



IEEE

Python3 Programming

IEEE Student Branch Python3 Programming Course



Programlamaya Genel Bakış

- Programlama kompleks ve/veya tekrara dayalı işleri gerçekleştirmek için kullanılır.
- Herkes programlama dilinin temelini öğrenebilir ancak önemli olan öğrenilen parçaları bir araya getirerek bir sorunun çözümünde kullanılabilmektir.
- Programlama ile bir sorun çözülmeden önce sorun mutlaka parçalara ayrılmalıdır.

Neden Python

- Başlangıç için ideal/Acemi dostu
- Anlaşılması kolay
- Çok esnek
- Topluluk: GitHub'da en çok tercih edilen 4.dil ve en büyük StackOverFlow Topluluğuna sahip.
- Kariyer Fırsatı: Angel List'te Python en çok talep edilen 2.yetenek ve ayrıca ortalama en yüksek maaşlı yetenek.
- TIOBE'a göre en çok tercih edilen 4.dil

Dezavantajları Neler?

- Yavaş: Python yorumlama(interpreter) tabanlı ve esnek bir dil olduğu için kodun işletilmesinde çok fazla makineye başvuru yapılır bu da yavaşlamasına neden olur.
- Bakımı Kolay Değildir: Dinamik bir dil olduğu için aynı kod, içeriğe göre farkı anlam kazanabilir. Özellikle büyük projelerde anlam karmaşası oluşturabilmektedir.

Nerelerde Kullanılır?

- Makine Öğrenmesi
- Yapay Zeka
- Veri Analizi
- Ağ Programcılığı (mail sunucusu vb.)
- Web Programcılığı (Google, Dropbox, Instagram vb.)
- Bilimsel
- Sistem Yönetimi (Gömülü Sistemler dahil)

Interpreter

- Python yorumlama tabanlı bir dildir.
- Python dilinde yazılmış bir kodu çalıştırabilmek için Python yorumlayıcısına (Python Interpreter) sahip olmak gerekir.
- Interpreter kodu doğru çalıştırabilmesi için kod sürümü ile interpreter sürümü uyumlu olmalıdır.

Inveractive Shell

- Etkileşimli Kabuk, bizim Python programlama dili ile ilişki kurabileceğimiz, yani onunla etkileşebileceğimiz bir üst katmandır.
- Komut Satırı da denebilir.
- Anlık olarak kodları deneyebileceğiniz veya herhangi bir .py dosyasını Run(Çalıştır) edebileceğiniz köprü görevi üstlenen yapıdır.

IDE(Integrated Development Environment) Nedir?

- Programcıların hızlı ve rahat bir şekilde program geliştirmesini sağlayan araçları bir arada toplayan yazılım geliştirme platformu.
- Syntax Renklendirme
- Interpreter, Debugger, Compiler gibi pek çok aracı içinde bulunduran platform
- Visual Studio, Code::Blocks, NetBeans, xCode vb.



Python Shell ve PyCharm Kurulumu



Değişkenler

- Programlamada verilerin konulduğu bellek alanlarına değişken denir.
- Diğer programlama dillerinin aksine Python'da değişkenleri önceden tanımlamak gerekmez.
- Ayrıca Python dinamik bir dil olduğu için değişkenlerin türleri sonradan değiştirilebilir.
- Bir değişkene değer atamak için = (atama) operatörünü kullanırız.
- Global değişken ve yerel değişken kavramı

Address	Content	Name	Type	Value
90000000	00	iii	int	000000FF (255 ₁₀)
90000001	00			
90000002	00			
90000003	FF			
90000004	FF	sss	short	FFFF (-1 ₁₀)
90000005	FF			
90000006	1F	ddd	double	1FFFFFFFFFFFFFFFFF (4.4501477170144023E-308 ₁₀)
90000007	FF			
90000008	FF			
90000009	FF			
9000000A	FF			
9000000B	FF			
9000000C	FF			
9000000D	FF			
9000000E	90	ptr	int*	90000000
9000000F	00			
90000010	00			
90000011	00			

Note: All numbers in hexadecimal

Değişken İsimlerinin Özellikleri

- Programcı değişken isimlerini seçmekte özgür olmakla birlikte, rakam ile başlayamaz ve alt çizgi haricinde özel karakter kullanamaz.
- Ayrıca Türkçe karakterlerin kullanılamaz.
- Değişkenleri kullanımına uygun isimlendirmek iyi programlama önerisidir.
- Değişken isimleri küçük-büyük harf duyarlılığına sahiptir.
- Ayrılmış anahtar kelimeler değişken ismi olarak kullanılamaz.

