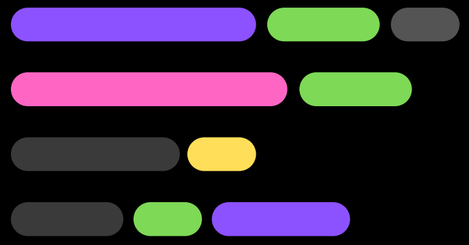
INTRODUCTION TO {AP PROJECT}

Instructor: Dr.Vahidi Asl
Presenter: Nazanin Farhanj
Fall 1403



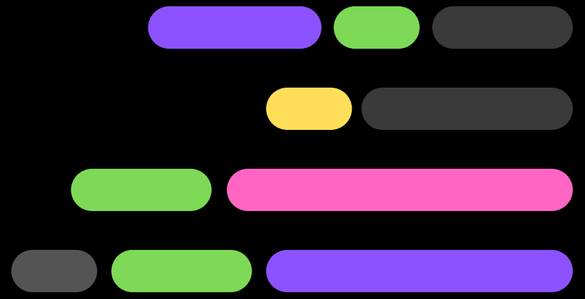
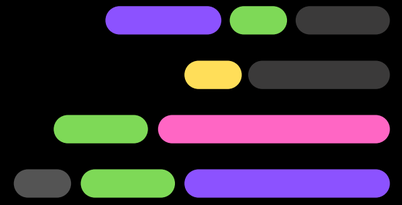
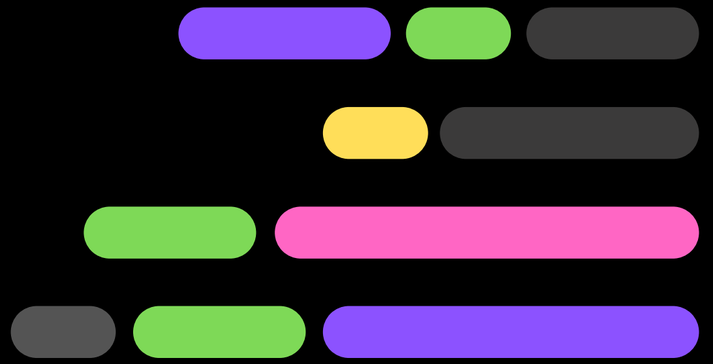


TABLE OF CONTENTS



01

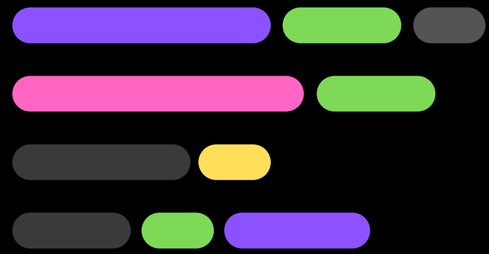


WHAT IS A FRAMEWORK?

A framework is a structure that you can build software on. It serves as a foundation, so you’re not starting entirely from scratch. Frameworks are typically associated with a specific programming language and are suited to different types of tasks.

A framework is similar to a template in that you can modify it and add certain features and higher functionalities to create a complex and broad project that many people can use.

WHY DO WE USE FRAMEWORKS?

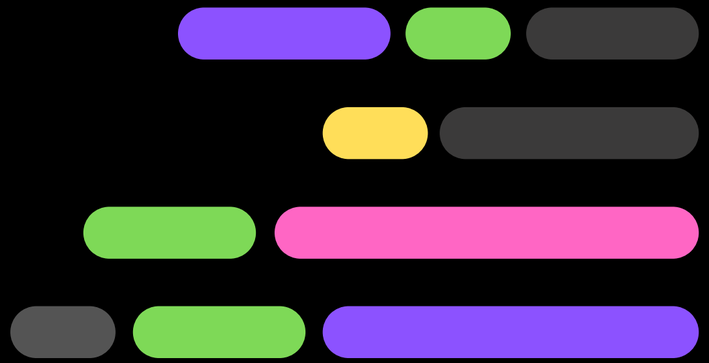


It's a good idea to use a software framework rather than re-inventing the wheel from scratch for numerous reasons. And perhaps the most important reason is that you won't have to write everything from scratch. This reduces the possibility of adding errors to your code.

