

Paralel Hesaplama

Öğr. Gör. Zafer SERİN

PYTHON NEDİR?

- Python, yüksek seviyeli, genel amaçlı bir programlama dilidir. Guido van Rossum tarafından 1991 yılında geliştirilmeye başlanan Python, basit ve okunabilir bir sözdizimine sahiptir.



PYTHON NEDİR?

- Python, nesne yönelimli programlama(OOP) özelliklerini destekler. Platform bağımsız bir programlama dilidir. Açık kaynaklıdır. Topluluk desteği çok yüksektir.



PYTHON NEDİR?

- Python, 1, 2 ve 3 sürümleri ile ön plana çıkar günümüzde 1 ve 2 sürümüne verilen destek sonlanmıştır; ancak halen bazı uygulamalarda kullanılabilmektedir. Özellikle 3 sürümü günümüzde kullanılır.



PYTHON NEDİR?

- Python, **yorumlanan bir dildir**. Bu, Python kodlarının doğrudan makine diline derlenmeden, bir yorumlayıcı tarafından satır satır çalıştırıldığı anlamına gelir. Python kodları, çalıştırılmadan önce ara bir adımda **bytecode**'a derlenir.

DERLENEN VE YORUMLANAN DİLLER

Derleyici(Compiler) Kullanan Diller

Kod, çalıştırılmadan önce makine diline derlenir.

Genellikle daha hızlı çalışır çünkü kod doğrudan makine diline derlenir.

Genellikle hata ayıklama daha zordur.

Derlenmiş kod, genellikle okunamaz hale gelir. Bu, kod gizliliği sağlar.

C, C++, Java gibi diller örnek olarak verilebilir.

Yorumlayıcı(Interpreter) Kullanan Diller

Kod, satır satır yorumlayıcı tarafından çalıştırılır.

Genellikle daha yavaş çalışır çünkü kod satır satır yorumlanır.

Genellikle hata ayıklama daha kolaydır.

Yorumlanan kod, genellikle kaynak kodu olarak kullanılır. Bu, kod gizliliğini zorlaştırır.

Python, JavaScript, Ruby gibi diller örnek olarak verilebilir.

YORUM SATIRI

- Python'da # işareti ile tek satırlık, ''' ''' ile çok satırlık yorum satırı oluşturulabilir.

```
# Tek satırlık yorum  
'''  
Çok satırlık  
yorum  
'''
```

DEĞİŞKEN TANIMLAMA

- Python'da 'degiskenAdi = deger' şeklinde doğrudan değişken tanımlaması yapılabilir. Kodların sonuna herhangi bir ';' koyulmaz!

```
degisken1 = 11 # tamsayı tanımlama  
degisken2 = 5.6 # ondalikli sayi tanımlama  
degisken3 = True # mantiksal tanımlama  
degisken4 = "Selam" # metinsel tanımlama
```


DEĞİŞKEN TANIMLAMA STANDARTLARI

- Genelde değişkene isim verilirken 3 temel yaklaşım kullanılır. Bunlar camelCase, snake_case ve PascalCase olarak kullanılır.

```
birinciDegiskeninDegeri = 11 # camelCase  
BirinciDegiskeninDegeri = 11 # PascalCase  
birinci_degiskenin_degeri = 11 # snake_case
```

DEĞİŞKEN İSİMLENDİRME KURALLARI

- ‘_’ harici özel karakterler değişken isminde kullanılamaz.
- Değişken adi sayı ile başlayamaz.
- Python’a özgü isimler değişken adı olarak kullanılamaz.
- Python büyük küçük harf duyarlı bir dildir.

DEĞİŞKEN TİPİ ÖĞRENME

- Python'da 'type(degisken_adi)' kodu ile ilgili değişkenin tipi öğrenilebilir.

```
plaka = 11  
print(type(plaka))
```

<class 'int'>

String İşlemleri

- Python'da bir string ifade içerisindeki karakterlere indisleri ile ulaşılabilir. Bu indisleme aşağıda gösterildiği gibi olur.

