

<Yeni Başlayanlar İçin/>



GENÇ DESTEK: 61

ARAŞTIRMA: 5

**ZAFER DEMİRKOL / YENİ BAŞLAYANLAR İÇİN KODLAMA**

Her hakkı saklıdır. Bu eserin aynen ya da özet olarak hiçbir bölümü, yayınevinin yazılı izni alınmadan kullanılamaz.

*İmtiyaz Sahibi:* Yelda Cumalıoğlu

*Genel Yayın Yönetmeni:* Özlem Esmergül

*Editör:* Devrim Yalkut *Kapak Tasarım:* İlknur Muştu *Sayfa Düzeni:* Cansu Poroy

*Sosyal Medya-Grafik:* Tuğçe Budak-Mesud Topal

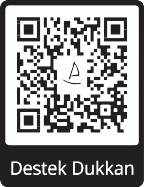
Genç Destek: Eylül 2021 Yayıncı Sertifika No. 13226

ISBN 978-625-441-280-6

© Genç Destek

Abdi İpekçi Caddesi No. 31/5 Nişantaşı/İstanbull Tel. (0) 212 252 22 42

Faks: (0) 212 252 22 43

[www.destekyayinlari.com](http://www.destekyayinlari.com/) [info@destekyayinlari.com](mailto:info@destekyayinlari.com) **facebook**.com/DestekYayinevi **twitter**.com/destekyayinlari **instagram**.com/destekyayinlari

Deniz Ofset – Çetin Koçak Sertifika No. 48625 Maltepe Mahallesi

Hastane Yolu Sokak No. 1/6 Zeytinburnu / İstanbul

Genç Destek, Destek Yayınları’nın tescilli markasıdır.



