

basic java tutorial for beginners

Basic Java Tutorial for Beginners: Your First Steps into Programming

basic java tutorial for beginners is the perfect starting point if you're eager to dive into the world of programming but don't know where to begin. Java is one of the most popular and versatile programming languages used worldwide, powering everything from mobile apps to large-scale enterprise systems. This tutorial will guide you through the fundamental concepts of Java programming, helping you build a solid foundation for your coding journey.

Why Choose Java as Your First Programming Language?

Before we jump into the basics, it's worth understanding why Java is often recommended for beginners. Java's syntax is clear and structured, making it easier to grasp programming logic. Plus, it is platform-independent thanks to the Java Virtual Machine (JVM), meaning your code can run on any device without modification. This portability, combined with a vast ecosystem and community support, makes learning Java extremely rewarding for newcomers.

Setting Up Your Java Environment

Installing Java Development Kit (JDK)

To start coding in Java, you need to install the Java Development Kit (JDK). The JDK includes essential tools such as the compiler and runtime needed to write and execute Java programs. You can download the latest JDK version from Oracle's official website or use OpenJDK, an open-source alternative.

Choosing an Integrated Development Environment (IDE)

While it's possible to write Java code using a simple text editor, beginners will find it easier to use an IDE. Popular options include IntelliJ IDEA, Eclipse, and NetBeans. These environments provide helpful features like code completion, debugging tools, and project management that make programming more intuitive.

Understanding the Basic Structure of a Java

Program

A simple Java program has a specific structure that you'll need to follow. Here's a quick breakdown of the main components:

```
```java
public class HelloWorld {
 public static void main(String[] args) {
 System.out.println("Hello, world!");
 }
}
```
```

- `public class HelloWorld`: This declares a class named HelloWorld. In Java, every application must contain at least one class.
- `public static void main(String[] args)`: This is the main method where the program starts executing.
- `System.out.println("Hello, world!");`: This line prints text to the console.

This simple example introduces you to the core concepts of classes, methods, and statements.

Basic Java Concepts You Need to Know

Variables and Data Types

Variables are containers for storing data values. Java is a statically typed language, which means you must declare the variable type before using it. Common data types include:

- **int**: for integer values (e.g., 10)
- **double**: for decimal numbers (e.g., 3.14)
- **char**: for single characters (e.g., 'A')
- **boolean**: for true/false values

Example:

```
```java
int age = 25;
double price = 19.99;
char grade = 'A';
boolean isJavaFun = true;
```
```

Operators and Expressions

Operators perform operations on variables and values. Java supports arithmetic operators such as +, -, *, /, and %, as well as comparison operators (==, !=, >, <) and logical operators (&&, ||, !). Understanding how to use these is crucial for making decisions and performing calculations.

Control Flow: Conditionals and Loops

Control flow statements help your program make decisions and repeat tasks. The most common control flow structures include:

- **if-else statements:** Execute code blocks based on conditions.
- **switch statements:** Select one of many code blocks to execute.
- **for loops:** Repeat a block of code a specific number of times.
- **while and do-while loops:** Repeat code while a condition is true.

Example of an if-else statement:

```
```java
int number = 10;
if (number > 0) {
 System.out.println("Number is positive");
} else {
 System.out.println("Number is zero or negative");
}
```
```

Working with Methods in Java

Methods are blocks of code designed to perform a particular task. They help you organize your program into reusable pieces, making your code cleaner and easier to maintain. A method typically has a name, return type, and parameters.

Here's an example of a simple method:

```
```java
public static int addNumbers(int a, int b) {
 return a + b;
}
```
```

You can then call this method from the main method:

```
```java
int sum = addNumbers(5, 7);
```
```

```
System.out.println("Sum is: " + sum);  
```
```

## Object-Oriented Programming Basics

Java is an object-oriented language, so understanding the basics of this paradigm is essential. At its core, object-oriented programming (OOP) revolves around classes and objects.

### Classes and Objects

- A **class** is a blueprint for creating objects. It defines properties (fields) and behaviors (methods).
- An **object** is an instance of a class.

Example:

```
```java  
public class Dog {  
    String name;  
    int age;  
  
    void bark() {  
        System.out.println("Woof!");  
    }  
}  
```
```

Creating an object:

```
```java  
Dog myDog = new Dog();  
myDog.name = "Buddy";  
myDog.age = 3;  
myDog.bark();  
```
```

### Encapsulation

Encapsulation involves hiding the internal state of an object and requiring all interaction to be performed through an object's methods. This is typically achieved by making fields private and providing public getter and setter methods.

Example:

```
```java
public class Person {
    private String name;

    public String getName() {
        return name;
    }

    public void setName(String newName) {
        this.name = newName;
    }
}
```
```

## Handling Errors with Exception Handling

Sometimes, your program might encounter unexpected situations like dividing by zero or accessing invalid array indexes. Java provides a robust way to handle such errors using try-catch blocks.

Example:

```
```java
try {
    int result = 10 / 0;
} catch (ArithmeticException e) {
    System.out.println("Error: Cannot divide by zero.");
}
```
```

This approach helps your program continue running even when errors occur.

## Tips for Learning Java Effectively

Learning Java or any programming language can feel overwhelming at first, but with the right approach, it becomes manageable and enjoyable.

- **Practice regularly:** Writing code daily strengthens your understanding and builds muscle memory.
- **Experiment:** Don't hesitate to modify examples and try your own ideas.
- **Understand concepts deeply:** Focus on why things work instead of just memorizing syntax.

- **Use online resources:** Websites like Oracle Java Tutorials, Codecademy, and freeCodeCamp offer invaluable practice and explanations.
- **Join communities:** Forums like Stack Overflow and Reddit's r/learnjava provide support and motivation.

## Next Steps After Mastering the Basics

Once you're comfortable with the fundamentals covered in this basic java tutorial for beginners, it's time to explore more advanced topics such as:

- Collections and data structures like ArrayList and HashMap
- File input/output operations
- Multithreading and concurrency
- Java frameworks (Spring, Hibernate)
- Building graphical user interfaces (GUIs) with JavaFX or Swing

Exploring these areas will expand your skill set and prepare you for real-world Java development projects.

---

Getting started with Java can open many doors in software development. With patience and practice, the basics will become second nature, and you'll be ready to tackle more complex programming challenges. Keep coding, stay curious, and enjoy the journey!

## Frequently Asked Questions

### What is Java and why is it important for beginners to learn?

Java is a high-level, object-oriented programming language widely used for building platform-independent applications. It's important for beginners because of its simplicity, portability, and vast community support, making it an excellent choice to start programming.

### How do I set up the Java Development Kit (JDK) on my computer?

To set up the JDK, download the latest version from the official Oracle website or OpenJDK. After downloading, install it following the instructions for your operating system, and set the environment variables (like PATH) to run Java commands from the terminal.

## What is the structure of a basic Java program?

A basic Java program includes a class definition, a main method, and statements inside the main method. For example:

