

Temel Python Eğitimi

Eğitim Hakkında

Temel Python Eğitimi, programlamaya giriş yapmak isteyen katılımcılar için Python dili üzerinden sistematik ve uygulamalı bir başlangıç sunar. Eğitim, yazılım mantığını öğrenmek isteyen yeni başlayanlardan, veri analizi, yapay zekâ ve otomasyon projelerinde Python'a ihtiyaç duyan teknik profesyonellere kadar geniş bir katılımcı kitlesine hitap eder.

Katılımcılar; değişkenler, veri tipleri, kontrol yapıları, fonksiyonlar, nesne yönelimli programlama, dosya işlemleri ve veritabanı bağlantısı gibi konuları adım adım öğrenecek, eğitim sonunda orta seviye Python projelerini tek başına geliştirebilecek bilgi ve yetkinliğe ulaşacaktır.

Ön Koşullar

- Programlama bilgisi **gerekmemektedir**.
 - Temel bilgisayar kullanımı ve algoritma mantığına ilgi duymak yeterlidir.
 - Kodlama disiplinine açık olmak ve düzenli pratik yapmak öğrenme sürecini hızlandırır.
-

Eğitim Süresi

Toplam Süre: 5 Gün (Günde 6 saat)

Toplam Eğitim Saati: 30 saat

Eğitim Formatı: Uygulamalı ve interaktif sınıf/online ortam

Kimler Katılmalı?

- Programlama dünyasına giriş yapmak isteyen herkes
 - Yazılım geliştirme temellerini Python üzerinden öğrenmek isteyen öğrenciler
 - Veri bilimi, yapay zekâ ve makine öğrenmesi gibi alanlara yönelmek isteyenler
 - Test, otomasyon, veritabanı işlemleri gibi alanlarda Python kullanmak isteyen teknik personel
 - Python diline ihtiyaç duyan tüm mühendislik, istatistik veya iş analitiği profesyonelleri
-

Eğitim İçeriği

- **Python Giriş**
 - Python programlama dilini tanıma
 - Python ve IDE Kurulumu
 - Output işlemleri
- **Data Types - Veri Türleri ve Bilgisayarın Çalışma Mantığı**

- NoneType ve Numeric (Sayısal) Veri Tipleri
 - Dizi Halindeki Veri Tipleri
 - Listeler (List)
 - Demetler (Tuple)
 - Setler (Set)
 - Sözlükler (Dictionary)
- **Operators - Operatörler**
 - Arithmetic Operators (Aritmetik Operatörler)
 - Assignment Operators (Atama operatörleri)
 - Comparison Operators (Karşılaştırma operatörleri)
 - Logical Operators (Mantıksal operatörler)
 - Operatörlerde KISA DEVRE (Associativity (Birleşme Özelliği))
 - Unary Operators (Tek terimli operatör)
 - Identity Operators (Benzerlik Operatörleri)
 - Bitwise Operators (Bitsel operatörler)
 - Membership Operators (Üyelik Operatörleri)
 - Ternary Conditional Operators - Üçlü Karşılaştırma Operatörleri
 - Number System Conversion (Sayı Sistemi Dönüşümü)
- **Math Modülü – Veri Alma işlemleri - DEBUG**
 - Math modülü incelemesi
 - Input işlemleri
 - Hata ayıklama
- **Flow Control - Akış Kontrolü**
 - If Clauses - If deyimi
 - Nested If Clauses – İç içe If deyimi
 - For Loop - For döngüsü
 - While Loop – While döngüsü
 - Break, Continue, Pass Statements - Break, Continue, Pass İfadeleri
 - Range Function - Range İfadesi
 - ASCII T
- **Functions - Fonksiyonlar**
 - Fonksiyonlara Giriş
 - Dönüş değeri olmayan parametresiz (argümanlı) fonksiyonlar
 - Dönüş değeri olmayan parametrelili (argümanlı) fonksiyonlar
 - Dönüş değeri olan parametresiz (argümanlı) fonksiyonlar
 - Dönüş değeri olan parametrelili (argümanlı) fonksiyonlar
 - Birden fazla dönüş değeri olan fonksiyonlar ve paketten çıkarma işlemi
 - Pass by value (değer geçişi) - Pass by reference (adres geçişi)
 - Types of Arguments (Argümanların tipleri)
 - Variable-Length Arguments (Sınırsız sayıda değer alma işlemleri)
 - Local Variable (Yerel değişken) vs Global Variable (Global değişken)

