denir.
- Yerel değişken içinde olduğu fonksiyonda global keywordü ile global hale getirilebilir. “global degiskenAdi”
- Bu keyword kullanılırken aynı anda değişkene değer ataması yapılamaz.
- global a = 5 ifadesi bir hatadır.

D) İçeriği Olmayan Fonksiyon

- Bir program hazırlarken bazı fonksiyonları bir prototip olarak bırakmak için bazı keyword’ler kullanılır. Bu sayede program hata vermez.
- pass ve return kullanarak fonksiyonlar bir prototip olarak kalır ve None değeri döndürür.

E) Fonksiyon Kısaltma (Lambda)

- lambda keyword ile tek satırlık fonksiyonlar kolayca oluşturulur.
- fonkAdi = lambda parametre : işlem

F) Özyinelemeli (Rekürsif) Fonksiyon

- Devamlı olarak kendisini çağıran fonksiyondur.
- Bu sebeple bir koşula bağlı olarak yazılmalıdır.

8. STRİNG İŞLEMLERİ

A) String Nedir ?

- Bir veya daha fazla karakterin bir değişkene atanmasıyla oluşan veri tipidir.
- Tek,çift ya da üç (‘ ’, “ ”, “‘ ’’’) tırnak işaretlerinden biri seçilip içine yazılan değer bir stringi ifade eder.
- Tırnak içerisine yazılan matematiksel ifadeler de bir string’ e dönüşür.
- String bir değişkenin değerinin her bir karakteri, bir indise sahiptir ve soldan sağa doğru 0’dan başlayarak artar.
- Her bir boşluğun da bir indisi olur.
- Ters indis ise değerin sağından soluna doğru -1’den başlayarak azalarak ilerleyen indis numaralarıdır.
- degisken=[indisDegeri] ile bir karakterin indis numarası öğrenilir.
- String ifadeleri print içinde değişken adı ve ‘+’ operatörü kullanarak birleştirebiliriz. print(X+Y+Z)
- Bundan farklı olarak bir değişken oluşturup diğer stringleri yine ‘+’ operatörü ile toplayıp yeni değişkene atayabiliriz. A=X+Y+Z
- String ifadeyi parçalara ayırırken degiskenAdi[1:5:2] gibi bir yöntem kullanılır. Bir örneği ele alalım y= “python”
- y[1:4] komutu ekrana yth yazdırır
- y[:5:2] komutu ekrana pto yazdırır.
- y[::-1] komutu ise nohtyp yazdırır.

B) String Güncelleme, Değiştirme ve Silme

- .replace(‘parametre1’, ‘parametre2’) ile değişkenin içindeki veriler yenisiyle değiştirilir ancak güncelleme yapmaz.
- Güncelleme işlemi için ‘=’ operatörü kullanılmalıdır.
- y=y.replace(‘on’, ‘10’). Burada artık ‘on’ yerine ‘10’ yazacak.
- for ve range() ile çeşitli silme işlemleri yapılabilir.

C) String'i Bir Listeye Dönüştürme, Uzunluğunu Bulma ve Karşılaştırma

- .split() ile string içindeki ifadeler bir parametre girilmesiyle o karakterin olduğu yerden bölünür ve bir liste oluşur.
- Varsayılan olarak boşluk parametresi alır.
- Bunun dışında "-, _" gibi karakterler de alabilir.
- len(degiskenAdi) ile değişken uzunluğu bulunur. Boşluk ve özel karakterleri de sayar.
- degiskenAdı.count(parametre) ile parametre yerine yazacağımız karakterin değişkende kaç tane bulunduğunu gösterir.
- Karşılaştırma operatörlerinin (==, !=) yanı sıra is ifadesi de kullanılır.
- in ve not in ile de bir ifadenin başka bir ifadede olup olmadığı karşılaştırılabilir.

D) String İçerisine Değişken Değeri Ataması ve Ters Çevirme

- String değişkeninin değerinin içine {} bırakılır ve ardından degiskenAdi.format(y) ile {} yerine artık y'nin değeri yazılır.
- ' '.join(reversed(degiskenAdi)) ile değişken değeri ters yazılır.
- Tırnak işaretli yere bir karakter girerek değerdeki boşluklar seçtiğimiz karakterle doldurulur.
- Parametre olarak bir string ifade de alabilir.