```
public class HelloWorld {
 public static void main(String[] args) {
 System.out.println("Hello, World!");
 }
}
```

This structure is essential as the main method is the entry point of any Java application.

## How do I compile and run a Java program from the command line?

To compile a Java program, use the command 'javac FileName.java' which generates a bytecode file 'FileName.class'. To run the program, use 'java FileName'. Ensure the JDK is correctly installed and the PATH variable is set.

## What are variables and data types in Java?

Variables are containers for storing data values. Java has several data types including primitive types like int (integer), double (floating-point number), char (character), boolean (true/false), and reference types like objects. Understanding these is fundamental for managing data in Java programs.

## Additional Resources

Basic Java Tutorial for Beginners: A Detailed Introduction to Java Programming

**basic java tutorial for beginners** serves as an essential starting point for anyone aiming to grasp the fundamentals of Java, one of the most widely used programming languages in the world. Java's versatility, platform independence, and robust security features have made it a preferred choice for developers across industries, ranging from web and mobile applications to enterprise software and embedded systems. This article delves deeply into the core concepts of Java, equipping novices with a structured understanding while highlighting the language's unique features and practical applications.

## Understanding Java: Why It Matters for Beginners

Java's prominence in the programming landscape is not accidental. Since its inception by Sun Microsystems in 1995, Java has evolved into a mature, object-oriented language that emphasizes code portability through the Java Virtual Machine (JVM). For beginners, this "write once, run anywhere" philosophy means that Java programs can operate across various devices and operating systems without modification, an advantage that other

languages lack to the same extent.

Moreover, Java's syntax is designed to be straightforward yet powerful, making it an ideal language for those new to programming. It balances readability with the depth needed to create complex applications. This balance is why a basic Java tutorial for beginners often includes a focus on syntax, data types, control structures, and object-oriented programming principles.

## Core Concepts in a Basic Java Tutorial for Beginners

### 1. Setting Up the Development Environment

Before writing Java code, beginners must set up their environment correctly. This step involves installing the Java Development Kit (JDK), which contains essential tools like the Java compiler (javac) and runtime environment (JRE). Popular integrated development environments (IDEs) such as Eclipse, IntelliJ IDEA, or NetBeans can further streamline coding by providing code completion, debugging, and project management features.

### 2. Java Syntax and Structure

At the heart of any basic Java tutorial for beginners is understanding the language's syntax. Java programs consist of classes and methods, with a typical starting point being the `main` method:

```
```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```
```

This simple example introduces several key elements:

- **Class Declaration:** Defines a blueprint for objects.
- **Main Method:** The entry point for Java applications.
- **Statements and Semicolons:** Java statements end with semicolons.
- **Standard Output:** Using `System.out.println` to print text.

### 3. Data Types and Variables



Java is a statically typed language, meaning every variable must have a declared type. Basic data types include:

- **int:** for integers (e.g., 10, -5)
- **double:** for floating-point numbers (e.g., 3.14)
- **char:** for single characters (e.g., 'a')
- **boolean:** for true/false values
- **String:** for sequences of characters (technically an object)

Proper understanding of variables and data types is crucial, as it affects memory management and program logic.

## 4. Control Flow: Conditionals and Loops

Control structures enable programs to make decisions and repeat actions. In Java, common statements include:

1. **If-else Statements:** Execute code blocks based on conditions.
2. **Switch Statements:** Select from multiple paths based on variable values.
3. **Loops:** For repeated execution, including for, while, and do-while.

Example of an if-else statement:

```
```java
int number = 10;
if (number > 0) {
    System.out.println("Positive number");
} else {
    System.out.println("Zero or negative number");
}
```
```

## 5. Introduction to Object-Oriented Programming (OOP)

Java is fundamentally an object-oriented language, and a basic Java tutorial for beginners emphasizes core OOP concepts such as:

- **Classes and Objects:** Classes define templates; objects are instances.
- **Inheritance:** Enables one class to inherit fields and methods from another.
- **Encapsulation:** Restricts direct access to some components.
- **Polymorphism:** Allows methods to behave differently based on the object.

For example, creating a simple class:

```
```java
public class Car {
    String color;
    int speed;

    void accelerate() {
        speed += 10;
    }
}
```
```

Then instantiating an object:

```
```java
Car myCar = new Car();
myCar.color = "Red";
myCar.accelerate();
```
```

## Comparing Java with Other Beginner-Friendly Languages

While Java is a solid choice for beginners, it's useful to briefly compare it with other popular languages such as Python and JavaScript. Python is often praised for its minimalistic syntax and rapid development capabilities, making it suitable for quick prototyping and data science. JavaScript excels in client-side web development, with its event-driven model.

However, Java's strong typing, extensive standard libraries, and robust community support give it an edge for learners interested in software engineering and large-scale applications. Its performance is generally superior to interpreted languages due to the JVM's Just-In-Time (JIT) compilation.

# Advantages and Challenges in Learning Java

A basic Java tutorial for beginners not only covers technical concepts but also addresses the practical challenges learners might face.

## Advantages:

- **Platform Independence:** Java programs run on any device with a JVM.
- **Rich API:** Provides extensive standard libraries for collections, networking, file I/O, and more.
- **Strong Community Support:** A vast ecosystem of frameworks, tools, and forums.
- **Industry Demand:** Java skills are consistently in high demand in the job market.

