

Erciyes Üniversitesi

Bilgisayar Mühendisliği Bölümü

BZ 313 Yazılım Mühendisliği

17. Program Geliştirme

Integrated Development Environment (IDE)

Temel yazılım geliştirmede şunlar gerekir:

- text editor (ör., Linux için vi editor)
- derleyici
- build system (ör., Linux için make)

IDE'de ise şunların birleşimi bulunur:

- Kaynak kodu düzenleyicisi
- Toplu derleyici
- Build otomasyon araçları
- Hata ayıklayıcı (debugger)
- Ve çok daha fazlası

IDE

The screenshot displays the Visual Studio Code IDE interface. The left sidebar shows the 'Project' view with a file explorer for the 'korubin_backend' project, including folders like 'account', 'diagrams', 'kep_device', 'kep_device_status', 'kep_device_tags', and 'migrations', along with various Python files. The main editor window shows the 'kep_tag_definition\models.py' file, which contains a Django model definition for 'TagDefinition'. The code includes a 'ForeignKey' field, a 'Meta' class with 'db_table', 'verbose_name', 'verbose_name_plural', and 'ordering' attributes, and a '__str__' method. A GitHub Copilot suggestion is visible on the right side of the editor, showing a corrected version of the '__str__' method. The bottom terminal window shows the command prompt for the 'D:\Calismalar\Github\Visseria\korubin_backend' directory.

```
24 tag_group = models.ForeignKey(to: 'kep_tag_group.TagGroup')
25
26 class Meta:
27     db_table = 'kep_tag_definition'
28     verbose_name = 'Tag Definition'
29     verbose_name_plural = 'Tag Definitions'
30     ordering = ['-created_at']
31
32 def __str__(self):
33     return self.code + " - " + self.tag_group.name
34
```

Terminal output:

```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. Tüm hakları saklıdır.

(venv) D:\Calismalar\Github\Visseria\korubin_backend>
```

IDE: PyCharm

PyCharm modern IDE'lerden biridir. JetBrains firması tarafından geliştirilmiştir. Java, C/C++, Python, C# ve SQL gibi pek çok dil için çeşitli IDE'leri bulunmaktadır.

Python için:

- Kaynak kod editörü
- debugger
- derleyici
- Dokümantasyon araçları
- otomasyon araçları
- version control
- Veritabanı araçları
- Yapay zeka araçları

Ve çok daha fazlası plug-inlerle sağlanmaktadır.

Program Tasarımı: IDE

IDE'ler program tasarlarken de yardımcı olmaktadır.

IDE'le zaten bir tasarımınız olduğunu varsayar:

- sınıflar
- metotlar
- Veri yapıları
- Interface'ler

Program tasarımı için seçenekler:

- UML gibi modelleme araçlarını kullanarak program tasarımı
- Kodlama Sırasında Tasarım: Tasarım — Kod — Yeniden Tasarım Döngüsü (küçük programlar için)
- var olan frameworkler
- framework ve geliştirme araçlarını birleştiren gelişmiş ortamlar

Bu farklı yaklaşımların yönlerini birleştirmek genellikle iyidir.

Tasarım — Kodlama — Tekrar Tasarım Döngüsü

Sınıf yapısı basitse, entegre geliştirme ortamını şu amaçlarla kullanmak mümkün olabilir:

- Sınıf yapısının ve arayüzlerin bir taslağını oluşturma
- Kod yazma
- Sınıf yapısını gerektiği gibi değiştirme ve kodu gerektiği gibi yeniden düzenleme

Bu da ancak yakın iletişim halinde olan küçük ekiplerle mümkündür.

Programın maksimum boyutu, programcı(ların) deneyimine ve programın karmaşıklığına bağlıdır.

Tek bir çevik sprint'i tamamlamak mümkün olabilir.

Ancak, yeniden çalışma miktarı bunaltıcı hale gelir.