E) Harf Dönüştürme

» ‘degiskenAdi-------’ ile bazı fonksiyonlar çağrılıp ilgili işlemler yapılır.

.lower()	.upper()	.swapcase()	.capitalize()
• Bütün karakterler küçük harfle yazılır.	• Bütün karakterler büyük harfle yazılır.	• Büyüğü küçük, küçüğü büyük yapar.	• Sadece ilk harf büyük, geriye kalanlar küçük harf olur

» Değer değişir ama güncellenmez. Bunun için aynı değişkene atama yapılmalıdır.

9. DİZİLER VE LİSTELER

A) Dizi (Array)

» Diziler, art arda sıralanmış ve aynı tür veri depolayan bir veri tipidir.
» Sayı ya da karakter dizilerinden bir çeşidini içerir.
» Python’da dizilerin veri tipi bir listeyi gösterir.
» Birçok değişkenin bir arada kullanılmasını pratik bir şekilde sağlar.

B) Liste

» Köşeli parantez içine yazılan sayısal veya sözel verilerin bir arada olabildiği veri tipidir.
» List = [], boş listeyi ifade eder.
» Farklı veri tipleri içerir.
» Liste içindeki her bir elemanın bir indisi vardır

C) Liste Elemanları ile İlgili İşlemler

- **listeAdi [x]:** x. Elemanı yazdırır. x değeri burada da eksi alabilir.
- **listeAdi.index("yusuf"):** yusuf karakterinin listeAdi listesinde kaçıncı indiste olduğunu gösterir.
- String bölmedeki aynı komutlarla dilimlere ayrılabilir.
- **listeAdi.append("emre"):** Listenin sonuna "emre" verisini ekler.
- **listeAdi.insert(1, "emre"):** Listenin 1. indisine "emre" ekler. Sağda kalan elemanların indis numaraları bir artar.
- **len(listeAdi):** Listedeki eleman sayısını verir.
- **listeAdi.count(1):** Listede 1 elemanının kaç tane olduğunu gösterir.
- String birleştirmedeki yöntemler burada da uygulanıp liste birleştirme yapılır.
- **L1.extend(L2):** L1 listesinin sonuna L2 listesini ekler ve L1 güncellenerek yeni bir liste olur. L2 ise aynı kalır.
- **L.reverse():** L listesini ters çevirip oluşan yeni listeyi L listesine atar.
- **min(L) veya max(L):** L listesi içindeki sayısal olarak en büyük veya en küçük değeri gösteren fonksiyonlardır.
- list(ad) fonksiyonu ile string ad değişkeninin değeri bir listeye dönüşür.
- **L.remove("y"):** L listesindeki y'yi siler. Eğer birden fazla y varsa sadece bir tanesini siler. Eğer y yoksa hata verir.
- **L.pop(2):** L listesinin 2. indisteki elemanını siler.
- **L.clear():** L listesindeki tüm verileri siler.
- Daha önce öğrendiğimiz in ve not in komutları ile liste içi arama yapılabilir.
- **L.sort():** Listeyi varsayılan olarak küçükten büyüğe sıralar.

D) enumerate() fonksiyonu

- Genellikle for döngüsü ile kullanılarak liste içindeki veriler numaralandırılır.
- for i, deger in enumerate(listeAdi)

E) Yığın (Stack) ve Kuyruk (Queue)

- Listenin son elemanı üzerinden ekleme ve çıkarma işlemlerini yapıldığı yapıya yığın denir.
- LIFO (Last in First out), son giren ilk çıkar mantığında çalışır.
- Yığın işlemleri için append() ve pop() gibi komutlar kullanılır.
- Listeye ekleme işinin en son elemandan ve çıkarma işinin ilk elemandan gerçekleştiği yapıya kuyruk denir.
- FIFO-LILO (First in First out-Last in Last out), ilk giren ilk çıkar son giren son çıkar mantığında çalışır.
- Kuyruk işlemleri için append() ve pop() fonksiyonları kullanılır.