## Challenges:

- **Verbose Syntax:** Java requires more boilerplate code compared to languages like Python.
- **Steep Learning Curve:** Object-oriented principles and JVM intricacies can be overwhelming initially.
- **Performance Overhead:** Though efficient, Java programs can consume more memory than native languages like C++.

## Best Practices for Beginners in Java Programming

To maximize the benefits of a basic Java tutorial for beginners, adopting certain learning strategies is advisable:

1. **Hands-On Coding:** Consistently write and test code to reinforce concepts.
2. **Understand Errors:** Debugging is a significant learning tool; analyze and fix compilation or runtime errors.
3. **Code Readability:** Use meaningful variable names and comments.

4. **Explore Standard Libraries:** Familiarize yourself with Java's built-in classes to avoid reinventing the wheel.
5. **Build Small Projects:** Create simple applications like calculators or to-do lists to apply learned concepts.

## Emerging Trends and Java's Future Outlook

Java continues to evolve, with recent versions introducing features like records, pattern matching, and enhanced support for functional programming through lambda expressions. These advancements aim to reduce verbosity and improve developer productivity, making the language even more accessible to beginners.

The integration of Java in Android app development also ensures sustained relevance, as mobile computing remains dominant. Additionally, frameworks like Spring Boot simplify building enterprise-grade applications, encouraging new programmers to explore full-stack development paths.

---

Embarking on a basic Java tutorial for beginners opens doors to a versatile and enduring programming language. While the initial learning journey may involve grappling with syntax and object-oriented paradigms, the payoff is a skill set applicable across a myriad of technology sectors. With continued practice and exploration, Java learners can progress from foundational concepts to advanced programming challenges, positioning themselves effectively within today's competitive software development landscape.

## [Basic Java Tutorial For Beginners](#)

**basic java tutorial for beginners: The Java Tutorial** Mary Campione, Kathy Walrath, Alison Huml, 2001 Based on the online version that has become one of the world's most visited programmer documentation sites, this is a remarkably clear, practical, hands-on introduction to the Java 2 Platform. The bonus CD-ROM contains all major versions of the Java Platform.

**basic java tutorial for beginners: The Fast Tutorial to Learn Database Programming Using Python GUI with Access and SQL Server** Vivian Siahaan, Rismon Hasiholan Sianipar, 2020-01-15 This book covers microsoft acces and SQL Server based GUI programming using pyqt. Intentionally designed for various levels of interest and ability of learners, this book is suitable for students, engineers, and even researchers in a variety of disciplines. No advanced programming experience is needed, and only a few school-level programming skill are needed. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a

number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In third chapter, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In fourth chapter, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create dan configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature\_Extraction, which has eight columns: feature\_id (primary key), suspect\_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have VARBINARY(MAX) data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The Police table has six columns: police\_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case\_File. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The Case\_File table has seven columns: case\_file\_id (primary key), suspect\_id (foreign key), police\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

**basic java tutorial for beginners:** *The Quick Tutorial to Learn Database Programming Using Python GUI with MariaDB and PostgreSQL* Vivian Siahaan, Rismon Hasiholan Sianipar, 2020-01-15 In this book, you will create two MariaDB and PostgreSQL driven projects using PyQt. The step-by-step guide in this book is expected to help the reader's confidence to become a programmer who can solve database programming problems. A progressive project is provided to demonstrate how to apply the concepts of MariaDB and PostgreSQL using Python. In second chapter, you will learn PyQt that consists of a number of Python bindings for cross-platform applications that combine all the strengths of Qt and Python. By using PyQt, you can include all Qt libraries in Python code, so you can write GUI applications in Python. In other words, you can use PyQt to access all the features provided by Qt through Python code. Because PyQt depends on the Qt libraries at run time, you need to install PyQt. In third chapter, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In fourth chapter, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In this chapter, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter five, you will create dan configure PotgreSQL database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter six, you will create a table with the name Feature\_Extraction, which has eight columns: feature\_id (primary key), suspect\_id

(foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have a VARCHAR data type (200). You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create two tables, Police and Investigator. The Police table has six columns: police\_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In chapter eight, you will create two tables, Victim and Case\_File. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The Case\_File table has seven columns: case\_file\_id (primary key), suspect\_id (foreign key), police\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

**basic java tutorial for beginners: The Absolute Beginner's Guide to Learn Database Programming Using Python GUI with MariaDB and SQL Server** Vivian Siahaan, Rismon Hasiholan Sianipar, 2020-01-16 This book is designed to introduce programmers to programming and computational thinking through the lens of exploring database. This book offers Python programmers one place to look when they need help guiding to Python as one of the fastest-growing computer languages including Web and Internet applications. This clear and concise introduction to the Python language is aimed at readers who are already familiar with programming in at least one language. This hands-on book introduces the essential topic of coding and the Python computer language to beginners and programmers of all ages. This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of MariaDB and SQL Server databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to MariaDB and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create dan configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature\_Extraction, which has eight columns: feature\_id (primary key), suspect\_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have a VARCHAR data type (200). You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two

tables, Police and Investigator. The Police table has six columns: police\_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In chapter nine, you will create two tables, Victim and Case\_File. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The Case\_File table has seven columns: case\_file\_id (primary key), suspect\_id (foreign key), police\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

**basic java tutorial for beginners: The Absolute Beginner's Guide to Learn Database Programming Using Python GUI with PostgreSQL and SQL Server** Vivian Siahaan, Rismon Hasiholan Sianipar, 2020-01-16 This book is a comprehensive guide to Python as one of the fastest-growing computer languages including Web and Internet applications. This clear and concise introduction to the Python language is aimed at readers who are already familiar with programming in at least one language. This hands-on book introduces the essential topic of coding and the Python computer language to beginners and programmers of all ages. This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of PostgreSQL and SQL Server databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to PostgreSQL and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will get introduction of postgresql. And then, you will learn querying data from the postgresql using Python including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using Python, updating data in postgresql database using Python, calling postgresql stored function using Python, deleting data from a postgresql table using Python, and postgresql Python transaction. In chapter seven, you will create dan configure PotgreSQL database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create a table with the name Feature\_Extraction, which has eight columns: feature\_id (primary key), suspect\_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have a