**genç DESTEK**





<Yeni Başlayanlar İçin/>

**İ Ç İ N D E K İ L E R**

**ÖNSÖZ** 11

**EDİTÖRLER** 13

IDE (Integrated Development Environment) 14

Kitaptaki Örnekleri Yapabileceğiniz Editörleri Tanıyalım 15

Replit 15

Try.dot.net 17

Google Colab 17

Visual Studio Code 18

**PROGRAMLAMA** 23

Her Dil Programlama Dili midir? 26

Değişkenler 27

Değişken Oluşturma ve Kullanma 28

Değişken Tipleri 30

C# Değişken Tipleri 33

C# Çalışma Ortamı 33

C#’da Değişkenlerin Tanımlanması 35

Koleksiyonlar: Diziler, Listeler, Sözlükler ve Diğerleri 39

Python Listelerinde Belirli Elemanları Seçmek 40

Python Listelerinde Eleman Eklemek, Çıkarmak 42

Yorumlanan ve Derlenen Diller 44

C#’da Dizi ve Listeler 45

C#’da Liste İşlemleri 47

C# Dizilerinde Belirli Elemanları Seçmek 50

Tuple 52

C# ve Tuple Yapısı 54

Dictionary 56

C# ve Dictionary Yapısı 58

SET 60

C# ve Set Yapısı 62

String’lerle Çalışmak 63

C#’da Stringler ile Çalışmak 65

**KOŞUL YAPILARI** 67

if Deyimi ile Program Akışını Kontrol Etmek 68

C#’da Koşul İfadesi 70

Birden Fazla Koşul Tanımlamak 71

C#’da Birden Fazla Koşul İfadesi 73

Else İfadesi 73

Yuvalanmış (Nested) İfadeler 74

Switch Yapısı 76

Mantıksal (Lojik) İfade ve Operatörlerin

Koşul Yapılarında Kullanılması 77

**DÖNGÜLER** 81

“For-in” Döngüsü 82

“While” Döngüsü 85

“else” Deyimi 87

“break” deyimi 88

C#’da döngüler 89

Algoritmalar 92

**HATALAR VE TANIMLI İSTİSNALAR** 93

Yazım Hataları 94

İstisnalar veya Mantık Hataları 97

Tanımlı Hatalar 99

C# İstisna, Hata Yönetimi 101

**PROGRAMLARI BÖLÜMLERE AYIRMAK –**

**FONKSİYONLAR/METOTLAR** 103

Fonksiyonlar 104

Python’da Fonksiyon Oluşturmak 104

Fonksiyon Parametreleri 106

Kullanıcıdan Veri Almak ve İşlemek –

Python Fonksiyon Uygulaması 108

Fonksiyonların Faydası 111

C# Metotlar 112

Tanımlı Gelen Fonksiyon ve Metotlar 116

**MODÜL VE PAKETLER** 117

Modül Oluşturmak 117

Modülleri Kullanmak 120

“from” ile Modülleri Kullanmak 123

Fonksiyonlar Dışında Modüllerde Tanımlanabilen

Diğer Programsal İfadeler 124

Paketler 126

Paket Oluşturmak 127

Paketlerin Kullanılması 130

init .py Dosyaları 132

Paketlerin Organizasyonu 133

**SINIF VE NESNE KAVRAMI – NESNEYE YÖNELİK PROGRAMLAMA** 135

Nesneler ve Sınıflar 135

Python’da Sınıf Oluşturmak 137

Nesnelerin İlk Değerleri 140

Sınıf Özelliği Tanımlamak 142

Metotlara Yakından Bakış 143

Nesneye Yönelik Programlamanın 4 Temel Prensibi 146

Soyutlama (Abstraction) 147

Kapsülleme (Encapsulation) 147

Miras (Inheritance) 149

Çok Biçimlilik (Polymorphism) 151

C# Sınıf ve Nesne Kavramı 152

C# Metot Tanımı 156

“Constructor” Kavramı 158

Aynı İsimde Farklı Metotlar – Çok Biçimlilik 159

C#’da Miras 160

**KONSOL, TERMİNAL, KOMUT SATIRI, KOMUT İSTEMCİSİ** 165

Windows’ta Konsol, Komut İstemcisi, Terminal 166

Konsolun Farklı Kullanımları – Python ile Çalışmak 169

PiP (Package installer for Python) Python Paket Yükleme 172

Browser’larda Konsol 173

**VERSİYON KONTROLÜ/Git - GitHub** 175

Git Versiyon Kontrol Altyapısı 176

“Git-GitHub” Yapısı 176

Git-GitHub ile Çalışmak 178

Değişikliklerin GitHub’a Eklenmesi 183

Güncellenmiş Dosyalarla Çalışmak 186

Önceki Sürümlere Dönmek 187

**UYGULAMA PLATFORMLARI** 189

Masaüstü Uygulamalar 190

Windows Masaüstü Uygulaması Geliştirmek 190

Mobil Uygulama Geliştirmek 198

Basit Bir Android Mobil Uygulaması Geliştirelim 199

Web Uygulamaları 209

Bir Web Uygulaması Geliştirelim 211

Veri Kaynakları, Veritabanları ile Çalışmak 216

MS Access Veritabanı ile Python Uygulaması 217

SQLite Veritabanı 220

Python ve SQLite Uygulaması 220

SQLite Verilerine Python Uygulamasından Erişmek 224

Verileri Güncellemek 225

Web Uygulamalarında Veritabanı Kullanmak 226

Rest Web Servisleri 229

Web Servisini Python’da Kullanma 231

Veri Bilimi, Makine Öğrenimi, Yapay Zekâ Çalışmaları 232

Python ile Makine Öğrenimi 235

Python ve Pandas 237

Python ve Matplotlib 239

Python ile Makine Öğrenimi Çalışması 241

Modelin Dosya Olarak Elde Edilmesi 244

# ÖNSÖZ

Bu kitabı programlama bilmeyen ama öğrenmek isteyenler için yazdım. Programlamanın temel mantığını, yapı ve kavramlarını açıklamaya odaklandım.

Kitapta programlama dili olarak, tamamen farklı yapılarda olan, buna karşın çok yaygın kullanılan Python ve C# dillerini seçtim. Örneklerimi bu iki dile ait kodlarla yaptım. Hedefim okuyucuya, iki farklı dünyanın sözdizimini (syntax) gösterirken, bu farklılıklara rağmen, programlamaya ait ortak kavramları da an- latabilmekti.

Programlamanın temel kavramlarını, işleyiş ve yapılarını bilen birisi, dilleri de kolayca öğrenmeye hazırdır. Daha sonraki süreçte yapılması gereken o dillerin yazım kurallarını öğrenmektir. Bu da referans kaynaklara dayanan bir çalışmadır. Kitabı bitirdiğinizde, Python ve C# dillerini de temel düzeyde öğrenmiş ola-

caksınız.