10. SÖZLÜKLER

A) Sözlük ve Fonksiyonları

- İki veriyi birbiriyle key-value biçiminde ilişkilendiren veri yapısıdır.
- İki farklı yolla oluşturulur.
- Mevsim {“kış”: 1, “ilkbahar”: 2, “yaz”: 3, “sonbahar”: 4 }
- TC = dict([(“talat”, 1234), (“ali”, 1235), (“hasan”, 1236)])
- **sozlukAdi.get(“yusuf”, “yok”)** ile sözlükte “yusuf” adlı key varsa key değerini yoksa “yok” mesajını yazar.
- **sozlukAdi.pop(“keyAdi”)** ile keyAdi sözlükten değeri ile birlikte çıkarılır.
- **sozlukAdi.keys()** ile sözlükteki anahtar kelimeler (key) ekrana yazdırılır.
- **sozlukAdi.values()** ile key değerleri ekrana yazdırılır.
- **sozlukAdi.items()** ile key-value birlikte ekrana yazılır.

B) Küme (Set)

- Süslü parantezlerle tanımlanabilir.
- =set(‘06jho’) ile de bir tanımlama olabilir. set() fonksiyonu burada her bir karakteri küme içine bir veri olarak atar.
- **Y1 | Y2:** Y1 ve Y2 kümeleri birleştirilir. Ortak değer varsa bir kez yazılır.

- **Y1 & Y2:** Ortak elemanları verir.
- **Y1 – Y2:** Y1'de olup Y2'de olmayan elemanları verir.

C) Demet (Tuple)

- Listelere çok benzerdir.
- Farklı olarak parantezlerle () ifade edilir.
- Bir kerelik tanımlanır ve listeler gibi değiştirilemez. Bunun sebebi hafızada çok az yer kaplamasından kaynaklanmaktadır.

11. NESNE VE SINIF

A) Sınıf (Class) Tanımlama ve self-pass İfadeleri

- Bir sınıf (class), kendisinden örnek oluşturulabilen bir nesnenin (object) veri tipidir.
- Sınıflara özellik ve metotlar (sınıfa özgü fonksiyon) atanabilir.
- Sınıflar da fonkisyonlar gibi girintilerle (indentation) başlayıp biter.
- class keyword'ü ile sınıf tanımlaması yapılır.
- class Araba (marka) : *# Parantez zorunlu değil*

```
model = marka
def metot (self):
    self.marka = "bmw"
```

- Sınıf içerisinde tanımlanan fonksiyonların self parametresini alması zorunludur.
- pass ifadesi ile sınıflar da prototip olarak bırakılabilir
- self ise sınıf içerisinde erişilecek elemanlar için kullanılır. Nesnenin kendisini referans etmesini sağlar.

B) Nesne ve Metotlar

- class hayvan ():

```
    pass
Aslan = hayvan () #Hayvan sınıfına ait aslan nesnesi oluşturuldu.
```

- type(Aslan) ile Aslan nesnesinin veri tipi ekrana yazılır.
  <class '__main__.hayvan'>
- Bir nesne üzerinde yapılan değişiklik sınıfı etkilemez.
- nesneAdi.sinifDegiskeni = 'yeniDeger' ile nesnenin bir değişkeninin değeri değiştirilir.
- Nesne metotları sınıflardan farklı olarak nesneAdi.metotAdi() ile çağrılır.
- dir.(nesneAdi) ile nesnenin sahip olduğu metot ve özellikleri öğrenebiliriz.

C) init Metotu

- İlk değer ataması için kullanılan özel bir fonksiyondur.
- return bir değer döndüremez.
- def __init__(self, prm1, prm2)
  self.prm1 = prm1
  self.prm2 = prm2

D) Kalıtım (İnheritence)

- Sınıf içinde sınıf tanımlanarak üst sınıfın özellikleri ve metotları alt sınıfa miras olarak aktarılır.
- class altSinif (ustSinif):
- class Araba():
  def __init__(self, model, fiyat)
  self.model = model
  self.fiyat = fiyat

  class Kamyon(Araba): *# Kamyon Arabadan Miras Aldı*

  def __init__(self, model, fiyat, renk):

  araba.__init__(self, model, fiyat, renk)

  self.renk=renk

E) Modül Kavramı

- import'la daha önce yapılmış olan programı dosya adını da kullanarak kendi programlarımızda kullanmamıza yarayan komuttur
- Oluşturulan her Python dosyası aslında bir modüldür.
- from dosyaAdi import * : dosyaAdi modülü içindeki tüm verilere erişim hakkı tanır.
- from dosyaAdi import fonkAdi : Sadece fonkAdi fonksiyonuna erişim izni verir. Bu sayede program daha hızlı çalışır.
- import math, random yazılarak tek satırda import yapılabilir.
- import yapıldıktan sonra dosyaAdi.fonkAdi(,) ile de erişim sağlanabilir.
- import dosyaAdi as dA : dosyaAdi kısaltılıp dA oldu.
- Bir dosyanın ana program tarafından mı yoksa import edilen bir modülden çalıştırıldığını öğrenmek için "__name__ == '__main__' " özelliğinden faydalanılır.
- Ekran çıktısı 'True' ise ana program 'False' ise bir modülü işaret eder.