VARCHAR data type (200). You will also create GUI to display, edit, insert, and delete for this table. In chapter nine, you will create two tables, Police and Investigator. The Police table has six columns: police\_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In chapter ten, you will create two tables, Victim and Case\_File. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The Case\_File table has seven columns: case\_file\_id (primary key), suspect\_id (foreign key), police\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

**basic java tutorial for beginners: The Absolute Beginner's Guide to Learn Python GUI with MySQL and SQL Server Databases** Vivian Siahaan, Rismon Hasiholan Sianipar, 2020-01-16 This hands-on book introduces the essential topic of coding and the Python computer language to beginners and programmers of all ages. This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of MySQL and SQL Server databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to MySQL and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create dan configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature\_Extraction, which has eight columns: feature\_id (primary key), suspect\_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have VARBINARY(MAX) data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The Police table has six columns: police\_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case\_File. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The Case\_File table has seven columns: case\_file\_id (primary key), suspect\_id (foreign key), police\_id



(foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables.

**basic java tutorial for beginners:** *Java Tutorials - Herong's Tutorial Examples* Herong Yang, 2020-10-10 This tutorial book is a collection of notes and sample codes written by the author while he was learning Java himself. Topics covered in the book include: Java language basics; execution environment; generic classes and parameterized type; generic methods and type argument inferences; enum types and constants; lambda expressions; annotation types and invocations; Java modules; execution threads and synchronization; memory management and garbage collection. Updated in 2024 (Version v8.22) with JDK 20. For latest updates and free sample chapters, visit <https://www.herongyang.com/Java>.

**basic java tutorial for beginners:** The Beginner's Guide to Learn Python GUI with MySQL and SQLite Vivian Siahaan, Rismon Hasiholan Sianipar, 2020-01-14 This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of MySQL and SQLite databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to MySQL and SQLite is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create dan configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature\_Extraction, which has eight columns: feature\_id (primary key), suspect\_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have VARBINARY(MAX) data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The Police table has six columns: police\_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case\_File. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The Case\_File table has seven columns: case\_file\_id (primary key), suspect\_id (foreign key), police\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables.

**basic java tutorial for beginners: LEARN FROM SCRATCH VISUAL BASIC .NET WITH MYSQL** Vivian Siahaan, Rismon Hasiholan Sianipar, 2020-11-04 This book will teach you with step-by-step approach to develop from scratch a MySQL-driven desktop application that readers can develop for their own purposes to implement school database project using Visual Basic .NET. In Tutorial 1, you will perform the steps necessary to add 8 tables using phpMyAdmin into School database that you will create. You will build each table and add the associated fields as needed. In this tutorial, you will also build login form and main form. In Tutorial 2, you will build such a form for Parent table. This table has thirteen fields: ParentID, FirstName, LastName, BirthDate, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, PhotoFile, and FingerFile). You need fourteen label controls, two picture boxes, six text boxes, four comboboxes, one check box, one date time picker, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, six buttons for other utilities, one button for searching member's name, one button to upload parent's photo, and button to upload parent's finger. Place these controls on the form. In Tutorial 3, you will build such a form for Student table. This table has fifteen fields: StudentID, ParentID, FirstName, LastName, BirthDate, YearEntry, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, PhotoFile, and FingerFile). You need sixteen label controls, two picture boxes, six text boxes, five comboboxes, one check box, two date time pickers, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for controlling editing features, one button for searching parent's name, one button to open parent form, one button to upload student's photo, and one button to upload student's finger. In Tutorial 4, you will build a form for Teacher table. This table has fifteen fields: TeacherID, RegNumber, FirstName, LastName, BirthDate, Rank, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, PhotoFile, and FingerFile). You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need sixteen label controls, one picture box, seven text boxes, five comboboxes, one check box, one date time picker, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, six buttons for controlling editing features, one button for searching teacher's name, and one button to upload teacher's photo. In Tutorial 5, you will build a form for Subject table. This table has only three fields: SubjectID, Name, and Description. You need four label controls, four text boxes, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for utilities, and one button for searching subject name. Place these controls on the form. You will also build a form for Grade table. This table has seven fields: GradeID, Name, SubjectID, TeacherID, SchoolYear, TimeStart, and TimeFinish. You need to add seven label controls, one text box, four comboboxes, and two date time pickers. You also need four buttons for navigation, seven buttons for controlling editing features, one button to open subject form, and one button to open teacher form. In Tutorial 6, you will build a form for Grade\_Student table. This table has only three fields: Grade\_StudentID, GradeID, and StudentID. You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need two label controls and two comboboxes. You also need four buttons for navigation, seven buttons for controlling editing features, one button to open grade form, and one button to open student form.

**basic java tutorial for beginners: COVID-19: Analisis, Klasifikasi, dan Deteksi Menggunakan Scikit-Learn, Keras, dan TensorFlow dengan Python GUI** Vivian Siahaan, Rismon Hasiholan Sianipar, 2021-09-02 Karena penyebaran COVID-19, pengembangan vaksin dituntut sesegera mungkin. Terlepas dari pentingnya analisis data dalam pengembangan vaksin, tidak banyak dataset sederhana yang dapat ditangani oleh pada analisis data menggunakan data science. Kumpulan data dan kode sampel telah dikumpulkan untuk prediksi epitop Bcell, salah satu topik penelitian utama dalam pengembangan vaksin, tersedia secara gratis. Dataset ini dikembangkan selama proses penelitian dan data yang terkandung di dalamnya diperoleh dari IEDB dan UniProt. Sel B yang menginduksi respon imun spesifik antigen in vivo menghasilkan sejumlah besar antibodi spesifik antigen dengan mengenali subregion (wilayah epitop) protein antigen. Sel B