Bunun yanında günümüz programcılığında olmazsa olmaz iki temel konu da kitabın bölümlerini oluşturuyor. Bunlardan ilki Konsol-Terminal kullanımı, diğeri de versiyon kontrolü için Git-GitHub platformu. Bu iki önemli konu da kitapta örneklerle, temel seviyede anlatılmıştır.

Kitabın son bölümünde programlamanın yaygın olarak kullanıldığı, günü- müzde çok popüler olan platformlarında, temel uygulamalar geliştirilmiştir.

Bu uygulamalar şunlardır:

*Zafer Demirkol // Yeni Başlayanlar İçin Kodlama*

* Masaüstü
* Mobil
* Web
* Veritabanı
* Rest Web Servisleri
* Veri Bilimi
* Makine Öğrenimi

Böylece okuyucu, edindiği temel bilgileri, farklı platformlarda uygulayabil- me imkânına kavuşuyor. Ayrıca bu popüler platform ve uygulamaların çalışma mantığını da görmüş oluyor.

Kitapta ayrıca, programlamada kullanılan temel araç ve servisleri de tanıttım. Bu kitabı bitirdiğinizde, programlamanın mantığını ve kavramlarını anlamış, popüler uygulama, platform, araç ve servislerini temel düzeyde deneyimlemiş ve

programlama dünyasına büyük bir adım atmış olacaksınız.

*Zafer Demirkol Kitabın Web Sayfası:* [www.zaferdemirkol.com/programlama](http://www.zaferdemirkol.com/programlama)

# EDİTÖRLER

Programları yazdığımız ortamlara, araçlara “editörler” denir. Programlamanın ilk dönemlerinde bunlar notepad gibi basit metin editörleriydi. Ancak kodlar karmaşıklaştıkça bu basit editörler yetersiz kaldı.

Kodları yazmamızı kolaylaştıracak araçlar, olası bir hatada yardımcı olacak yöntemler sunmuyorlardı.

Zaman içinde programlama dillerine özel kod yazmak için “editör” program- ları geliştirildi. Bu editörler sadece bir dile odaklı olabileceği gibi, aynı anda pek çok dilde program yazmamızı kolaylaştıracak araçlara sahiplerdir.

Editörleri, programlama dillerine özel ifadeleri farklı renklerde göstererek, standart ifadeleri tamamlayarak, hata oluştuğunda hatanın oluştuğu bölümü göstererek pek çok yardımcı araç ve yöntemler sunarlar. Yazdığımız kodların sonucunu görüp, düzenlemeler yapmamızı daha kolay hale getirirler.

Pek çok editör bulunmaktadır, bunların bir kısmı yerel olarak, yani bilgisaya- ra kurulum yapılarak çalıştırılabilirken, bir kısmı da online bulut servisi olarak hizmet verirler.

Günümüzde artık bulutta çalışan editörler kullanmak çok daha etkin ve ma- liyetsizdir.

Bir servis olarak bulut editörü seçmemizin pek çok avantajı vardır.

* Kolay paylaşım, ortak çalışma imkânı
* Her zaman güncel sürüm ve özelliklerden faydalanma
* Bulut servisin işlem gücünden faydalanma
* Kurulum yapma zorunluluğu olmaması

*Zafer Demirkol // Yeni Başlayanlar İçin Kodlama*

Bu ve bunun gibi pek çok faydası bulunur.

Öte yandan eğer isterseniz bir editörü bilgisayarınıza kurup internete bağlı olmadan doğrudan da kullanabilirsiniz. Tercih size kalmış.

İster bulut editör kullanın isterseniz yerel, her iki kullanım için neredeyse sayısız seçenek mevcuttur.

Buradaki seçim tamamıyla kişisel alışkanlıklar, algılar, renkler vs. gibi pek çok faktöre bağlıdır.

Benim en çok tercih ettiğim yerel çalışan editör “Visual Studio Code”dur. Bununla birlikte daha pek çok harika kod editörü mevcuttur. Bazıları ücretli,

bazıları ücretsizdir.

Bunlardan bir kısmı şunlardır:

* Pycharm
* Sublime Text
* Notepad ++

*Visual Studio Code* çok sayıdaki eklentisi ile pek çok dil için yardımcı araçlar ve ortamlar sunmaktadır ve bu eklentiler her geçen gün artar. En güzel yanı da ücretsiz olmasıdır.