12. RASTGELE SAYI ÜRETİMİ

A) Matematiksel İşlem Fonksiyonları

abs(a) • Mutlak değer	round(a.b) • En yakın tam sayı	max(a,b) • Sayıların en büyüğü	min(a,b) • Sayıların en küçüğü
pow(a,b) • a üssü b = a^b	bin(a) • İkili sayı sistemi	hex(a) • On altılık sayı sistemi	chr(a) • ASCII değeri
	sum( ) • Girilen sayıların toplamı	gcd(a,b) • import math • Sayıların EBOB'u	

B) random.------ Komutları

.random() • 0 ile 1 arası float bir değer	.randint(a,b) • a ile b arasında int bir değer	.randrange(a,b,c) • a'dan b'ye c kadar artan aralıktaki sayılardan bir değer
.uniform(a,b) • a ile b arasında float bir değer	.choice([]) • Girilen listeden rastgele bir değer seçilir.	.shuffle(L1) • L1 listesinin elemanlarını rastgele sıralar.
	.sample(L1, a) • L1 listesinden a kadar elemanı seçer.	

13. TURTLE İLE ÇİZİM UYGULAMALARI

A) Turtle

- Turtle, kullanıcıların sanal bir tuval oluşturarak resim ve şekiller oluşturmasına olanak tanıyan, önceden yüklenmiş bir Python modülüdür.
- Çizim için kullandığınız ekran kaleminin adı Turtle'dır ve kütüphaneye adını veren de budur.
- Turtle kartezyen koordinat sistemini kullanır (x,y). Koordinat sisteminin ölçü birimi ise pixel olarak adlandırılır.
- **turtle.position()** ya da kısaltması olan **turtle.pos()** komutları ile pozisyon öğrenilir.
- **turtle.goto(100,100)**: Ekranda x = +100 ve y = +100 konumuna gidilmiştir.

- **turtle.setposition(-50,-50)**: Ekranda x = -50 ve y = -50 konumuna gidilmiştir. Kısaltma olarak setpos da yazılabilir. ‘goto’ ile aynı görevdelerdir.
- **turtle.reset()** ya da **turtle.home()**: Başlangıç noktası olan (0,0) konumuna dönülür.
- **turtle.setx(200)** ve **turtle.sety(200)**: Sadece x ya da y ekseninde konum değişikliği yapmak için kullanılır (200,200).

B) Hareket, Simgeler ve Çizimler

- turtle----- ile ileri-geri, sağ-sol hareketleri yapılır. Bunlar:
- **.forward(100)** = **.fd(100)**: 100 piksel ileri
- **.backward(100)** = **.bk(100)**: 100 piksel geri
- **.right(90)** = **.rt(90)**: 90 derece sağa dön
- **.left(90)** = **.lt(90)**: 90 derece sola dön
- Turtle’da imlecin hangi şekillerde olabileceğini **print(turtle.getshapes())** ile ekrana yazdırabiliriz.
- Ekrana gelen şekillerden biri seçilip imleç **turtle.shape(‘sekiladi’)** komutuyla değiştirilir.
- **turtle.hideturtle()**: Hareket sonrası imleci görünmez yapmak için kullanılır.
- **turtle.show()**: Gizlenen imleci tekrardan görünür kılmak için kullanılır.
- **pensize(5)**: Kenar çizgilerinin kalınlığını ayarlar.
- **color(‘yellow’, ‘green’)**: ‘yellow’ argümanı kenar çizgi rengini ifade eder. ‘green’ ise oluşan cismin iç dolgu rengini ifade eder.
- İç dolgu rengini oluşturmak için fonksiyon, **begin_fill()** ve **end_fill()** arasına yazılır. ‘begin’ ile işlem başlar ve ‘end’ ile sonlanır.
- Farklı renk tonları yakalamak için RGB (red,green,blue) renk kodlarından yararlanılır.
- Her bir RGB kod 0 ile 255 arasında değer alır.
- RGB renk kodlarını uygulamak için **colormode(255)** fonksiyonu çalıştırılmalıdır.
- **circle(100)**: Yarıçapı 100 piksel olan bir daire çizer. Çizime saat yönünün tersinde başlar. Değer negatif girilseydi bu kez saat yönünde çizerdi.

