

# İleri Seviye Flutter Eğitimi

## Eğitim Hakkında

Digital Vizyon Akademi'nin İleri Seviye Flutter eğitimi, temel ve sağlam Flutter uygulamalarını başarılı bir şekilde oluşturmak için gereken uygulamalı programlama becerilerini sağlar.

Dart programlama dilini kullanmayı, Flutter'da hata ayıklamayı, özel widget'lar oluşturmayı, bir ekran düzenini ve hareketlere yanıt vermeyi amaçlar.

Neler Öğreneceksiniz

Tüm katılımcılar aşağıdakilerin nasıl yapıldığını öğreneceklerdir:

- Dünyadaki 5 milyar iOS / Android cep telefonunun herhangi birinde ve ayrıca tarayıcı ve masaüstü ortamlarında çalışacak bir çapraz platform uygulaması yazmayı
- Flutter uygulamaları geliştirmeyi ve hata ayıklamayı
- Flutter uygulamalarında Dart programlama dilinin zarafetinden yararlanmayı
- Temaları ve stilleri uygulamayı
- Özel widget'lar yazmayı
- Dokunma, kaydırma gibi hareketlere yanıt vermeyi
- Uygulamaların düzenini duyarlı bir şekilde hassas bir şekilde kontrol etmeyi
- Kullanıcılardan form veri girişini işlemeyi
- Gezinme, menüler ve sekmelerle çoklu ekran uygulamaları yapmayı
- Çevrimiçi bir RESTful API'den veri okumak ve yazmak için Flutter'ı kullanmayı
- 3. taraf kitaplıkları bulmayı ve dahil etmeyi

Eğitim İçeriği

### Giriş

#### Hello Flutter

Flutter nedir?

Neden Flutter?

Diğer seçenekler

Yerel çözümler

## **Dart Diline Genel Bakış**

Dart nedir?

Beklenen özellikler - Dart Cheatsheet

Veri türleri, Diziler / listeler

Sınıflar

Koşullar ve döngüler

Dart hakkında beklenmedik şeyler

Çıkarım türü

final ve sabit

\$ ile dize enterpolasyonu

Yayımlama operatörü

Harita <foo, bar>

Fonksiyonlar nesnelerdir

Büyük ok / Şişman ok

Adlandırılmış işlev parametreleri

"Yeni" ve "bu" atlanıyor.

Sınıf yapıcı parametresi kısaltması

Özel sınıf üyeleri

Mixins

Kaskad operatörü (..)

Aşırı yükleme yok

Adlandırılmış yapıcılar

## **Flutter'da Geliştirme**

Flutter araç zinciri

Flutter SDK

IDE'ler

IDE Geliştirme Araçları

Emülatörler

Araçları güncel tutmak

Flutter geliştirme süreci

Uygulamayı ve dosyaları iskele

Uygulamanızı çalıştırma

## **Her Şey Widget'tır**

Kod olarak kullanıcı arayüzü

Yerleşik Flutter widget'ları

Değer widget'ları

Düzen widget'ları

Gezinme widget'ları

Diğer aletler

Durum bilgisi olmayan pencere öğeleri nasıl oluşturulur

Widget'ların anahtarları vardır

Widget'ınıza bir değer iletme

Durum Bilgisi İçermeyen ve Durum Bilgili pencere öğeleri

Peki hangisini yaratmalıyım?

## **Değer Widget'ları**

Metin widget'ı

Simge widget'ı  
Görüntü widget'ı  
Gömülü görüntüler  
Ağ görüntüleri  
Bir görüntüyü boyutlandırma  
Giriş widget'ları  
Metin alanları  
Form widget'larını bir araya getirme  
Form widget'ı  
FormField widget'ı  
Büyük bir Form örneği

### **Hareketlere Yanıt Verme**

Meet the button family  
RaisedButton  
FlatButton ve IconButton  
FloatingActionButton  
CupertinoButton  
Reddeditilebilir  
Özel widget'larınız için özel hareketler  
Uzun basışa tepki  
Yeni bir öğe eklemek için sıkıştırma  
Sola veya sağa kaydırma  
Hareket alanı