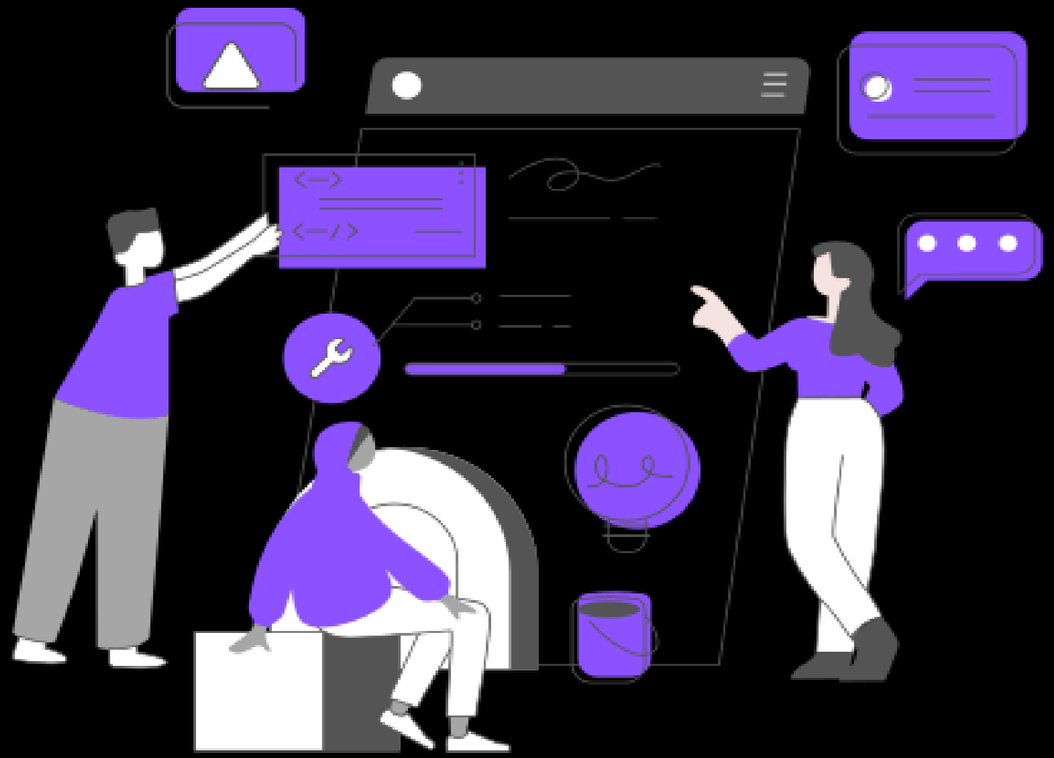
Other reasons to use a framework include:

- More secure code
- Simpler testing and debugging
- Avoiding duplicate code
- Clean and easily adaptable code
- Able to focus on writing code specific to the project
- Can be extended

And a lot more.



CHALLENGES OF USING A FRAMEWORK



- Compatibility issues
- Over-Reliance on the Framework
- Limited flexibility
- Too many features
- Learning curve
- Choosing the Right Tools and Frameworks

FRAMEWORKS VS. PROGRAMMING LANGUAGES

Framework

Frameworks are dynamic, allowing them to be reused when needed. More or less, they can be thought of as templates. Templates in web design, for example, make it much easier to create a website.

Programming language

A programming language is a set of commands that instructs a software or computer’s functionality. While a framework provides the foundation for implementing the language’s commands.

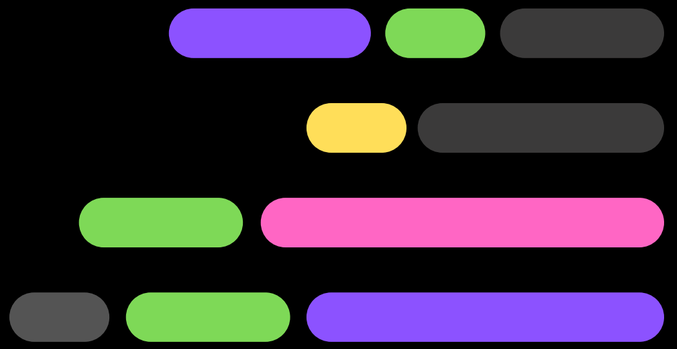
FRAMEWORKS VS. PROGRAMMING LANGUAGES

FRAMEWORK

- Frameworks are the architecture that already includes some of the code that is used in the development of an application.
- Templates are present to develop applications.
- Frameworks depend on the type of applications to be developed like mobile applications, web applications, etc.

