

Uygulama Geliştirme Yaşam Döngüsü, bir uygulamanın geliştirilmesi için izlenmesi gereken adımları tanımlar. Bu döngü, yedi aşamadan oluşur:

### Gereksinim Toplama

- Kullanıcı, iş ve teknik gereksinimlerin toplanması.
- Örnek: Otel rezervasyon uygulaması için kullanıcıların odaları görüntüleyebilmesi gibi gereksinimler.

### Analiz ve Tasarım

- Toplanan gereksinimlerin analiz edilmesi ve tasarımın oluşturulması.
- Tasarım sürecinde belgelerin güncellenmesi ve doğrulanması önemlidir.

### Kodlama ve Test

- Tasarım belgelerine dayanarak uygulamanın kodlanması ve test edilmesi.
- Birim testi, uygulamanın her bir parçasının gereksinimleri karşıladığı doğrular.

### Kullanıcı ve Sistem Testi

- Kullanıcıların uygulamanın işlevsellliğini test etmesi ve sistem entegrasyon testlerinin yapılması.
- Performans testi, uygulamanın hızını ve ölçülebilirliğini değerlendirir.

### Üretim

- Uygulamanın son kullanıcılar tarafından erişilebilir hale getirilmesi.
- Uygulamanın kararlı bir durumda tutulması gerekmektedir.

### Bakım

- Uygulamanın güncellenmesi veya yeni özelliklerin eklenmesi gerektiğinde, bu yeni özelliklerin tüm aşamalardan geçmesi gereklidir.

Bu aşamalar, uygulama geliştirme sürecinin düzenli ve etkili bir şekilde yürütülmesini sağlar.

Python kodu yazarken okunabilirliği artırmak için bazı önemli kurallar ve standartlar bulunmaktadır. Bu içerik, Python Style Guide ve Coding Practices üzerine odaklanmaktadır.

### PEP-8 Kılavuzları

- Dört boşluk kullanarak girinti yapın; bu, kodun tutarlığını artırır.
- Fonksiyonlar ve sınıflar arasında boş satırlar kullanarak kodun bölümlerini ayırin.

### Kod Okunabilirliğini Artırma

- Operatörler ve virgüllerden sonra boşluklar kullanarak komutların okunabilirliğini artırın.
- Daha büyük kod blokları için ayrı fonksiyonlar oluşturun; bu, kodun yeniden kullanılabilirliğini ve yönetilebilirliğini artırır.

## Kodlama Konvansiyonları

- Fonksiyon ve dosya isimlerini küçük harflerle ve alt çizgi ile yazın.
- Sınıf isimlerini CamelCase ile yazın; bu, sınıfları ve fonksiyonları ayırt etmeye yardımcı olur.
- Sabit isimlerini büyük harflerle ve kelimeleri alt çizgi ile ayırarak yazın.

## Statik Kod Analizi

- Statik kod analizi, kodu çalıştırmadan belirli standartlara göre değerlendirmek için kullanılır. Örneğin, PyLint kütüphanesi PEP-8 uyumluluğunu kontrol etmek için kullanılabilir.

## Unit Testing Nedir?

- Unit Testing, kodun test edilebilir küçük parçalarını (unit) doğrulamak için kullanılır.
- Örnek olarak, bir mymodule.py dosyasında square ve doubler fonksiyonları bulunmaktadır.

## Unit Test Süreci

- Kod geliştirme sırasında her bir unit, iki aşamada test edilir: yerel sistemde ve sunucu ortamında (CICD test sunucusu).
- Test başarısız olursa, hata nedenleri belirlenir ve düzelttilir.

## Unit Test Dosyası Oluşturma

- unittest Python kütüphanesi içe aktarılır.
- Test edilecek fonksiyonlar içe aktarılır.
- Test sınıfı oluşturulur ve unittest.TestCase sınıfından miras alınır.
- Her fonksiyon için test fonksiyonları oluşturulur ve test ile başlamalıdır.

## Test Durumları ve Sonuçları

- Test durumları oluşturulurken, assertEquals gibi assertion yöntemleri kullanılır.
- Test sonuçları, testin geçip geçmediğini gösterir; başarısız olursa hata detayları sağlanır.

## Modüller

- Python modülü, Python tanımlamaları, ifadeleri, fonksiyonları ve sınıflarını içeren bir .py dosyasıdır. Modüller, diğer betiklere ve not defterlerine aktarılabilir.
- Örneğin, bir modül olan module.py dosyasında iki fonksiyon tanımlanmıştır: square ve doubler.

## Paketler

- Paket, bir dizin içinde bulunan Python modüllerinin bir koleksiyonudur ve bu dizin, onu sadece bir Python betiği dizininden ayıran bir `init.py` dosyasına sahiptir.
- Örneğin, `my_project` adlı bir paket, `module1.py` ve `module2.py` adlı iki modül içerir ve bir `init.py` dosyası bulundurur.

## Kütüphaneler

- Kütüphane, bir veya daha fazla paketten oluşan bir koleksiyondur. NumPy, PyTorch ve Pandas gibi örnekler, genellikle paket olarak da adlandırılır.
- Paket ve kütüphane terimleri sıkılıkla birbirinin yerine kullanılır.