Z	a	f	e	r
0	1	2	3	4
-5	-4	-3	-2	-1

String İşlemleri

- Python'da bir string ifade içerisindeki karakterlere indisleri ile ulaşılabilir. Aşağıda gösterildiği gibi bu indislere ulaşılabilir.

```
isim = "Zafer"  
print(isim[0])  
print(isim[-3])
```

String Bölme

- Python'da bir string ifadeyi bölmek için şu ifade kullanılabilir: 'degisken_adi[baslangic_indisi:bitis_indisi:artim_miktari]'. Burada bitiş indisinin ifadeye dahil olmayacağı unutulmamalıdır!

```
isim = "Zafer"  
print(isim[1:3:1])
```

af

```
isim = "Bilecik Şeyh Edebali Üniversitesi"  
print(isim[5:22:3])
```

işhda

String Bölme

- Python'da bir string ifadeyi bölmek için şu ifade kullanılabilir: 'degisken_adi[baslangic_indisi:bitis_indisi:artim_miktari]'. Burada bitiş indisinin ifadeye dahil olmayacağı unutulmamalıdır!

```
isim = "Bilecik Şeyh Edebali Üniversitesi"  
print(isim[-15:-4:2])
```

İnvri

```
isim = "Bilecik Şeyh Edebali Üniversitesi"  
print(isim[::-1])
```

isetisrevinÜ ilabedE hyeŞ kiceliB

String Uzunluğu

- len() kodu ile bir stringin uzunluğu elde edilebilir.

```
isim = "Bilecik Şeyh Edebali Üniversitesi"  
print(len(isim))
```


Önemli String Metotları

- `upper()`: Tüm stringi büyük harfe çevirir.
- `lower()`: Tüm stringi küçük harfe çevirir.
- `title()`: String ifadedeki kelimelerin ilk harflerini büyük yapar.
- `capitalize()`: String ifadenin sadece ilk harfini büyük harf yapar.
- `islower()`: Stringin tamamı küçük harfse true değilse false döner.
- `isupper()`: Stringin tamamı büyük harfse true değilse false döner.
- `strip()`: Stringin başındaki ve sonundaki boşlukları siler.
- `split()`: Verilen parametreden stringi böler.
- `index()`: Verilen parametrenin string ifadedeki indisini döner.
- `startswith()`: String ifade verilen parametre ile başlıyorsa true değilse false döner.
- `endswith()`: string ifade verilen parametre ile bitiyorsa true değilse false döner.

Python Operatörleri

+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Mod alma
**	Üs alma

Not: + operatörü stringlerde birleştirme yapar

math Modülü

- Python'da math modülü ile matematiksel işlemler yapılabilir

```
import math  
print(math.factorial(5))
```

LİSTELER

- Listeler sıralı elemanlar dizisidir. Python'da [] ile veya list() constructorı ile liste tanımlanabilir.

```
liste = ["deneme", False, 2.5, 5]
```

```
liste1 = list(["deneme", False, 2.5, 5])
```

- Liste elemanlarına stringlerde olduğu gibi indisleri ile erişilebilir. İndisler 0'dan başlar ve son elemana kadar(listenin uzunluğunun 1 eksiği) gider. Elemanlara tersten erişmek içinde -1 indisi ile başlanarak bu sefer sola doğru -2, -3 ... şeklinde ulaşılabilir.

LİSTELER

- Örnek:

```
liste = ["deneme", False, 2.5, 5]  
print(liste[1])  
print(liste[-1])
```

False

5

LİSTELER

- Listelerin elemanları değiştirilebilir(mutable). Stringlerin elemanları değiştirilemez!(immutable)

```
liste = ["deneme", False, 2.5, 5]  
liste[0] = "test"
```

Çalışır

```
isim = "Zafer"  
isim[1] = "e"
```

Çalışmaz!

ÖNEMLİ LİSTE METOTLARI

- `append()`: listenin sonuna eleman ekler
- `clear()`: listeyi tamamen boş hale getirir
- `copy()`: listeyi kopyalar
- `count()`: listedeki bir elemanın sayısını verir
- `extend()`: listeyi başka bir eleman ile genişletir
- `index()`: listedeki bir elemanın indisini verir
- `insert()`: listenin belirli bir indisine eleman eklemeyi sağlar
- `pop()`: varsayılan olarak liste sonundan eğer verilirse indisten değeri siler
- `remove()`: verilen değeri listeden siler