# Introduction to Programming!

- Simple Introduction to programming
- We will learn about , What, Why, How, of Programming

# What is Programming?

- Today, we're going to learn about something super cool called "programming." Imagine you have a robot friend, and you want to tell it what to do. That's what programming is all about!

# Why We Need Programming!

- Programming is like giving instructions to a computer so it can do different tasks. It's like talking to a computer using special words and rules that it understands.
- Think of it as giving directions to your friend to build a LEGO castle. You tell them step by step what pieces to use and how to put them together.
- Computers are like super fast and smart helpers, but they need clear and precise instructions, just like how you need to be clear when telling your friend how to build that LEGO castle

# Why We Need Programming

- We use programming to make video games, design websites, create apps for our phones, and even control robots like self-driving cars.
- Imagine telling your robot friend to make you a sandwich. You'd need to give it step-by-step instructions, like "pick up the bread," "spread the peanut butter," and so on. Similarly, we program computers to follow our instructions and do tasks for us.
- Without programming, computers wouldn't know what to do. It's like trying to play a new game without reading the rules first!

# How Do We Program?

- Okay, now you might be wondering, how do we actually talk to computers? Well, we use special languages that computers understand. These are called "programming languages."

- **Examples:** Some programming languages are Python, Scratch, and JavaScript. These languages have their own words and grammar, just like you have rules for writing sentences.

- It's a bit like teaching your pet a new trick. You use words and actions they understand to make them do the trick. In the same way, we use programming languages to communicate with computers.

- But don't worry, you don't need to learn all the languages at once. Just like you start with simple words and sentences, we start with basic instructions in programming.

# How Does a Computer Understand Programs?

- Good question! Computers are smart, but they need things explained in a special way. Let's talk about how they understand the instructions we give them.
- Computers only understand binary language, which is a series of 0s and 1s. It's like their own secret code.
- When we write a program in a programming language, like Scratch or Python, we're actually using words and symbols that humans can understand. But computers can't read those directly.
- So, we have special programs called "compilers" or "interpreters" that translate our human-friendly code into the computer's secret binary code. It's like having a translator to help your robot friend understand you!
- Once the computer has the instructions in its secret code, it follows them step by step, just like how you follow a recipe to bake cookies.

# Types of Programming Languages

- There are different types of programming languages, like different tools for different jobs.
- **High-level languages:** These are like everyday languages that are easier for us to understand, like Python or Java. They're great for writing big programs quickly.
- **Low-level languages:** These are closer to the computer's secret binary code. They're used to make programs run really fast, but they're harder for humans to work with.
- **Visual languages:** These let you build programs by dragging and dropping blocks, like Scratch. They're awesome for beginners and help you see your program as you create it.
- **Domain-specific languages:** These are special languages for specific tasks, like making websites or working with data.
- Just like you choose the right tool for building with LEGO, programmers choose the right language for their job.

# Why Choose Python?

- Imagine if programming was like talking to your friend. Well, that's how Python works! Its code looks a lot like regular English.
- **Simple Syntax:** Python's way of writing is super easy to understand. You don't need to learn lots of confusing symbols or rules.
- **Readability:** If you can read a sentence, you can read Python code. It's like magic words that computers understand.
- You know those times when you just want to start doing cool stuff? Python is the perfect buddy for that.
- **Less Code:** In some other languages, you might need lots of lines to do something simple. With Python, you can get things done with fewer lines.
- **Faster Learning Curve:** Python's simplicity means you'll spend less time figuring out complex rules and more time creating awesome things.
- Imagine if you had a backpack full of tools that make your work super easy. Python is that magical backpack for programmers!

# Why Choose Python?

- **Builtin Functions:** Python comes with tons of pre-made tools, like functions, that you can use right away. No need to build everything from scratch.
- **Libraries:** It's like borrowing books from a library. Python has libraries with special powers for things like math, graphics, and data analysis.
- You're not alone in this coding adventure. Python has a community that's always ready to help out and show you the ropes.
- **Supportive People:** If you get stuck, you can ask questions online or join forums. Experienced programmers are happy to guide you.
- **Tons of Resources:** There are tutorials, guides, and projects online that'll help you learn Python step by step.
- Picture this: your superpower works in different cities, countries, and planets. That's Python's secret!
- **Versatility:** Python isn't just for one type of job. It's used in web development, data science, games, and even space exploration.
- **Cross-Platform:** Whether you're using Windows, Mac, or Linux, Python works like a charm on all major platforms.

# What is a IDE?

- Imagine you're an artist. You need a special studio with all your art supplies in one place. An IDE is like that studio, but for coding!
- **IDE:** Stands for "Integrated Development Environment." It's a fancy name for a special software that helps you write and create code.
- **All-in-One:** Just like your art studio has paints, brushes, and an easel, an IDE has tools to write code, check for errors, and see your results.

# Parts of IDE

- Think of coding like creating a painting. You use colors to make a picture. In coding, you use words and symbols to make a program.
- **Code Editor:** Inside an IDE, you have a space called the code editor. This is where you write your instructions for the computer.
- Artists sometimes make mistakes while painting, right? Similarly, coders make errors too. The IDE helps you find them!
- **Error Messages:** When you make a mistake in your code, the IDE shows you a message that helps you figure out what's wrong.
- Now, when artists finish a painting, they want to see how it looks. Coders want to see how their program works!
- **Run Button:** The IDE has a button that you click, like magic. It makes your code run, and you see what your program does.

# Parts of IDE-2

- Artists organize their paints and brushes neatly. Coders organize their files, folders, and projects in the IDE.
- **Project Manager:** The IDE keeps everything in order. You can open different projects, like opening different art pieces in your studio.
- Most common IDE-
- Jupyter Notebooks, VS code, Spyder, Colab Notebooks