Sınıf Hiyerarşileri

Sınıf hiyerarşilerinin tasarımı zor olduğundan, mevcut frameworkleri kullanmak iyi bir seçenektir.

Genellikle sınıfların çoğu sizin için yazılmış olacaktır veya kendi alt sınıflarınız için temel olarak kullanabileceğiniz soyut sınıflar sağlanır.

Örnekler:

- Programlama dillerinin parçası olan sınıf hiyerarşileri
- toolkitler (örn. grafik kullanıcı arayüzleri için)
- Tasarım desenleri
- Web geliştirme ve mobil uygulamalar için frameworkler

Sınıf Hiyerarşileri: Programlama Dilleri

Ör: Java

Java, çok zengin bir sınıf hiyerarşileri kümesine sahip nispeten basit bir dildir.

- Java programları, işlevlerinin çoğunu standart sınıflardan alır.
- Sınıfları öğrenmek ve anlamak zordur.
- Deneyimli Java programcıları karmaşık sistemleri hızlı bir şekilde yazabilir.
- Deneyimsiz Java programcıları zarif olmayan ve hatalı programlar yazabilir.

Java ve Python gibi diller zaman içinde sınıf hiyerarşilerini sürekli olarak değiştirir. Genellikle değişiklikler, özel amaçlı işlevselliği daha genel çerçevelerle değiştirir.

Programlarınızı, sınıf hiyerarşilerini dil geliştiricilerinin amaçladığı tarzda kullanacak şekilde tasarlarsanız, uzun süreli bakıma yardımcı olabilir.

Web Geliştirme Frameworkleri

Bir web geliştirme framework'ü, web siteleri oluşturmak için bir iskelet sağlar.

Erken bir örnek, üç katmanlı bir mimari uygulayan Cold Fusion'dı.

Ruby on Rails ve Django gibi modern örnekler, genellikle üç katmanlı bir mimarinin özelliklerini bir model-view-controller (MVC) frameworküyle birleştirir.

XML ve CSS gibi web standartlarından kapsamlı bir şekilde yararlanırlar.