ini dapat menghambat fungsinya dengan mengikat antibodi ke protein antigen. Memprediksi daerah epitop bermanfaat untuk desain dan pengembangan vaksin yang bertujuan untuk menginduksi produksi antibodi spesifik antigen. Sel B inilah menjadi dataset utama yang dipakai pada proyek ini. Dataset ini memuat kolom: `parent_protein_id`, `protein_seq`, `start_position`, `end_position`, `peptide_seq`, `chou_fasman`, `emini`, `kolaskar_tongaonkar`, `parker`, `hydrophobicity`, `isoelectric_point`, `aromacity`, `stability`, dan `target`. Selanjutnya, Anda akan belajar menggunakan Scikit-Learn, Keras, TensorFlow, NumPy, Pandas, Seaborn, dan sejumlah Pustaka lain untuk memprediksi COVID-19 Epitope menggunakan dataset COVID-19/SARS B-cell Epitope Prediction yang disediakan di Kaggle. Model-model machine learning yang digunakan adalah K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, Support Vector Machine, Adaboost, Gradient Boosting, XGB classifier, dan MLP classifier. Kemudian, Anda akan mempelajari cara menerapkan model deep learning, CNN sekuensial dan VGG16, untuk mendeteksi dan memprediksi Covid-19 X-RAY menggunakan COVID-19 Xray Dataset (Train & Test Sets) yang disediakan di Kaggle. Folder itu sendiri terdiri dari dua subfolder: `test` dan `train`. Terakhir, Anda akan mengembangkan GUI menggunakan PyQt5 untuk menampilkan batas-batas keputusan tiap model, ROC, distribusi fitur, keutamaan fitur, skor validasi silang, nilai-nilai prediksi versus nilai-nilai sebenarnya, matriks confusion, rugi pelatihan, dan rugi akurasi.

**basic java tutorial for beginners: Learn PyQt By Example: A Quick Start Guide to MySQL and SQLite Driven Programming** Vivian Siahaan, Rismon Hasiholan Sianipar, 2019-11-24  
This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of MySQL and SQLite databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to MySQL and SQLite is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create dan configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: `suspect_id` (primary key), `suspect_name`, `birth_date`, `case_date`, `report_date`, `suspect_status`, `arrest_date`, `mother_name`, `address`, `telephone`, and `photo`. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name `Feature_Extraction`, which has eight columns: `feature_id` (primary key), `suspect_id` (foreign key), `feature1`, `feature2`, `feature3`, `feature4`, `feature5`, and `feature6`. The six fields (except keys) will have `VARBINARY(MAX)` data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The Police table has six columns: `police_id` (primary key), `province`, `city`, `address`, `telephone`, and `photo`. The Investigator

table has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case\_File. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The Case\_File table has seven columns: case\_file\_id (primary key), suspect\_id (foreign key), police\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables.

**basic java tutorial for beginners: Learn PyQt The Hard Way: A Quick Start Guide to PostgreSQL and SQLite Driven Programming** Vivian Siahaan, Rismon Hasiholan Sianipar, 2019-11-25 This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of PostgreSQL and SQLite databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to PostgreSQL and SQLite is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six and chapter seven, you will get introduction of postgresql. And then, you will learn querying data from the postgresql using Python including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using Python, updating data in postgresql database using Python, calling postgresql stored function using Python, deleting data from a postgresql table using Python, and postgresql Python transaction. In chapter eight, you will create dan configure PotgreSQL database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter nine, you will create a table with the name Feature\_Extraction, which has eight columns: feature\_id (primary key), suspect\_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have a VARCHAR data type (200). You will also create GUI to display, edit, insert, and delete for this table. In chapter ten, you will create two tables, Police and Investigator. The Police table has six columns: police\_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In chapter eleven, you will create two tables, Victim and Case\_File. The Victim table has nine columns: victim\_id (primary key), victim\_name,

crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The Case\_File table has seven columns: case\_file\_id (primary key), suspect\_id (foreign key), police\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

**basic java tutorial for beginners: Learn JDBC By Example: A Quick Start Guide to MariaDB and SQL Server Driven Programming** Vivian Siahaan, Rismon Hasiholan Sianipar, 2019-11-24 This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of MariaDB and SQL Server databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to MariaDB and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from MariaDB and SQL Server. As you would expect, this book shows how to build from scratch two different databases: MariaDB and SQL Server using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. You will also learn how to create and store salt passwords and verify them. In chapter two, you will create a PostgreSQL database, named Bank, and its tables. In chapter three, you will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter four, you will create an Account table. This account table has the following ten fields: account\_id (primary key), client\_id (primarykey), account\_number, account\_date, account\_type, plain\_balance, cipher\_balance, decipher\_balance, digital\_signature, and signature\_verification. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter five, you create a table named Client\_Data, which has seven columns: client\_data\_id (primary key), account\_id (primary\_key), birth\_date, address, mother\_name, telephone, and photo\_path. In chapter six, you will be taught how to create a SQL Server database, named Crime, and its tables. In chapter seven, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. In chapter nine, you will be taught to create Java GUI to view, edit, insert, and delete Feature\_Extraction table data. This table has eight columns: feature\_id (primary key), suspect\_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter ten, you will add two tables: Police\_Station and Investigator. These two tables will later be joined to Suspect table through another table, File\_Case, which will be built in the seventh chapter. The Police\_Station has six columns: police\_station\_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter eleven, you will add two tables: Victim and File\_Case. The File\_Case table will connect four other tables: Suspect, Police\_Station, Investigator and Victim. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The File\_Case has seven columns: file\_case\_id (primary key), suspect\_id (foreign key), police\_station\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MariaDB/SQL Server programmer.