### **Widget'larınızı Yerleştirme**

Tüm sahneyi düzenlemek  
MaterialApp widget'ı  
The Scaffold widget  
AppBar widget'ı  
SafeArea widget'ı  
SnackBar widget'ı  
Flutter bir widget'ın boyutuna nasıl karar verir?  
The dreaded “unbounded height” error  
Flutter'ın düzen algoritması  
Widget'ları diğerlerinin yanına veya altına yerleştirmek  
Widget'larınız asla sığmayacak!  
Ya fazladan boşluk varsa?  
mainAxisAlignment  
crossAxisAlignment  
Genişletilmiş widget  
Ya yeterli alan yoksa?  
ListView widget'ı  
Kapsayıcı gereci ve kutu modeli  
Bir Kap içinde hizalama ve konumlandırma  
Peki bir Konteynerin boyutunu nasıl belirlersiniz?  
Özel düzen widget'ları  
Yığın widget'ı  
GridView widget'ı

Tablo widget'ı

### **Gezinme ve Yönlendirme**

Yığın gezinme

İleri ve geri gitme

Bir sahne kapandıktan sonra sonuç alın

Çekmece de gezinme

Çekmece widget'ı

Çekmecenin doldurulması

Sekmede Gezinme

TabController

TabBar ve Sekmeler

İletişim kutusu widget'ı

showDialog () ve AlertDialog

İletişim Kutulu Yanıtlar

Gezinme yöntemleri birleştirilebilir

### **Pencere Öğelerini Stilize Etme**

Flutter Tarzlarında Düşünmek

Renkler hakkında bir kelime

Metnin Stilini Ayarlama

Yazı stili

Özel yazı tipleri

Konteyner dekorasyonları

Kenarlık

BorderRadius

BoxShape

Widget'ları istifleme

Konumlandırılmış widget

Kart gereci

Temalar

Tema özelliklerini uygulama

### **Yönetim Devleti**

Devlet nedir?

StatefulWidget'ta neler var?

Devletle ilgili en önemli kural!

Belirtilen geçme

Durum yedeklemesini kaldırma

Devlet yönetimine bir örnek

Devleti ne zaman kullanmalıyız?

Gelişmiş durum yönetimi

DevralınanWidget

BLoC

ScopedModel

Kancalar

Sağlayıcı

Redux

### **Flutter Uygulamanız Dosyalarla Çalışabilir**

Flutter uygulamanıza kitaplıklar dahil etme  
Bir kütüphane bulmak  
Pubspec.yaml dosyasına ekleniyor  
Kitaplığı içe aktarma  
Kitaplığı kullanma  
Futures, async ve wait  
Why would it wait?  
Await  
Async  
Uygulamanıza bir dosya eklemek  
Bir dosya yazmak  
Ve okuyorum!  
JSON kullanma  
Uygulamanızın belleğini JSON'a yazma  
JSON'u belleğe okuma  
Paylaşılan Tercihler  
Tercihleri yazmak için  
Tercihleri okumak için

### **HTTP ile RESTful API Çağrılarını Yapma**

API isteklerinin çeşitleri  
HTTP GET veya DELETE isteği yapmak  
HTTP PUT, POST veya PATCH isteğinde bulunma  
Widget'lara HTTP yanıtları  
Brute force - The easy way  
FutureBuilder - Temiz yol  
Kesin yazılmış sınıflar  
Bir işletme sınıfı oluşturun  
FromJSON () yöntemi yazın  
Nesneyi hidratlamak için fromJSON () kullanın  
Büyük bir örnek  
Flutter'da bir GET isteği  
Flutter'da bir DELETE isteği  
Flutter'da bir POST ve PUT isteği

### **Firestore'i Flutter ile kullanma (izin verilen süre)**

Firestore ile tanışın  
Cloud Firestore  
Bulut İşlevleri  
Doğrulama  
Firestore'in kendisini kurma  
Firestore projesi oluşturma  
Veritabanının oluşturulması  
Bir iOS uygulaması oluşturma  
Bir Android uygulaması oluşturma  
FlutterFire eklentileri ekleme  
Firestore'u kullanma  
Bir koleksiyon almak için  
Sorgulamak

Yükseltmek için  
Silme  
Buradan nereye gidilir