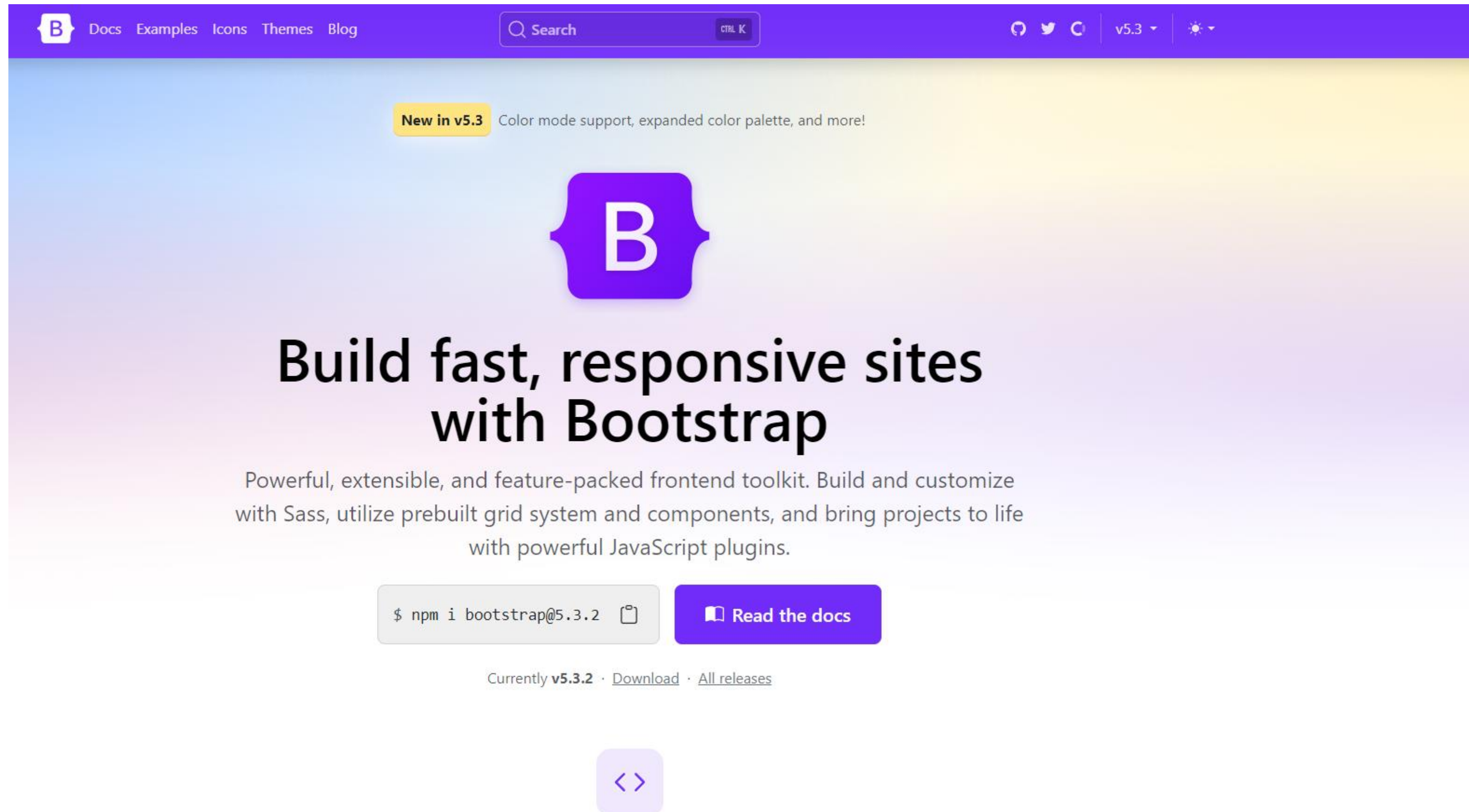
Web Geliştirme Framework'ü: Django

Django, web siteleri geliştirmek için bir Python framework'üdür.