**basic java tutorial for beginners:** Learn JDBC The Hard Way: A Hands-On Guide to PostgreSQL and SQL Server Driven Programming Vivian Siahaan, Rismon Hasiholan Sianipar, 2019-11-23 This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to PostgreSQL and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from PostgreSQL and SQL Server. As you would expect, this book shows how to build from scratch two different databases: PostgreSQL and SQL Server using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In chapter two, you will learn querying data from the postgresql using jdbc including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using jdbc, updating data in postgresql database using jdbc, calling postgresql stored function using jdbc, deleting data from a postgresql table using jdbc, and postgresql jdbc transaction. In chapter three, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. You will also learn how to create and store salt passwords and verify them. In chapter four, you will create a PostgreSQL database, named Bank, and its tables. In chapter five, you will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter six, you will create an Account table. This account table has the following ten fields: account\_id (primary key), client\_id (primarykey), account\_number, account\_date, account\_type, plain\_balance, cipher\_balance, decipher\_balance, digital\_signature, and signature\_verification. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter seven, you create a table named Client\_Data, which has seven columns: client\_data\_id (primary key), account\_id (primary\_key), birth\_date, address, mother\_name, telephone, and photo\_path. In chapter eight, you will be taught how to create a SQL Server database, named Crime, and its tables. In chapter nine, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter ten, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. In chapter eleven, you will be taught to create Java GUI to view, edit, insert, and delete Feature\_Extraction table data. This table has eight columns: feature\_id (primary key), suspect\_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter twelve, you will add two tables: Police\_Station and Investigator. These two tables will later be joined to Suspect table through another table, File\_Case, which will be built in the seventh chapter. The Police\_Station has six columns: police\_station\_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter thirteen, you will add two tables: Victim and File\_Case. The File\_Case table will connect four other tables: Suspect, Police\_Station, Investigator and Victim. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The File\_Case has seven columns: file\_case\_id (primary key), suspect\_id (foreign key), police\_station\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database

programming skills for every Java/PostgreSQL/SQL Server programmer.

**basic java tutorial for beginners:** Coding For Programmers Using Python: The Step-by-Step Guide to Learn PyQt and Database Applications Vivian Siahaan, Rismon Hasiholan Sianipar, 2019-11-29 This book is a comprehensive guide to Python as one of the fastest-growing computer languages including Web and Internet applications. This clear and concise introduction to the Python language is aimed at readers who are already familiar with programming in at least one language. This hands-on book introduces the essential topic of coding and the Python computer language to beginners and programmers of all ages. This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of PostgreSQL and SQL Server databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to PostgreSQL and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will get introduction of postgresql. And then, you will learn querying data from the postgresql using Python including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using Python, updating data in postgresql database using Python, calling postgresql stored function using Python, deleting data from a postgresql table using Python, and postgresql Python transaction. In chapter seven, you will create dan configure PotgreSQL database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect\_id (primary key), suspect\_name, birth\_date, case\_date, report\_date, suspect\_status, arrest\_date, mother\_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create a table with the name Feature\_Extraction, which has eight columns: feature\_id (primary key), suspect\_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have a VARCHAR data type (200). You will also create GUI to display, edit, insert, and delete for this table. In chapter nine, you will create two tables, Police and Investigator. The Police table has six columns: police\_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator\_id (primary key), investigator\_name, rank, birth\_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In chapter ten, you will create two tables, Victim and Case\_File. The Victim table has nine columns: victim\_id (primary key), victim\_name, crime\_type, birth\_date, crime\_date, gender, address, telephone, and photo. The Case\_File table has seven columns: case\_file\_id (primary key), suspect\_id

(foreign key), police\_id (foreign key), investigator\_id (foreign key), victim\_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

**basic java tutorial for beginners: VISUAL BASIC .NET AND DATABASE: PRACTICAL TUTORIALS** Vivian Siahaan, Rismon Hasiholan Sianipar, 2020-10-31 This book aims to develop a MySQL-driven desktop application that readers can develop for their own purposes to implement library project using Visual Basic .NET. In Tutorial 1, you will build a Visual Basic interface for the database. This interface will be used as the main terminal in accessing other forms. This tutorial will also discuss how to create login form and login table. You will create login form. Place on the form one picture box, two labels, one combo box, one text box, and two buttons. In Tutorial 2, you will build a school inventory project where you can store information about valuables in school. The table will have nine fields: Item (description of the item), Quantity, Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). In Tutorial 3, you will perform the steps necessary to add 5 new tables using phpMyAdmin into Academy database. You will build each table and add the associated fields as needed. Every table in the database will need input form. In this tutorial, you will build such a form for Author table. Although this table is quite simple (only four fields: AuthorID, Name, BirthDate, and PhotoFile), it provides a basis for illustrating the many steps in interface design. SQL statement is required by the Command object to read fields (sorted by Name). Then, you will build an interface so that the user can maintain the Publisher table in the database (Academy). The Publisher table interface is more or less the same as Author table interface. This Publisher table interface only requires more input fields. So you will use the interface for the Author table and modify it for the Publisher table. In Tutorial 4, you will perform the steps necessary to design and implement title form, library member form, and book borrowing form. You start by designing and testing the basic entry form for book titles. The Title table has nine fields: BookTitle, PublishYear, ISBN, PublisherID, AuthorID, Description, Note, Subject, and Comment. Then, you will build such a form for Member table. This table has twelve fields: MemberID, FirstName, LastName, BirthDate, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, and PhotoFile). You need thirteen label controls, one picture box, six text boxes, four combo boxes, one check box, one date time picker, one openFileDialog, and one printpreviewdialog. You also need four buttons for navigation, six buttons for controlling editing features, one button for searching member's name, and one button to upload member's photo. Finally, you will build such a form for Borrow table. This table has seven fields: BorrowID, MemberID, BorrowCode, ISBN, BorrowDate, ReturnDate, and Penalty. In this form, you need fourteen label controls, seven text boxes, two combo boxes, two date time pickers, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for other utilities, one button to generate borrowing code, and one button to return book.