Aritmetik Operatörler

Operatör	Operatör Karakteri	Örnek	Sonuç
Toplama	+	4+4	8
Çıkarma	-	4-4	0
Çarpma	*	4*4	16
Bölme	/	19/4	4.75
Tam Sayı Bölme	//	19//4	4
Mod	%	4%2	0
Kuvvet Alma	**	4**2	16

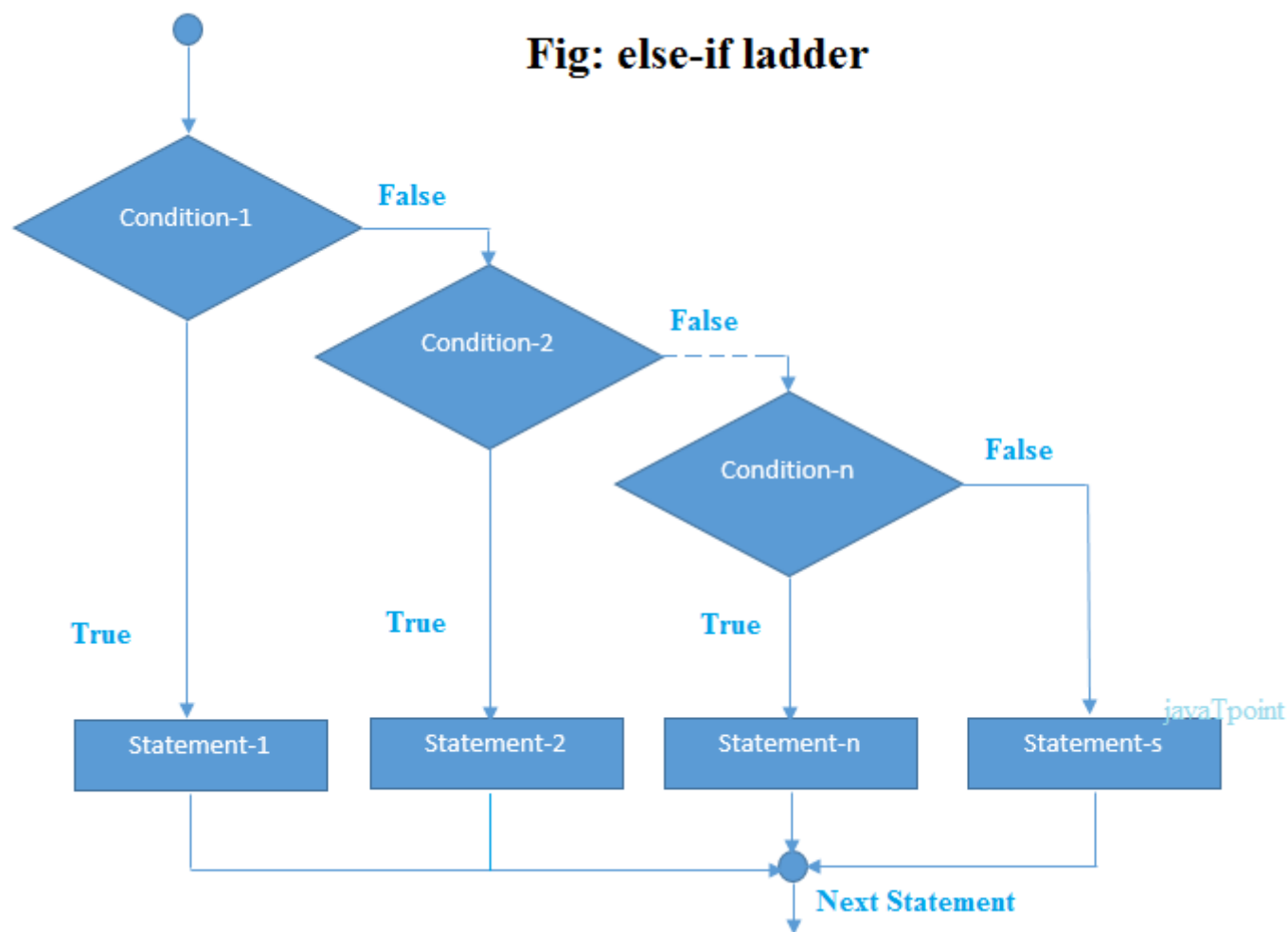
Lojik (Mantıksal) Operatörler

Operatör	Anlamı
if	Eğer koşulu, Bir veya birden fazla koşulun doğruluğunu karşılaştırmak ve doğru olduğu durumda yorumlanacak komutları belirlemek için kullanılır.
else	Değilse koşulu, if koşulundan sonra eğer if koşulu sağlanmıyorsa işletilecek komutları belirlemek için kullanılır
elif	Değilse eğer koşulu, if koşulu yanlış ise tekrar bir koşul doğruluğunu kontrol etmek ve doğru olduğu durumda yorumlanacak komutları belirler

Lojik (Mantıksal) Operatörler

Operatör	Anlamı
and	İki veya daha fazla koşulu ve bağlacı ile birbirine bağlar
or	İki veya daha fazla koşulu veya bağlacı ile birbirine bağlar
not	Değil operatörü, bir koşulun doğruluğunu tersine çevirir.

Fig: else-if ladder



Döngü Yapıları

- Bir kod bloğu içinde birden fazla tekrar etmesi gereken işlemlerde döngü bloklarını kullanabiliriz.
- İki tane döngü yapısı vardır: for ve while
- for döngüsü diğer kodlama dillerinden biraz farklıdır.
- Genellikle bir dizi veya liste içerisinde iteratör bir değişken ile birlikte kullanılır.
- while döngüsü yine bir koşula bağlı olarak kullanılabilir.

Döngü Yapıları

- range() fonksiyonu for yapıları için en önemli fonksiyonlardan biridir.
- Genellikle bir değişke range ile oluşturulan bir dizi içerisinde gezinerek döngü sağlar.
- Üç tarzda kullanılabilir:

range(stop)

range(start, stop)

range(start, stop, step)