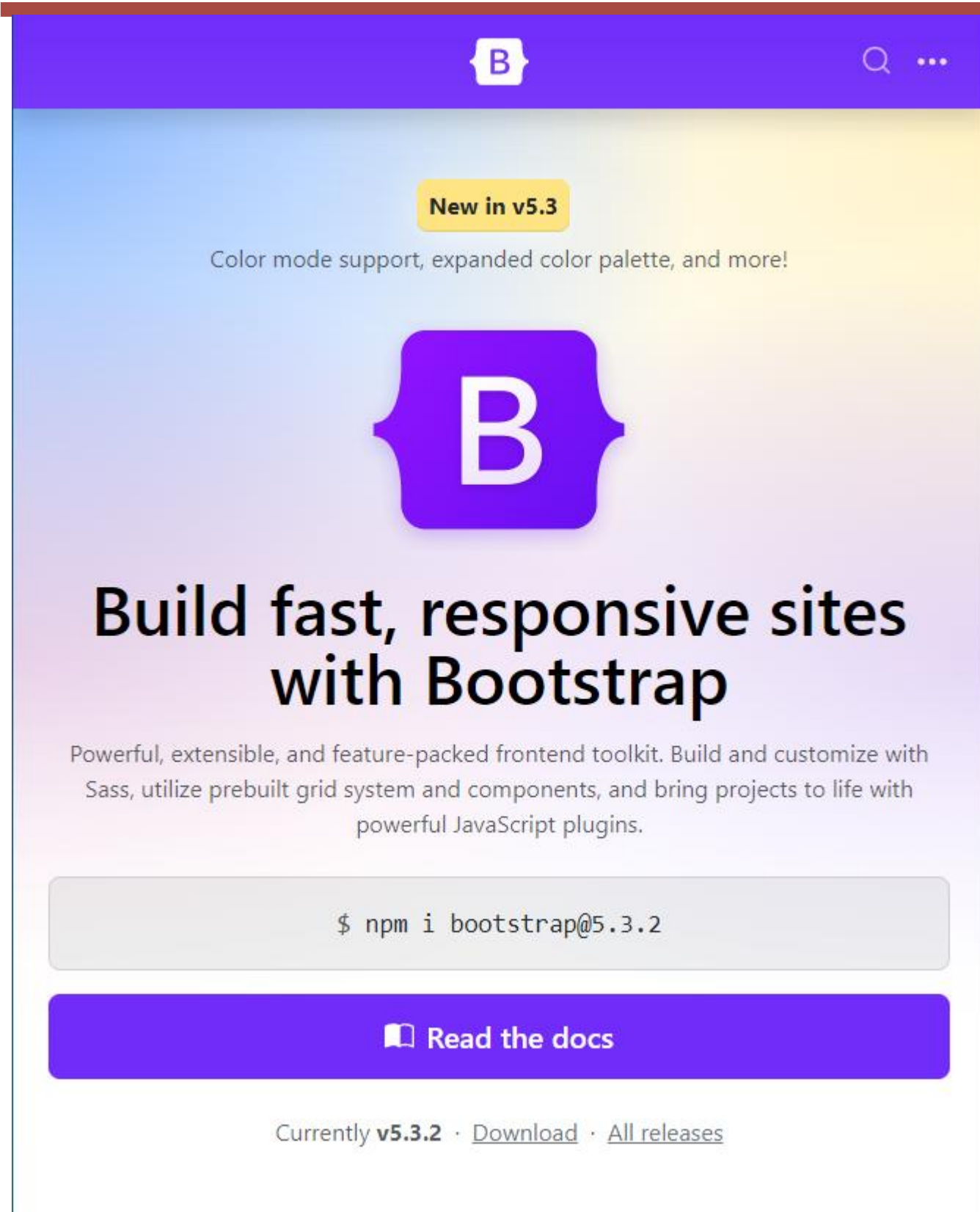
- gevşek bir şekilde MVC mimarisine dayalıdır
- Çeşitli web ve veritabanı sunucularını destekler
- Web Şablon Sistemi
- Kimlik doğrulama sistemi
- Yönetim Arayüzü
- Web saldırılarının azaltılması

Django karmaşık bir çerçevedir. Takımlar öğrenme için bolca zaman ayırmalıdır.

Responsive Web Tasarımı için Frameworkler: Bootstrap



Frameworks for Responsive Web Design: Bootstrap



CSS media query, responsive web tasarımı için güçlü bir araçtır, ancak iyi kullanımı karmaşıktır.

Bootstrap gibi framework'ler, responsive web siteleri oluşturmak için basit yollar sağlar.

Bu slayt, Mobil'de görüntülenen bir web sayfasını göstermektedir. Bunu, aynı sayfayı bir dizüstü bilgisayardaki bir pencerede gösteren önceki slaytla karşılaştırın.

Gelişmiş Geliştirme Ortamları

Uygulama frameworkleri, Django ve PyCharm gibi herhangi bir program geliştirme ortamıyla kullanılabilir

Gelişmiş bir geliştirme ortamı şunları birleştirmektedir:

- integrated development environment (IDE)
- uygulama framework'ü
- Kullanıcı Arayüzü Düzen Yöneticisi
- ve daha fazlası...