**basic java tutorial for beginners: STROKE: Analysis and Prediction Using Scikit-Learn, Keras, and TensorFlow with Python GUI** Vivian Siahaan, Rismon Hasiholan Sianipar, 2023-07-15 In this project, we will perform an analysis and prediction task on stroke data using machine learning and deep learning techniques. The entire process will be implemented with Python GUI for a user-friendly experience. We start by exploring the stroke dataset, which contains information about various factors related to individuals and their likelihood of experiencing a stroke. We load the dataset and examine its structure, features, and statistical summary. Next, we preprocess the data to ensure its suitability for training machine learning models. This involves handling missing values, encoding categorical variables, and scaling numerical features. We utilize techniques such as data imputation and label encoding. To gain insights from the data, we visualize its distribution and relationships between variables. We create plots such as histograms, scatter plots, and correlation matrices to understand the patterns and correlations in the data. To improve model performance and reduce dimensionality, we select the most relevant features for prediction. We employ techniques such as correlation analysis, feature importance ranking, and domain knowledge to



identify the key predictors of stroke. Before training our models, we split the dataset into training and testing subsets. The training set will be used to train the models, while the testing set will evaluate their performance on unseen data. We construct several machine learning models to predict stroke. These models include Support Vector, Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Gradient Boosting, Light Gradient Boosting, Naive Bayes, Adaboost, and XGBoost. Each model is built and trained using the training dataset. We train each model on the training dataset and evaluate its performance using appropriate metrics such as accuracy, precision, recall, and F1-score. This helps us assess how well the models can predict stroke based on the given features. To optimize the models' performance, we perform hyperparameter tuning using techniques like grid search or randomized search. This involves systematically exploring different combinations of hyperparameters to find the best configuration for each model. After training and tuning the models, we save them to disk using joblib. This allows us to reuse the trained models for future predictions without having to train them again. With the models trained and saved, we move on to implementing the Python GUI. We utilize PyQt libraries to create an interactive graphical user interface that provides a seamless user experience. The GUI consists of various components such as buttons, checkboxes, input fields, and plots. These components allow users to interact with the application, select prediction models, and visualize the results. In addition to the machine learning models, we also implement an ANN using TensorFlow. The ANN is trained on the preprocessed dataset, and its architecture consists of a dense layer with a sigmoid activation function. We train the ANN on the training dataset, monitoring its performance using metrics like loss and accuracy. We visualize the training progress by plotting the loss and accuracy curves over epochs. Once the ANN is trained, we save the model to disk using the h5 format. This allows us to load the trained ANN for future predictions. In the GUI, users have the option to choose the ANN as the prediction model. When selected, the ANN model is loaded from disk, and predictions are made on the testing dataset. The predicted labels are compared with the true labels for evaluation. To assess the accuracy of the ANN predictions, we calculate various evaluation metrics such as accuracy score, precision, recall, and classification report. These metrics provide insights into the ANN's performance in predicting stroke. We create plots to visualize the results of the ANN predictions. These plots include a comparison of the true values and predicted values, as well as a confusion matrix to analyze the classification accuracy. The training history of the ANN, including the loss and accuracy curves over epochs, is plotted and displayed in the GUI. This allows users to understand how the model's performance improved during training. In summary, this project covers the analysis and prediction of stroke using machine learning and deep learning models. It encompasses data exploration, preprocessing, model training, hyperparameter tuning, GUI implementation, ANN training, and prediction visualization. The Python GUI enhances the user experience by providing an interactive and intuitive platform for exploring and predicting stroke based on various features.

**basic java tutorial for beginners: HEPATITIS C: Classification and Prediction Using Scikit-Learn, Keras, and TensorFlow with Python GUI** Vivian Siahaan, Rismon Hasiholan Sianipar, 2023-08-19 In this comprehensive project focusing on Hepatitis C classification and prediction, the journey begins with a meticulous exploration of the dataset. Through Python, Scikit-Learn, Keras, and TensorFlow, the project aims to develop an effective model to predict Hepatitis C based on given features. The dataset's attributes are systematically examined, and their distributions are analyzed to uncover insights into potential correlations and patterns. The subsequent step involves categorizing the feature distributions. This phase sheds light on the underlying characteristics of each attribute, facilitating the understanding of their roles in influencing the target variable. This categorization lays the foundation for feature scaling and preprocessing, ensuring that the data is optimized for machine learning. The core of the project revolves around the development of machine learning models. Employing Scikit-Learn, various classification algorithms are applied, including K-Nearest Neighbors (KNN), Decision Trees, Random Forests, Naive Bayes, Gradient Boosting, AdaBoost, Light Gradient Boosting, Multi-Layer

Perceptron, and XGBoost. The models are fine-tuned using Grid Search to optimize hyperparameters, enhancing their performance and generalization capability. Taking the project a step further, deep learning techniques are harnessed to tackle the Hepatitis C classification challenge. A key component is the construction of an Artificial Neural Network (ANN) using Keras and TensorFlow. This ANN leverages layers of interconnected nodes to learn complex patterns within the data. LSTM, FNN, RNN, DBN, and Autoencoders are also explored, offering a comprehensive understanding of deep learning's versatility. To evaluate the models' performances, an array of metrics are meticulously employed. Metrics such as accuracy, precision, recall, F1-score, and AUC-ROC are meticulously calculated. The significance of each metric is meticulously explained, underpinning the assessment of a model's true predictive power and its potential weaknesses. The evaluation phase emerges as a pivotal aspect, accentuated by an array of comprehensive metrics. Performance assessment encompasses metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Cross-validation and learning curves are strategically employed to mitigate overfitting and ensure model generalization. Furthermore, visual aids such as ROC curves and confusion matrices provide a lucid depiction of the models' interplay between sensitivity and specificity. The culmination of the project involves the creation of a user-friendly Graphical User Interface (GUI) using PyQt. The GUI enables users to interact seamlessly with the models, facilitating data input, model selection, and prediction execution. A detailed description of the GUI's components, including buttons, checkboxes, and interactive plots, highlights its role in simplifying the entire classification process. In a comprehensive journey of exploration, experimentation, and analysis, this project effectively marries data science and machine learning. By thoroughly examining the dataset, engineering features, utilizing a diverse range of machine learning models, harnessing the capabilities of deep learning, evaluating performance metrics, and creating an intuitive GUI, the project encapsulates the multi-faceted nature of modern data-driven endeavors.

**basic java tutorial for beginners: Artificial Intelligence in Education** Chee-Kit Looi, 2005  
The field of Artificial Intelligence in Education includes research and researchers from many areas of technology and social science. This study aims to open opportunities for the cross-fertilization of information and ideas from researchers in the many fields that make up this interdisciplinary research area.