PROGRAMMING LANGUAGE

- Developers have to write the code from scratch to develop an application.
- No templates are available to develop an application.
- Programming languages depend on procedures, functions, and objects.



FRAMEWORKS VS. LIBRARIES

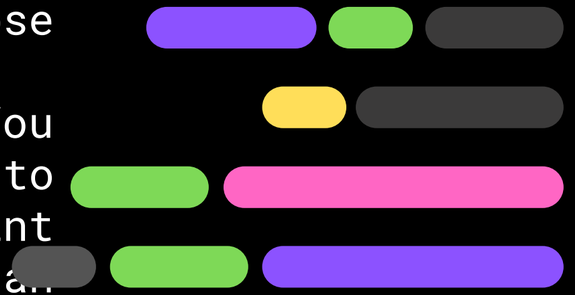
Both frameworks and libraries are code written by someone else that is used to help solve common problems.

The technical difference between a framework and a library is defined by a concept known as inversion of control. When you use a library, you control the application flow, including when and where to contact the library. When you use a framework, the framework itself controls the flow.

Let's use a house as a metaphor for web development concepts.

A library is like going to Ikea. You already have a home, but you need a bit of help with furniture. You don't feel like making your own table from scratch. Ikea allows you to pick and choose different things to go in your home. You are in control.

A framework, on the other hand, is like building a model home. You have a set of blueprints and a few limited choices when it comes to architecture and design. Ultimately, the contractor and blueprint are in control. And they will let you know when and where you can provide your input.



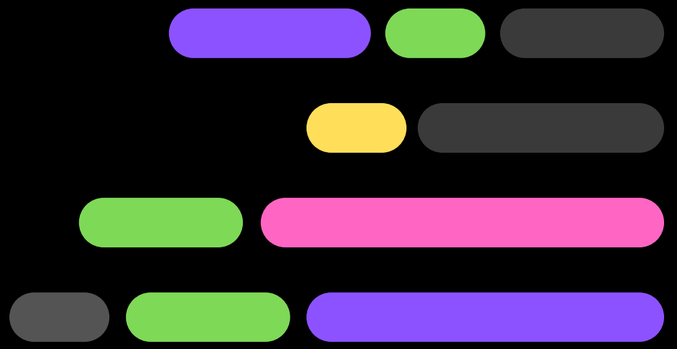
FRAMEWORKS VS. LIBRARIES

FRAMEWORK

- Includes a variety of APIs, compilers, support applications, libraries, and so on.
- Are tough to replace.
- Inversion of control, i.e. the framework calls us.
- The construction of a framework necessitates large amounts of code, which reduces performance and increases load time.

LIBRARY

- A set of assistance modules, objects, classes, functions, pre-written code, and so on.
- Can be easily substituted by another library.
- When we call a method from a library, we are in control.
- Since developing a library needs less code, performance and load time are improved.