Ör:

Apple'ın iOS için Xcode'u

QuicksilverPhotos > iPad 2

Finished running QuicksilverPhotos on iPad 2

QuicksilverPhotos

ListPhotoRecords.swift

EditTextViewController.swift

PhotoRecord.swift

QuicksilverPhotos

PhotoViewController.swift

Main.storyboard

PhotoTableViewCell.swift

PhotoTableViewController.swift

Assets.xcassets

LaunchScreen.storyboard

Info.plist

AppDelegate.swift

Products

Photo Image View

New Photo

[Enter Title]

No photo selected

Photo...r.swift

photoPicker(...)

```
//
// PhotoViewController.swift
// Quicksilver Photos
//
// Created by William Arms on 2/27/16.
//

import UIKit

class PhotoViewController: UIViewController,
    UITextFieldDelegate,
    UIImagePickerControllerDelegate,
    UINavigationControllerDelegate
{
    // MARK: Properties

    @IBOutlet weak var titleText: UITextField!
    @IBOutlet weak var photoImageView: UIImageView!
    @IBOutlet weak var photoDescription: UITextView!
    @IBOutlet weak var saveRecordButton:
        UIBarButtonItem!

    var viewRecordsList = ListPhotoRecords()
    var currentViewIndexPath: NSIndexPath?
    var wasNewRecordAdded = false

    var photoRecord: PhotoRecord?
    let defaultPhotoDescription = "[Enter description of
    photograph]"
    let defaultTitle = ""/[Enter title]"

    // MARK: Navigation

    override func viewDidLoad()
    {
        super.viewDidLoadLoad()
        titleText.delegate = self

        setCurrentRecord()
    }

    func setCurrentRecord()
    {
        if currentViewIndexPath!.row == viewRecordsList.
            list.count // Add new record
        {
            photoRecord = PhotoRecord(title:
                defaultTitle, photo: UIImage(named:
                "noPhotoSet"), photoText:
                defaultPhotoDescription)
            viewRecordsList.list += [photoRecord!]
            wasNewRecordAdded = true
        }
        else
        {
            photoRecord = viewRecordsList.list
                [currentViewIndexPath!.row]
            wasNewRecordAdded = false
        }
    }
}
```

Image View

Image noPhotoSet

Highlighted Highlighted Image

State ☐ Highlighted

View

Mode Aspect Fit

Semantic Unspecified

Tag 0

Interaction ☒ User Interaction Enabled

☐ Multiple Touch

Alpha 1

Background Black Color

Tint

Drawing ☒ Opaque ☐ Hidden

☒ Clears Graphics Context

☐ Clip Subviews

☒ Autorelease Subviews

Stretching 0 0

X Y

1 1

Width Height

☒ Installed

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

Navigation Controller - A controller that manages navigation.

Gelişmiş Geliştirme Ortamları

Gelişmiş bir geliştirme ortamı, bir geliştiricinin ihtiyaç duyduğu her şeyi sağlamayı amaçlar.

Geliştiricinin sağlanan program seçeneklerini izlemesi beklenir.

Örneğin, Xcode iOS ile kullanıldığında çok özel bir amacı vardır: iPhone'lar, iPad'ler gibi Apple cihazları için mobil uygulamalar.

- Özel programlama dili (Swift veya Objective C)
- MVC framework

Genel program tasarımını kabul ederseniz, çok güçlüdür:

- Grafik arayüzlerin otomatik yerleşimi
- Kullanıcı arayüzleri ve gezinme için kapsamlı sınıf seti
- Tüm Apple aygıtları için simülatörler

Geliştirme Framework'lerini Kullanma

Geliştirme frameworkleri güçlü ve esnektir.

Uygulamanız framework'e uyuyorsa, program tasarımının çoğunu yaparlar ve herhangi bir uygulamanın standart parçalarının çoğu için yüksek kaliteli kod sağlarlar.

Uygulamanın bazı bölümlerinin ayrı ayrı tasarlanması gerekebilir.

Ama dikkat edin:

- Uygulamanızı sağlanan framework içinde oluşturmak zorunda kalırsınız.
- Framework'ler sürekli olarak değiştirilir.
- Bu framework'ler karmaşıktır ve öğrenilmesi uzun zaman alır.