Bulut servis olarak kullanılabilecek pek çok editör veya hizmet vardır:

* Colab (Google Colaboratory)
* Codesandbox
* CodePen
* Replit
* try.dot.net

## IDE (Integrated Development Environment)

Editörlerin haricinde *IDE* (Integrated Development Environment) denilen ge- liştirme ortamları vardır. IDE herhangi bir platform için gereken bütün araçları sunan entegre geliştirme ortamlarıdır. Örneğin Mac işletim sisteminde çalışan

*Zafer Demirkol // Yeni Başlayanlar İçin Kodlama*

*XCode* uygulama “IDE”si, *iPhone* telefon simülatöründen, *iPad* altyapısına kadar her türlü yazılımsal yapıyı ve araçları sunar. Veya Windows Ortamı’nda *Visual Studio* (code değil) IDE’si pek çok platform için uygulama geliştirmemizi sağla- yan altyapı ve araçları sunar.

Programcı bu bütünleşik ortamda, o platforma ait bütün araçları bulabilir ve kullanabilir.

IDE daha çok belirli bir alana yönelik uygulamalar için kullanılır.

Kitabımızın **Uygulama Platformları** bölümünde bu editörlerle uygulamalar geliştireceğiz.

## Kitaptaki Örnekleri Yapabileceğiniz Editörleri Tanıyalım

Ben kitap boyunca 4 farklı editör kullanacağım.

C# kodları için:

* Replit
* try.dot.net

Online editörleri ile uygulamalar yapacağım. Python için de iki farklı editör kullanacağım:

* Colab
* Visual Studio Code

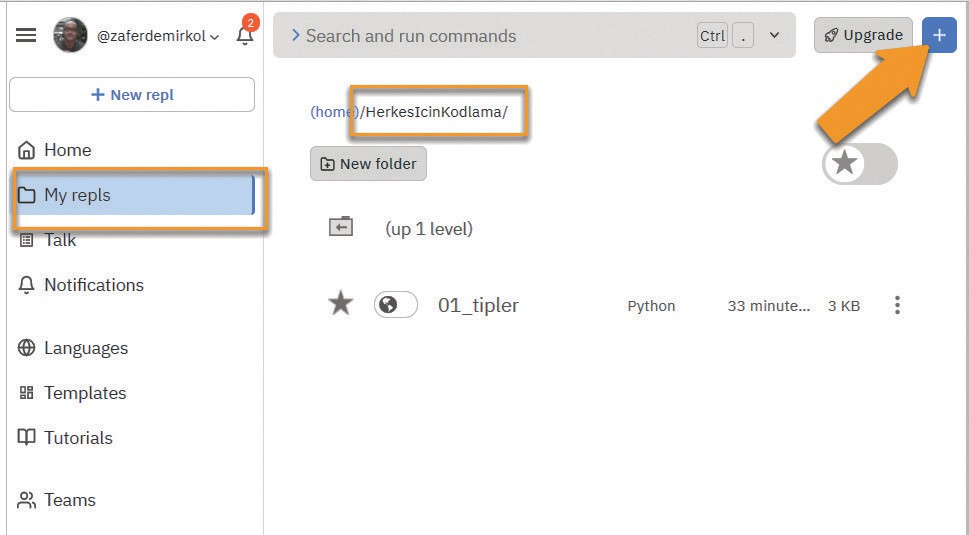
Şimdi gelin bu ortamları kısaca tanıyalım.

## Replit

https://replit.com adresinde sunulan online bir editördür. Belirli bir kullanıma kadar ücretsiz olan sürümü bizim için yeterli olacaktır.