**basic java tutorial for beginners: COVID-19: Analysis, Classification, and Detection Using Scikit-Learn, Keras, and TensorFlow with Python GUI** Vivian Siahaan, Rismon Hasiholan Sianipar, 2023-08-11  
In this comprehensive project, COVID-19: Analysis, Classification, and Detection Using Scikit-Learn, Keras, and TensorFlow with Python GUI, the primary objective is to leverage various machine learning and deep learning techniques to analyze and classify COVID-19 cases based on numerical data and medical image data. The project begins by exploring the dataset, gaining insights into its structure and content. This initial data exploration aids in understanding the distribution of categorized features, providing valuable context for subsequent analysis. With insights gained from data exploration, the project delves into predictive modeling using machine learning. It employs Scikit-Learn to build and fine-tune predictive models, harnessing grid search for hyperparameter optimization. This meticulous process ensures that the machine learning models, such as Naïve Bayes, K-Nearest Neighbors, Decision Trees, Random Forests, Gradient Boosting, Extreme Gradient Boosting, Multi-Layer Perceptron, AdaBoost, and Logistic Regression, are optimized to accurately predict the risk of COVID-19 based on the input features. Transitioning to the realm of deep learning, the project employs Convolutional Neural Networks (CNNs) to perform intricate image classification tasks. Leveraging Keras and TensorFlow, the CNN architecture is meticulously crafted, comprising convolutional and pooling layers, dropout regularization, and dense layers. The project also extends its deep learning capabilities by utilizing the VGG16 pre-trained model, harnessing its powerful feature extraction capabilities for COVID-19 image classification. To gauge the effectiveness of the trained models, an array of performance metrics is utilized. In this project, a range of metrics are used to evaluate the performance of machine learning and deep learning models employed for COVID-19 classification. These metrics include Accuracy, which

measures the overall correctness of predictions; Precision, emphasizing the accuracy of positive predictions; Recall (Sensitivity), assessing the model's ability to identify positive instances; and F1-Score, a balanced measure of accuracy. The Mean Squared Error (MSE) quantifies the magnitude of errors in regression tasks, while the Confusion Matrix summarizes classification results by showing counts of true positives, true negatives, false positives, and false negatives. These metrics together provide a comprehensive understanding of model performance. They help gauge the model's accuracy, the balance between precision and recall, and its proficiency in classifying both positive and negative instances. In the medical context of COVID-19 classification, these metrics play a vital role in evaluating the models' reliability and effectiveness in real-world applications. The project further enriches its analytical capabilities by developing an interactive Python GUI. This graphical user interface streamlines the user experience, facilitating data input, model training, and prediction. Users are empowered to input medical images for classification, leveraging the trained machine learning and deep learning models to assess COVID-19 risk. The culmination of the project lies in the accurate prediction of COVID-19 risk through a combined approach of machine learning and deep learning techniques. The Python GUI using PyQt5 provides a user-friendly platform for clinicians and researchers to interact with the models, fostering informed decision-making based on reliable and data-driven predictions. In conclusion, this project represents a comprehensive endeavor to harness the power of machine learning and deep learning for the vital task of COVID-19 classification. Through rigorous data exploration, model training, and performance evaluation, the project yields a robust framework for risk prediction, contributing to the broader efforts to combat the ongoing pandemic.

## Related to basic java tutorial for beginners

10 Basic 13 vb vb 10 Basic Basic Basic - Basic Basic “ BASIC ” -- Edsger Wybe Di base basic basis ? - basic base basis APP basis 120/ GBASIC BASIC - BASIC BASIC language BASIC Pascal BASIC BASIC Microsoft BASIC 1,723 ICT ICT - ICT Information and Communications Technology ICT=IT+CT windows7 - 99% Windows windows10 ipgw.neu.edu.cn 10 Basic - 13 vb vb 10 Basic Basic Basic - Basic Basic “ BASIC ” -- Edsger Wybe Di base basic basis ? - basic base basis APP basis 120/ GBASIC BASIC

**BASIC** - BASIC language BASIC Pascal BASIC Microsoft BASIC [ ] 1,723 ICT ICT Information and Communications Technology ICT=IT+CT windows7 99% Windows windows10 windows10 ipgw.neu.edu.cn 10 Basic 13 vb vb 10 Basic Basic - Basic “ BASIC ” -- Edsger Wybe Di base basic basis? - basic base basis APP basis GBasic 120/ GBASIC BASIC - BASIC language BASIC Pascal BASIC Microsoft BASIC [ ] 1,723 ICT ICT Information and Communications Technology ICT=IT+CT windows7 99% Windows windows10 windows10 ipgw.neu.edu.cn 10 Basic 13 vb vb 10 Basic Basic - Basic “ BASIC ” -- Edsger Wybe Di base basic basis? - basic base basis APP basis GBasic 120/ GBASIC BASIC - BASIC language BASIC Pascal BASIC Microsoft BASIC [ ] 1,723 ICT ICT Information and Communications Technology ICT=IT+CT windows7 99% Windows windows10 windows10 ipgw.neu.edu.cn

## Related to basic java tutorial for beginners

**Java tutorial for beginners, part 2** (Android Authority9y) In part 1 of this series, Gary uploaded an awesome post and video explaining the basics of Java programming. Java is the official programming language of Android; so if you want to build fully-fledged

**Java tutorial for beginners, part 2** (Android Authority9y) In part 1 of this series, Gary uploaded an awesome post and video explaining the basics of Java programming. Java is the official programming language of Android; so if you want to build fully-fledged

**Java beginner course - A free and comprehensive guide to the basics of Java** (Android Authority4y) Java is one of the most highly sought after programming languages, not to mention one of the two official languages for Android development. In this Java beginner course, we'll go over the basics to

**Java beginner course - A free and comprehensive guide to the basics of Java** (Android Authority4y) Java is one of the most highly sought after programming languages, not to mention one of the two official languages for Android development. In this Java beginner course, we'll go over the basics to

**Best-Rated YouTube Channels for Coding: Beginners' Guide** (Analytics Insight7d) Learn coding basics through structured tutorials on Python, JavaScript, and web development with beginner-friendly

**Best-Rated YouTube Channels for Coding: Beginners' Guide** (Analytics Insight7d) Learn coding basics through structured tutorials on Python, JavaScript, and web development with beginner-friendly

## Related Articles

- [forensic science final exam study guide answers](#)
- [strategies for successful writing](#)
- [beyond the rim s fowler wright](#)

[Back to Home](#)