Production Programming

Murphy Yasası:

Bir şeyler ters gidebilirse, gider.

Zorluklar:

- Kod, farklı sistem yazılımlarıyla uzun vadede korunmalıdır.
- Arayüzler yeni ve beklenmedik şekillerde kullanılır.
- Olası her hata, sonunda mümkün olan en kötü zamanda ortaya çıkacaktır (kötü veriler, donanım ve sistem yazılımı arızaları).
- Güvenlik saldırıları olması muhtemeldir.

Production Programming

Robust programlama:

- Basit kod yaz.
- Riskli programlama yapılarından kaçının.
- Kodun okunması zorsa, yeniden yazın.
- Değişikliklerden sonra sistem durumunu kontrol etmek için kodu yedekleyin.
- Örtük (implicit) varsayımları açıkça (explicit) test edin, örneğin diğer rutinlerden alınan tüm parametreleri kontrol edin.
- Kaynak kodundaki tüm uyarıları ortadan kaldırın.
- Kodunuz için kapsamlı bir test senaryosu kümesine sahip olun.

Bir üretim ortamında, kodlama ve test için çok daha fazla zaman harcayacağınızı unutmayın.

Program Geliştirmede AI

EVOSUITE

Sample explanations

Dense code

Web development

Conceptual Insight

Tensorflow

C++ templates

Contextual Insight

DevOps

```
import tensorflow as tf

@wrap_objective
def l2(batch=None):
    def inner(T):
        return -tf.reduce_mean((T("input") - 0.5)**2)
    return inner
```

Show Explanation

- The code is a function that takes an input tensor and returns the squared difference between it and 0.5.

- The code is written to take in a batch size, which defaults to None.

- The code is used to calculate the L2 norm of a batch of data.

- This code calculates the L2 norm for each sample in a batch, and then returns the average value across all samples.

EvoMaster

Public

Watch 24

Fork 69

Starred 383

master

107 branches

22 tags

Go to file

Add file

Code

arcuri82 Merge pull request #858 from EMResearch/config-toml

2d21adc 2 days ago

8,696 commits

.circleci	clarification	2 years ago
.github	fix for IntelliJ weird autocomplete bug	4 days ago
.mvn	trying workaround for Kotlin compiler issue	last year
client-dotnet	2.0.1-SNAPSHOT	2 months ago
client-java	clarifications	3 days ago
client-js	2.0.1-SNAPSHOT	2 months ago
core-driver-it	refactored package names	last month
core-graphql-it	refactored package names	last month
core-it	more fixes	last month
core	Merge pull request #858 from EMResearch/config-toml	2 days ago
dbconstraint	2.0.1-SNAPSHOT	2 months ago

About

The first open-source AI-driven tool for automatically generating system-level test cases (also known as fuzzing) for web/enterprise applications. Currently targeting whitebox and blackbox testing of Web APIs, like REST, GraphQL and RPC (e.g., gRPC and Thrift).

kotlin

java

testing

graphql

rest

grpc

thrift

evolutionary-algorithms

fuzzing

api-rest

fuzzer

api-testing

rpc-api

test-case-generation

search-based-software-testing

Readme

LGPL-3.0 license

Activity

383 stars

24 watching



codeium

Google Translate

rise

Playground

Blog

Company

Get Codeium

Introducing Codeium Teams

The modern coding superpower

A free extension to

Get Extension

Try in browser

Loved by hundreds of thousands of developers

VS Code 447.5K

IntelliJ 201.8K

PyCharm 30.0K

JetBrains 2.3K

and 40+ more IDEs

GitHub Copilot

The world's most widely adopted AI developer tool.

Buy now

Compare Plans

Erciyes Üniversitesi

Bilgisayar Mühendisliği Bölümü

BZ 313 Yazılım Mühendisliği

17. Program Geliştirme

Ders Sonu