*Zafer Demirkol // Yeni Başlayanlar İçin Kodlama*

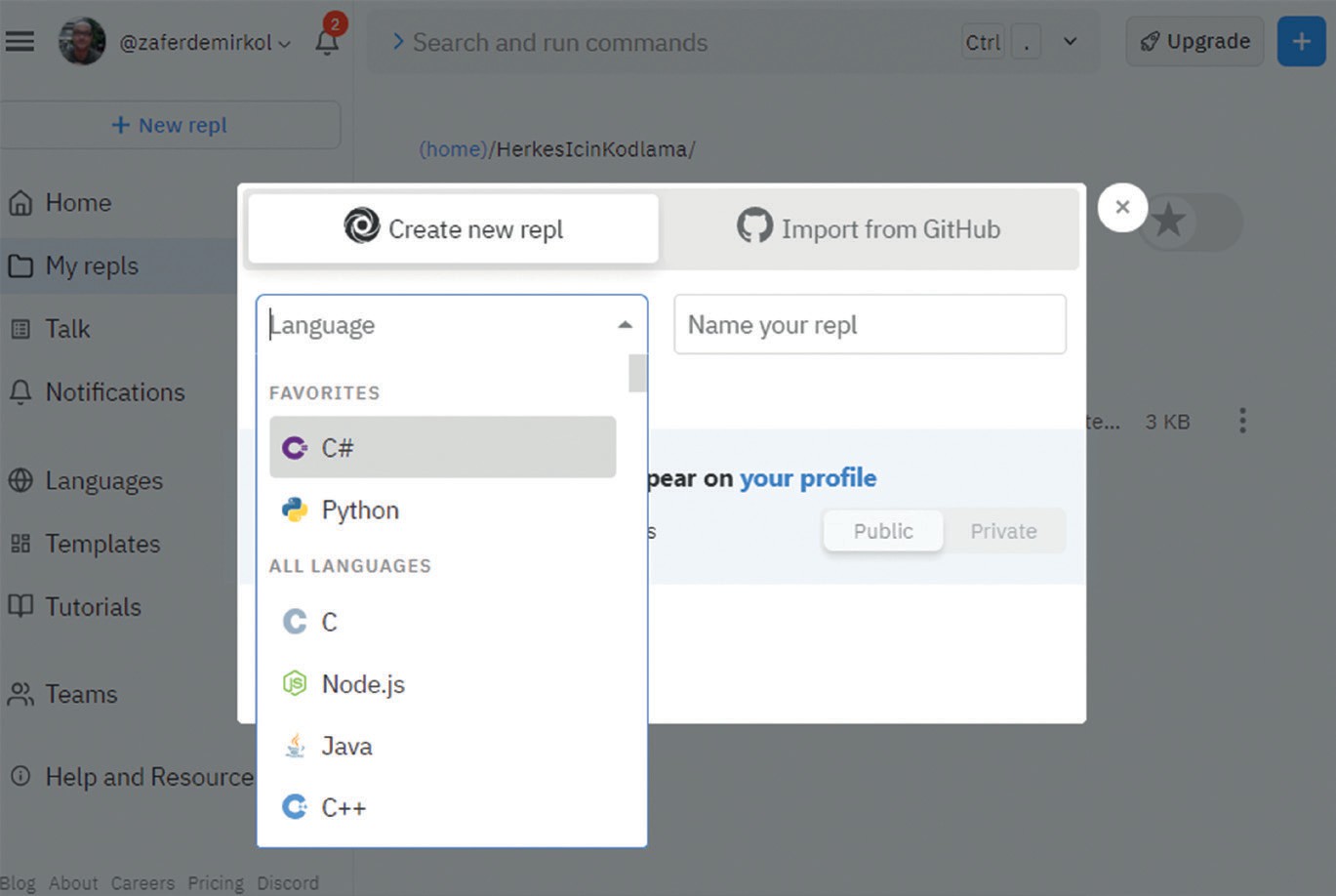
Replit’e üye olduktan sonra ana sayfadaki sol menüden yeni bir klasör oluştu- rup çalışmalarınızı bunun içine kaydedebilirsiniz:



Daha sonra sağ üst köşedeki “+” butonuna tıklayarak yeni bir “repl” yani kod sayfası ekleyebilirsiniz.

*Replit*’in en güzel özelliklerinden biri çok sayıda programlama dili ile çalış- mamıza imkân tanımasıdır.

Bu ikona tıkladığımızda bize çalışmak istediğimiz dil soruluyor:



*Zafer Demirkol // Yeni Başlayanlar İçin Kodlama*

Bu kitap boyunca ben iki programlama dili ile örnekler yapacağım, birincisi

***Python*** diğeri de ***C#****.*

***Replit*** bu her iki dilde de çalışma imkânı sunuyor. Kitap boyunca yapacağı- mız örnekleri bu dilleri seçerek uygulayabiliriz. Ancak ben kitap boyunca ***Rep- lit***’te ***C#*** kodlarını yazacağım. ***Python*** kodları için, ***Google Colab*** ve ***Visual Stu- dio Code*** kullanacağım.

