

YAPAY ZEKA VE PYTHON PROGRAMLAMA

Temel İşlemler
Emir Öztürk

Doç. Dr. Altan MESUT'un "Programlama Dillerine Giriş" ders slaytları kaynak alınmıştır.

DEĞİŞKENLER

- Veri saklamak
- Veriler üzerinde işlem yapmak
- Tür belirlenmesi

Programlarda kullanılan verileri saklamak ve gerektiğinde o veriler üzerinde işlem yapmak için değişkenler kullanılır.

Birçok programlama dilinde bir değişken tanımı yapıldığında, o değişkenin türüne göre istenilen büyüklükte bir hafıza bölgesi işletim sisteminden istenir. Tanımlanan değişkene bir değer atandığında, değişken için ayrılan hafıza bölgesine o değer yazılır.

Basic ve Python gibi bazı dillerde ise değişken tanımı yapılmadan da o değişken kullanılabilir. Python'da bir değişkene ilk defa değer atandığında o değer türüne göre değişkenin türü belirlenir.

DEĞİŞKEN TÜRLERİ VE DEĞİŞKENE DEĞER ATAMA

- `x = 3` (Tamsayı)
- `x = 3.0` (Ondalıkli sayı)
- `x = "Deneme"` (Karakter dizisi, string)
- Tür dönüşümü desteği
- Runtime hata ihtimali

Yukarıda görüldüğü gibi Python'da bir değişkene farklı türde bir değer atanmak istendiğinde hata vermez. Bu durum programcıya kullanım kolaylığı sağlasa da, bazı durumlarda çalışma zamanı (run-time) hatalarına veya tutarsız sonuçlara neden olabilir. Python bu yönü ile weakly typed bir dil olarak sınıflandırılır. C, C++, JAVA gibi diller ise bu gibi durumlarda hata verip derlemedikleri için daha güvenli olarak görülürler ve bu nedenle strongly typed olarak sınıflandırılırlar.

DEĞER ATAMA KURALLARI

- a=3, b=5, a=b
- a=b
- b=a
- a,b = 1,2
- a,b = b,a
- a=a+1

Değişken ismi solda, atanacak değer sağda olmalıdır (7 = a doğru değildir):

a = b a değişkenine b değişkenindeki değeri atar

b = a b değişkenine a değişkenindeki değeri atar

Aynı anda iki yada daha fazla değişkene değer atanabilir.

a, b = 4, 5 a değişkenine 4, b değişkenine 5 atar

a, b = b, a a'ya b'nin değerini, b'ye a'nın değerini atar

a = a+1 bir sonraki adımda a artık a+1 değeri alacak anlamına gelmektedir.

DEĞİŞKEN İSİMLENDİRME KURALLARI

- Özel karakterler olmaması(_ hariç)
- İlk karakterin sayı olmaması
- Aynı isimde birden çok değişken olmaması
- Ayrılmış kelimelerin kullanılmaması
- Büyük küçük harf ayrımı

Değişken isminde : ; , . / ' # [] ! " \$ % ^ & * () { } karakterleri ve boşluk karakteri kullanılamaz. Ancak _ kullanılabilir.

Değişken isminin ilk karakteri harf olmalıdır. Yani değişkenler rakamla başlamaz (sayi2 olur 2sayi olmaz).

Bir fonksiyon veya prosedürde aynı değişken ismi birçok defa tanımlanamaz (farklı fonksiyonlarda tanımlanabilir).

Programlama diline ait bir komut ismi, değişken ismi olarak tanımlanamaz.

SQL ve VB.NET gibi bazı dillerde değişken isimlerinde büyük/küçük harf ayrımı yoktur (case insensitive). Abc, abc veya ABC aynı değişkeni ifade eder.

Python, C, C++, C# ve JAVA gibi dillerinde büyük/küçük harf ayrımı vardır (case sensitive).

OPERATÖRLER

- Aritmetik
 - +, -, *, /, %, //, **
- Mantıksal
 - and, or, not
- Karşılaştırma
 - <, >, <=, >=, ==, !=
- Atama
 - =, +=, -=, *=, /=, %=, //=, **=

Python dilinde kullanılan belirli operatörler verilmiştir. Aritmetik operatörler temel aritmetik işlemlerin yanında tam bölme, mod ve üs alma işlemlerini destekler. Mantıksal operatörler bool değerler üzerinde kullanılırlar. Karşılaştırma operatörleri büyüklük, küçüklük, eşitlik ve farklılık durumlarının kontrolü için kullanılırlar. Atama operatörleri değerin değiştirilip tekrar aynı değişken üzerine atılması için kısayol sağlarlar.

VERİ TÜRLERİ

- int
- float
- str
- type()

Python'da temel olarak 3 veri türü vardır:

int: tamsayıları saklar (integer)

float: ondalıklı sayıların saklanması içindir

str: Bir veya daha çok karakterin saklanmasını sağlar. Tek tırnak veya çift tırnak içinde yazılan her şey rakam bile olsa str (string) olarak saklanır.

Bir değişkenin veya veri yapısının türünü öğrenmek için type() fonksiyonu kullanılabilir.

type(a) yazdığınızda <class 'int'> gibi bir çıktı alırsınız

VERİ TÜRÜ DÖNÜŞÜMÜ

- fonksiyonlar
 - int()
 - float()
 - str()

Veri türleri ile aynı isimde olan ve farklı türdeki verileri o türe dönüştüren fonksiyonlar vardır. Bu fonksiyonlara başka veri türünde bir değişken verildiğinde tür dönüşümü işlemi gerçekleştirirler.

ARİTMETİK İŞLEMLER

- > İşlem önceliği
- > **
- > * (çarpma) / (bölme) // (tam bölme)
- > + -
- > string * (string çoklama)

Aritmetik işlemlerde işlem önceliği bulunmaktadır.
Parantez içi ve üs alma işlemleri en başta gerçekleştirilir.
Çarpma ve bölme bundan sonra yapılır.
String'ler üzerinde + ve * operatörleri kullanılabilir.

EKRANDAN GİRDİ ALINMASI

- > input() fonksiyonu
- > Enter'a basılana kadar
- > string

input() fonksiyonu ENTER basılana kadar girilen karakter dizisini (string) döndürür. Döndürdüğü değeri bir değişkene atanabilir
Kullanıcının girdiği değerin hepsi rakam olsa bile str türünde ad değişkenine atanır

EKRANA YAZDIRMA

- > print() fonksiyonu
- > string, int veya float olabilir.
- > Birden fazla değerin gösterilmesi için ","
- > sep, end

Ekrana veri yazdırma için print kullanılır. string, int veya float değerler print ile ekrana yazdırılabilirler. sep özelliği ile birden fazla yazılan değer arasında ayraç belirlenir, end özelliği ile satır sonunda ne yazılması gerektiği verilebilir.

STRING BİRLEŖTİRME

- + operatörü
- iki string
- string + sayı
- sayı + string

•Eğer print içerisindeki tüm değerler str türünde ise ',' yerine '+' kullanılabilir. Fakat '+' aralara boşluk karakteri eklemes.

print('python',3,'öğreniyorum') için çıktı “python 3 öğreniyorum” olurken

print('python'+ '3'+ 'öğreniyorum') için çıktı “python3öğreniyorum” olacaktır.

ÖRNEK UYGULAMALAR

- Ekrandan girilen x değeri için $5x^2 + 7x - 9$ değerini hesaplayan uygulama
- Ekrandan girilen x değeri için $x + x * x + x * x * x$ değerini ekrana gösteren uygulama
- Ekrandan girilen x değerinin y değerine göre bölümünden kalanı ekrana gösteren uygulama