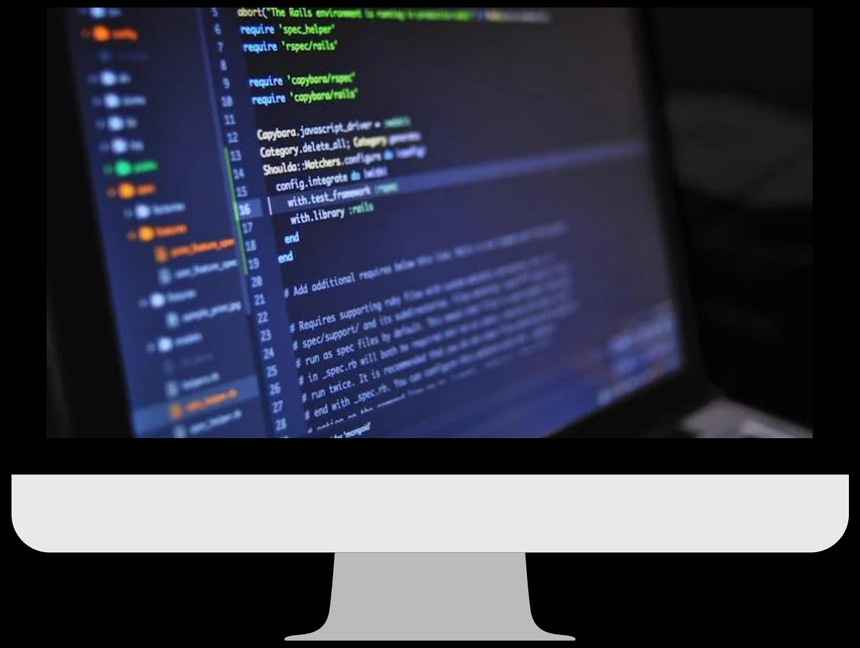


MOBILE APP DEVELOPMENT FRAMEWORKS

Cross-platform applications are very useful as they are developed once and can be installed on any platform. Mobile frameworks help in the development of mobile applications. These frameworks consist of the code related to the common tasks which are used in most of the mobile apps.

WHAT IS DART?

Dart is an open-source, general-purpose, object-oriented programming language with C-style syntax developed by **Google in 2011**. The purpose of Dart programming is to create a frontend user interfaces for the web and mobile apps. It is under active development, compiled to native machine code for building mobile apps, inspired by other programming languages such as Java, JavaScript, C#, and is Strongly Typed. Since Dart is a compiled language so you cannot execute your code directly; instead, the compiler parses it and transfer it into machine code.



WHAT IS FLUTTER?

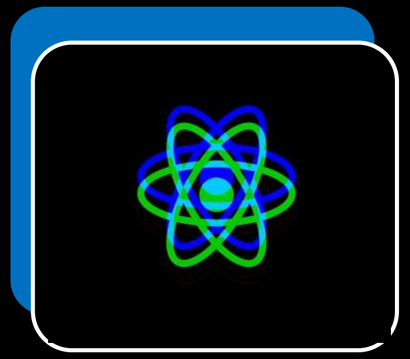
Flutter is an open-source UI software development toolkit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, and the web from a single codebase.
It uses the Dart programming language and provides a rich set of pre-designed widgets to create beautiful, fast, and responsive user interfaces.

```dart
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Welcome to Flutter'),
        ),
        body: const Center(
          child: const Text('Hello World'),
        ),
      ),
    );
  }
}
```

POPULAR FRAMEWORKS

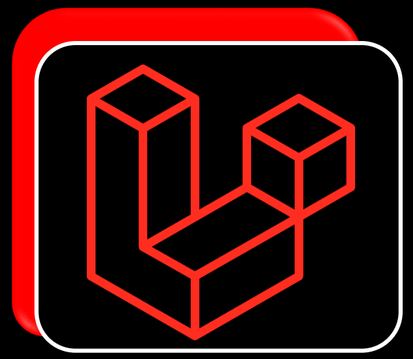


REACT

React is a widely-used JavaScript library for building interactive user interfaces. It's ideal for dynamic web applications, single-page applications (SPAs), and mobile.

NODE.JS

Node.js is an open-source and cross-platform JavaScript runtime environment. Node.js allows developers to create both front-end and back-end applications.



LARAVEL

Laravel is a PHP-based backend framework known for its elegance and simplicity. It follows the MVC pattern and provides rich set of tools, including a modular packaging system and a powerful ORM.

