# **Test-Driven Development (TDD) – Explained Simply**

### **🔄 TDD Process (Red → Green → Refactor)**

1️⃣ **🟥 Write a Test**

Define a test for a new function or feature — even before writing the code.  
 ✏️ *Example: “Check if login returns success with correct password.”*

2️⃣ **❌ Run the Test**

It fails, as the code doesn’t exist yet — that’s expected!

3️⃣ **🟩 Write the Code**

Just enough code to make the test pass. No extra fluff!

4️⃣ **✅ Run the Tests Again**

Confirm that the test now passes.

5️⃣ **🔁 Refactor the Code**

Clean up, simplify, and organize the code without changing what it does.

6️⃣ **🔂 Repeat**

Continue this cycle for each small feature or change.

### **🌟 Benefits of TDD**

✅ **Fewer Bugs** – Issues are caught early  
 ✅ **Better Code Quality** – Clean, modular, testable  
 ✅ **Confidence in Changes** – Easily detect breaking changes  
 ✅ **Faster Debugging** – Failures point to specific code  
 ✅ **Encourages Simplicity** – Only write what is necessary

### **📚 TDD Improves Code Readability**

TDD leads to **clean, well-structured code** because developers write only what’s needed and refactor regularly.  
 Tests also act as **live documentation**, helping others understand what the code should do.