- **penup()** ve **goto(x,y)** ile çizim yapılmadan istenilen yere hareket edilir.
- **pendown()** ile imlece tekrardan çizim özelliği kazandırılır.
- **circle(100,360,5)**: komutu ile bir beşgen oluşur. Son kısımdaki değeri değiştirerek birçok çokgen kolayca oluşturulabilir.

C) Veri Girişi için İletişim Kurma

- **numinput('poligon', 'kenar sayısı', 5)** : Kullanıcıya bir mesaj kutusu gönderir ve sayısal bir değer girişi ister. Kutunun üstünde poligon yazar. Veri girişinin gerçekleşeceği yerde ise kenar sayısı yazar. Veri girişinin default değeri ise 5 olarak belirlenmiştir. Bu ifade aslında float bir değer döndürmesine karşın komutun başına int() eklenerek sorun çözülür.
- **textinput('', '')** komutunda ise metinsel bir veri kullanıcıdan istenir. İlk parametre pencere başlığını, ikinci parametre ise veri girişi için kullanıcıya mesajı ifade eder.

D) Pencereye Resim Ekleme

- Pencereye eklenecek resim mutlaka .gif uzantılı olmalıdır.
- **setup(512,512)** ile resim çerçevesi eklenir. İlk parametre genişliği son parametre ise yüksekliği ifade eder.
- **bgpic("resim.gif")** ile resim dosyası programa dahil edilir.
- Oluşan pencereye başlık eklemek için title("pencereBaslik") komutu kullanılabilir.
- Çizim ortamının kaybolmaması için **mainloop()** fonksiyonu kullanılır.

14. OLAY (EVENT) TABANLI PROJELER

A) Komutlar

onclick(f) - onscreenclick(f)

- Fareye tıklandığında f fonksiyonu çağrılır.

onrelease(f)

- Fare butonu bırakıldığında f fonksiyonu çağrılır.

ondrag(f)

- Fare sürüklenmesi olduğunda f fonksiyonu çağrılır.

listen()

- Tuş olaylarını yakalamak için turtle ekranı dinlenir.

onkey(f,tus) - onkeyrelease(f,tus)

- İlgili tuş bırakıldığında f fonksiyonu çağrılır.

onkeypress(f,tus)

- İlgili tuşa basıldığında f fonksiyonu çağrılır.

ontimer(f,t)

- t milisaniye sonra f fonksiyonunu çağırarak sayaç kurar.

mainloop() - done()

- Olay yakalama döngüsünü başlatır. Son satırda yer almalıdır.

exitonclick()

- Fareye tıklandığında bye() fonksiyonunu çağırır. O da pencereyi kapatır.

write()

- Ekrana yazı yazar. Yazının konumu ve fontu da ayarlanabilir.

delay()

- Milisaniye cinsinden çizim hareketlerinin gecikmesini sağlar.

tracer(n,delay)

- Animasyon ayarlarını değiştirir. Animasyonsuz için tracer(0) kullanılır.

update()

- Çizim ekranını günceller. tracer() komutundan sonra mutlaka kullanılmalıdır.

distance()

- Nesnenin konumu ile bilgileri verilen noktanın arasındaki mesafedir.

B) Projeye Ses Ekleme

- Ses dosyasının uzantısı .mp3 ya da .waw olmalıdır.
- Playsound modülü import edilmelidir.
- playsound.playsound(“dosyaAdi.mp3”) ile ilgili yerlere çağrılır.

BİLGİ TEKNOLOJİLERİ VE İLETİŞİM KURUMU

BTK AKADEMİ

Sertifika No: Yx1h8P7nkm

KATILIM SERTİFİKASI

YUSUF EMRE ÖZÜGÜZEL

"Yeni Başlayanlar için Python Programlama"
çevrim içi eğitimini tamamlayarak
"30.07.2025"
tarihinde bu sertifikayı almaya hak kazanmıştır.

Mustafa ERMİŞ

İnsan Kaynakları ve Eğitim Dairesi Başkanı
Bilgi Teknolojileri ve İletişim Kurumu