DJANGO

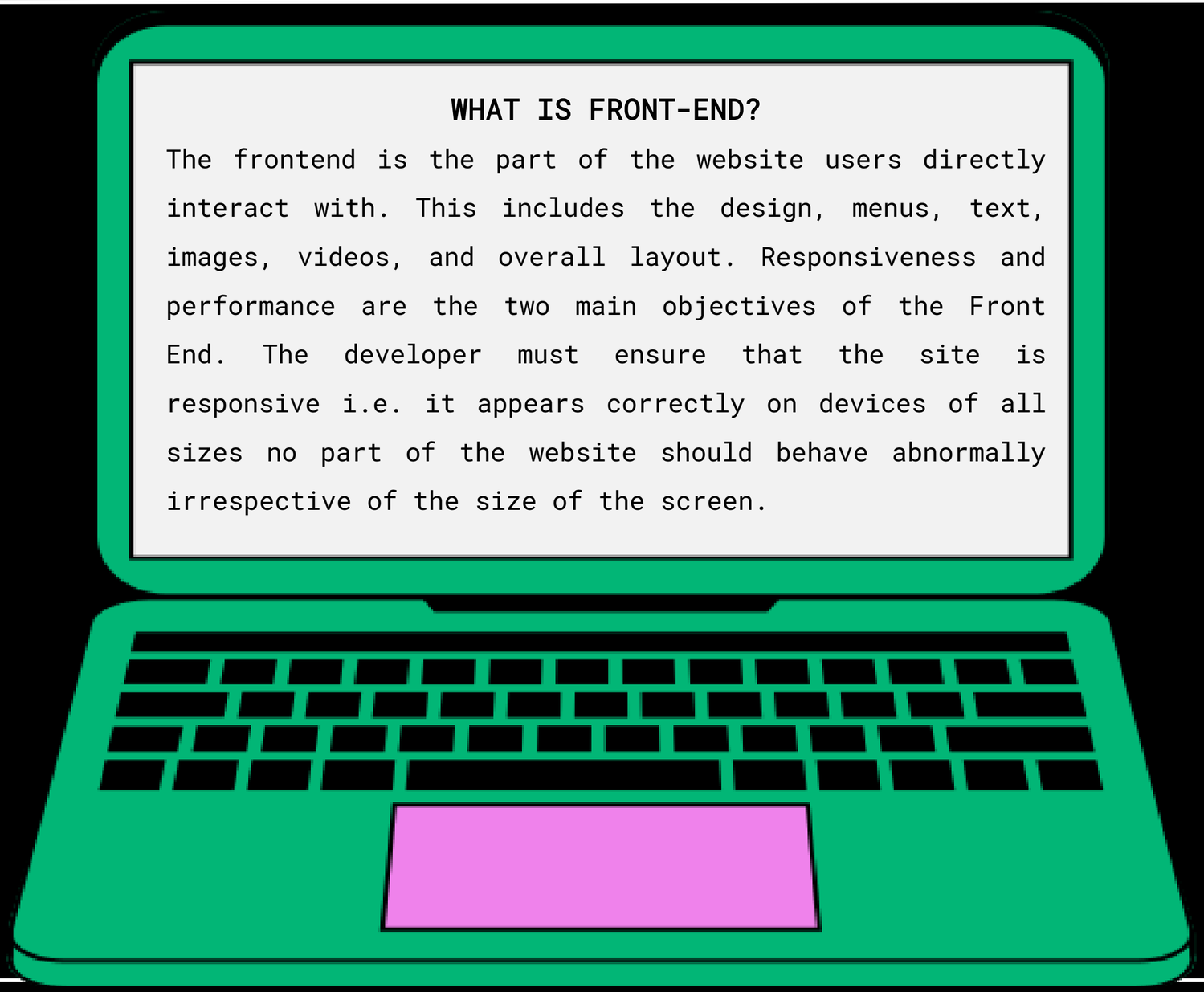
Django is a Python-based backend framework simplifying the creation of complex, scalable, and data-driven applications. Django powers websites like Instagram and Mozilla.

ANGULAR

Angular is an open-source JavaScript-based frontend framework for creating high-performance, large-scale applications. is perfect for enterprise-level applications, with notable users like PayPal, Forbes, and Microsoft Xbox.