- Recursive ve Iterative Fonksiyonlar
 - Lambda Anonymous Function (İsmi olmayan fonksiyon)
 - Özel Fonksiyonlar (map, filter, reduce, zip, enumerate, all, any fonksiyonu)
 - Decorators (Dekoratörler diğer bir adıyla Süsleyiciler)
- **Modules - Modüller**
 - Modüllerle Çalışma
 - Kendi modülümüzü oluşturma ve kullanma
 - Special Variable (Özel değişken) olan **name** değişkeninin işlevi
- **Object Oriented Programming – Nesneye Yönelik Programlama**
 - Built-in Types (Tanımlı Tipler) (Python'da varsayılan veri tipleri)
 - Fonksiyon objeleriyle çalışma
 - Kendi class'ımızı (sınıfımızı - veri tipimizi) oluşturma
 - Special Variables ve Special Methods (Özel değişkenler ve metotlar)
 - Class kullanılarak oluşturulan objelerin özelliklerinin özelleştiri
 - Objelerin bilgilerinin güncellenmesi ve karşılaştırılması
 - Instance vs Class (Static) Variables
 - Instance vs Class vs Static Methods
 - Inner Class (class içinde class)
 - Inheritance (Kalıtım)
 - MRO – Method Resolution Order
 - Constructor
 - Polymorphism (Çok biçimlilik)
 - Duck Typing (Ördek Yazımı)
 - Operator Overloading (Operatörün aşırı yüklenmesi)
 - Magic Methods (Büyülü Metotlar)
 - Method Overloading (Metotun aşırı yüklenmesi)
 - Method Overriding (Metotun geçersiz kılınması)
 - Abstract Classes (Soyut Sınıflar) ve Abstract Methods (Soyut Metotlar)
- **Iterators - Generators**
 - Iterators(İteratörler - Yineleyici) Kavramı
 - Generators (Üreteçler)
 - Kendi fonksiyonumuzu modüle ekleme
- **Exception Handling**

- Exception Handling (İstisnayı idare etme)
 - Exception bilgisi görüntüleme ve finally anahtar kelimesi
 - Multi Exception Handling (Çoklu Olağandışılığı İdare Etme)
 - Programımıza aykırı durumları Handle Etme (Logical Error)
 - Duruma göre kendi Exception'ımızı (olağandışılığımızı) fırlatma
 - **Dosya İşlemleri**
 - Dosya oluşturma ve yazma işlemleri
 - Dosya okuma işlemleri
 - Dosyalarda kullanılan fonksiyonlar
 - Dosyalarda değişiklik yapma
 - **SQLite Veritabanı**
 - SQLite Veri tabanı ve tablo oluşturma
 - Tablolara veri ekleme
 - Tablodaki verileri çekme
 - Tablodaki verileri silme ve güncelleme
-

Eğitim Sonunda Kazanacaklarınız

- Python programlama dilinde yetkinlik kazanma
 - Temel algoritmalar ve kontrol yapıları kurma becerisi
 - Fonksiyonel ve nesne yönelimli programlama pratiği
 - Hata yönetimi, dosya işlemleri ve veritabanı bağlantısı kurabilme
 - Veri işleme, otomasyon ve analiz gibi ileri konulara sağlam bir temel
 - Kendi başına projeler geliştirebilecek teknik özgüven
-