Amacım mümkün olduğunca farklı editörle tanışmanızı, çalışmanızı sağlamak.

## Try.dot.net

Bazı C# kodlarını başka bir online editör olan https://try.dot.net adresindeki ***C#*** editöründe çalıştıracağım. Bu servis ***Replit*** gibi kapsamlı bir üyelik altyapısı ile çalışmıyor. Bir seferliğe mahsus C# kodlarını çalıştırmak ve denemek için tasar- lanmış. Diğer bir ifade ile, ***Replit***’te olduğu gibi kendinize ait bir alan oluşturup, kodlarınızı saklayamıyorsunuz. Bir kodu çalıştırıp sonucunu görüyorsunuz o kadar. Paylaşım da yapamıyorsunuz.

Bu editörü bazı kodlar için kullandım çünkü en son kod derleme altyapısını sunuyor ve ***Replit***’e göre daha dinamik hata gösterimi yapıyor. Burada çalıştırdı- ğınız kodları **Replit**’te kaydedebilirsiniz.

C# için birazdan göstereceğim ***Visual Studio Code*** editörünü de kullanabi- lirsiniz. Tercih sizin.

## Google Colab

Google’ın sunduğu bir servis olan ***Colab*** (Colaboratory) Python kodlarımız ile online çalışmamıza imkân tanıyan bir editördür. Gelişmiş paylaşım seçenekleri sunan ve gerekli bütün Python altyapısı ile birlikte gelen ***Colab***, aynı zamanda istediğimiz Python kütüphanelerini kurup kullanmamıza izin verir.

Özellikle Makine Öğrenimi ve Yapay Zekâ çalışmalarında kullanılan ***Colab***

günümüzde oldukça yaygın kullanılan ve kabul görmüş bir Python editörüdür.

Yapısal olarak notebook tarzında olan bir editördür. Yani kodların yanı sıra metinsel açıklamaları, notları düzenlememizi de sağlayan bir arabirimi de vardır. **Colab**’ı kullanmak için bir Google aboneliğinizin olması yeterlidir. Daha sonra **Colab** olarak bir arattırma yaparsanız servise kolayca ulaşırsınız. Bir

*Zafer Demirkol // Yeni Başlayanlar İçin Kodlama*

***Colab*** dosyasını ***Google Drive***’ınızda saklayabilir, yazdığınız kodları, indirebi- lir veya paylaşabilirsiniz.



Kod satırları ile notları yazacağınız satırlar notebook tarzı editörlerde iç içe geçmiştir.

Kodları kod satırının başındaki butona basarak çalıştırabileceğiniz gibi, ***Çalışma Zamanı*** menüsünü açıp ***“Tümünü çalıştır”*** diyerek bütün satırları yukarıdan aşağı tek tek çalıştırabilirsiniz.

## Visual Studio Code

Günümüzün en popüler en yaygın kullanılan kod editörü ***Visual Studio Co- de***’dur. Microsoft’un geliştirdiği bu editör pek çok eklentisiyle bir ekosistem de oluşturmuştur. Bu eklentileri ile pek çok dil ve uygulamayı geliştirebilecek bir arabirim sunar. Ücretsiz olması en büyük avantajlarından biridir.

https://code.visualstudio.com adresinden indirip (yaklaşık 50 MB) kolayca kurabileceğiniz ***Visual Studio Code***’u Microsoft’un diğer bir uygulama geliştir- me aracı olan Visual Studio ile karıştırmamak gerekir.

Visual Studio ağırlıklı olarak .NET altyapısını kullanarak uygulamalar geliş- tirmek için tasarlanmış bir IDE’dir (Integrated Development Environment). Çok daha büyük boyutta ve çok farklı araçlar sunan bir uygulama geliştirme aracıdır. Bu ortamı kitabımızda uygulama platformları bölümünde tanıtacağım.

***Visual Studio Code*** ise her platformda çalışan çok farklı programsal uy- gulamalar geliştirmek amaçlı oluşturulmuş, daha az kaynak ve yer tüketen bir editördür.