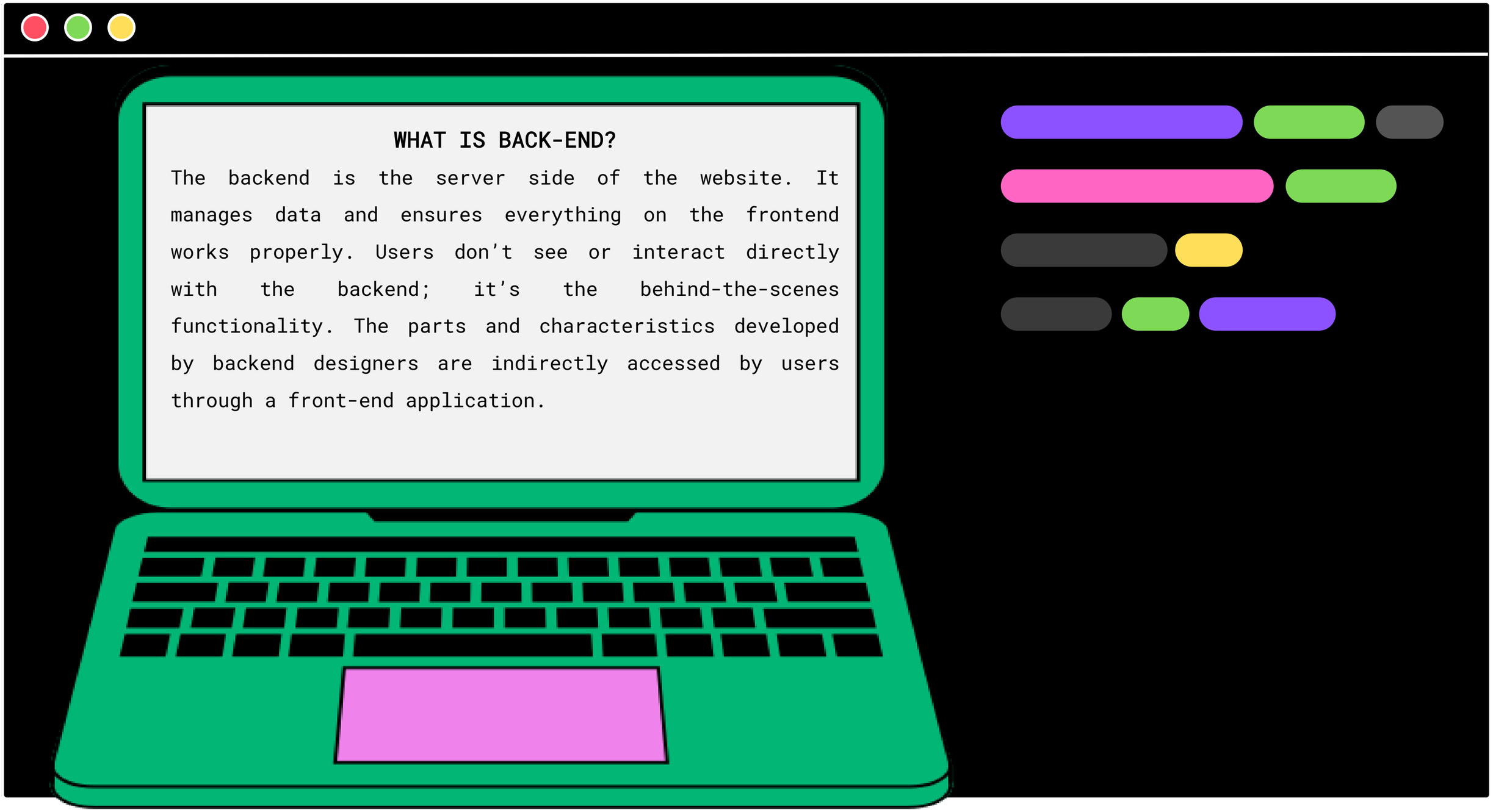
WHAT IS FRONT-END?

The frontend is the part of the website users directly interact with. This includes the design, menus, text, images, videos, and overall layout. Responsiveness and performance are the two main objectives of the Front End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen.

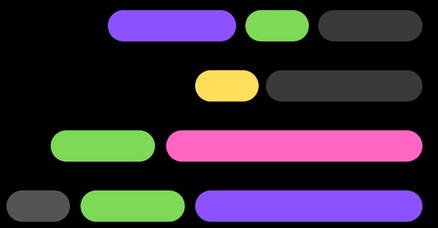


WHAT IS BACK-END?

The backend is the server side of the website. It manages data and ensures everything on the frontend works properly. Users don’t see or interact directly with the backend; it’s the behind-the-scenes functionality. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application.



ANDROID/IOS DEVELOPMENT LANGUAGES



- Java

Java is a great language to experience the full joys of Android App Development. However, it may be a little complex for beginners who would prefer to start with something easier and then return to it.

- Kotlin

Kotlin is much simpler for beginners to try as compared to Java and it can also be used as an “entry point” for Android App Development.

- C++

While C++ is useful for Android App Development in some cases, it is much more difficult to set up and is much less flexible.

- Dart

Dart mainly focuses on making UI development easier for developers with features such as hot-reload, which lets developers see changes instantly while working on the app. Dart is also known for its fast performance.

- Swift

Swift is a general-purpose programming language that’s approachable for newcomers and powerful for experts. It is fast, modern, safe, and a joy to write.

THANK YOU!