

Python 101

Ders 4:

+Nesne Tabanlı
Programlamaya Giriş

Eğitmen : Şevket Aşurgan
İletişim:sevketasurgan@gmail.com

Nesne Tabanlı Programlama Nedir ?

Nesne Tabanlı Programlama, birçok modern programlama dilinde kullanılan bir paradigmaya denir. Bu paradigmada, programlar nesnelerin kullanımına dayanır. Nesne, bir şeyin özelliklerini ve işlevlerini taşıyan bir birimdir. Nesne tabanlı programlama, programlamacıların daha büyük ve karmaşık projeleri yönetmelerine yardımcı olan modüler ve yapılandırılmış bir yaklaşım sunar.

Python, Nesne Tabanlı Programlama konusunda oldukça güçlü bir dil olarak kabul edilir. Python'da her şey bir nesne olarak ele alınır. Bu, temel veri tiplerinin bile bir nesne olarak ele alındığı anlamına gelir. Python'da bir sınıf, bir nesnenin özelliklerini ve işlevlerini tanımlar. Sınıflar, bir şablon olarak düşünülebilir ve nesnelerin oluşturulması için kullanılır.



Nesne Tabanlı Programlama'nın Temel Konseptleri

Sınıflar: Sınıflar, nesnelerin özelliklerini ve işlevlerini tanımlayan şablonlardır. Bir sınıf, bir nesnenin nasıl oluşturulacağını ve hangi özelliklere sahip olacağını belirtir. Sınıflar, özellikler (attributes) ve işlevler (methods) içerebilir.

Nesneler: Nesneler, bir sınıfın örnekleridir. Bir sınıftan birden çok nesne oluşturulabilir ve her nesne ayrı bir özelliklere ve işlevlere sahip olabilir.

Kalıtım: Kalıtım, bir sınıfın başka bir sınıftan özelliklerini ve işlevlerini devralmasıdır. Bu, bir sınıfın bir başka sınıfın tüm özelliklerine ve işlevlerine sahip olmasını sağlar ve kodun tekrar kullanılmasını kolaylaştırır.

Polimorfizm: Polimorfizm, farklı nesnelerin aynı işlevleri kullanarak farklı sonuçlar üretmesini sağlayan bir konsepttir. Bu, kodun daha esnek ve modüler olmasını sağlar.

Veri Kapsülleme: Veri Kapsülleme, bir nesnenin verilerinin korunmasını ve doğrudan değiştirilememesini sağlayan bir konsepttir. Bu, nesnelerin daha güvenli ve düzenli olmasını sağlar.

Sınıflar

Python, Nesne Tabanlı Programlama konusunda oldukça güçlü bir dil olarak kabul edilir. Python'da her şey bir nesne olarak ele alınır. Bu, temel veri tiplerinin bile bir nesne olarak ele alındığı anlamına gelir. Python'da bir sınıf, bir nesnenin özelliklerini ve işlevlerini tanımlar. Sınıflar, bir şablon olarak düşünülebilir ve nesnelerin oluşturulması için kullanılır.

Sınıflar

Python'da sınıflar, class anahtar kelimesi kullanılarak tanımlanır. Bir sınıfın temel yapısı şöyle görünür:

```
class SınıfAdi:  
    # Sınıf özellikleri ve işlevleri burada yer alır  
    pass
```

Sınıflar

Sınıfın adı büyük harfle başlamalı ve CamelCase şeklinde yazılmalıdır. Sınıfın içindeki özellikler ve işlevler pass kelimesi yerine tanımlanır. Aşağıdaki örnekte bir sınıfın nasıl tanımlanacağı görülebilir:

```
class Kedi:  
    def miyavla(self):  
        print("Miyav!")
```

Sınıflardan Nesne Oluşturma

Bir sınıfın nesnesi oluşturmak için, sınıfın adını kullanarak bir değişken tanımlanır ve parantez içinde sınıfın adını yazılır.

```
benim_kedim = Kedi()
```

Sınıf Özellikleri

Bir sınıfın özellikleri, sınıfın değişkenleri olarak tanımlanır. Bu özelliklere, sınıfın herhangi bir işlevinde erişilebilir. Özelliklerin tanımlanması, sınıfın içinde gerçekleştirilir. Aşağıdaki örnekte, Kedi sınıfına renk adında bir özellik eklendi:

```
class Kedi:  
    renk = "Beyaz"  
    def miyavla(self):  
        print("Miyav!") ``
```


Sınıf İşlevleri

Sınıf işlevleri, sınıfın işlevleri olarak tanımlanır ve def anahtar kelimesi kullanılarak tanımlanır. Sınıf işlevleri, nesne tarafından çağrılabilir ve self kelimesi ile nesneye erişilebilir.

```
class Kedi:  
    renk = "Beyaz"  
  
    def miyavla(self):  
        print("Miyav!")  
    def beslen(self, yemek):  
        print("Kedi", yemek, "yiyor.")
```

Sınıf Yapıcı Metodu(Constructor)

Sınıfın yapıcı metodunu,» `__init__` «olarak adlandırılır ve nesne oluşturulurken otomatik olarak çağrılır. Yapıcı metodu kullanarak nesne özellikleri tanımlanabilir.

Örnek Kullanım

```
class Kedi:  
    def __init__(self, renk):  
        self.renk = renk  
  
    def miyavla(self):  
        print("Miyav!")  
  
    def beslen(self, yemek):  
        print("Kedi", yemek, "yiyor.")
```

Sınıf Kalıtımı

Python'da sınıf kalıtımı, bir sınıfın başka bir sınıftan özelliklerini devralmasıdır. Kalıtım, daha büyük ve karmaşık projeleri yönetmek için oldukça kullanışlı bir özelliktir.

Örnek Kullanım

```
class EvKedisi(Kedi):  
    def uyku(self):  
        print("Kedi uyuyor.")
```

Polimorfizm

Polimorfizm, nesne tabanlı programlama konseptlerinden biridir ve farklı nesnelerin aynı işlevleri farklı şekilde uygulamasına izin verir.

Fonksiyon Polimorfizmi

Python, fonksiyon polimorfizmi konseptini destekleyen bir nesne tabanlı programlama dilidir. Polimorfizm, farklı nesnelerin aynı işlevleri farklı şekillerde uygulamasına olanak tanır ve programlamada kod tekrarını azaltır. Fonksiyon polimorfizmi, aynı işlevin farklı parametrelerle, veri tipleriyle ve diğer fonksiyonlarla kullanılabilmesini sağlar.

Örnek Kullanım

```
print(str(42)) # Çıktı: '42'
```

```
print(int('42')) # Çıktı: 42
```


sevketasurgan@gmail.com
