# Property Based Testing

# What is property based testing

Property-Based Testing is a testing methodology that generates random inputs to verify that certain properties always hold true.

Instead of writing individual test cases, you define general properties that your system should always satisfy.

These properties guide test generation, covering edge cases and unexpected scenarios that traditional testing might miss.

# Key benefits

Higher Coverage

Edge Case Discovery

Reduced Test Maintenance

# Classic Unit Tests vs. Property-Based Tests

| Feature | Classic Unit Tests | Property-Based Tests |
| --- | --- | --- |
| Approach | Test specific cases with predefined inputs and expected outputs. | Test properties and invariants by generating multiple random inputs. |
| Scope | Focus on example-based testing. | Focus on general properties of the system. |
| Use cases | Best for well-defined scenarios and business logic. | Useful for testing algorithms, mathematical functions, and systems with broad input domains. |

# Coverage Ecosystem

Classic unit tests

Property based tests

E2E

# How it works

- **Defining Properties Instead of Examples**
  - Instead of writing individual test cases (e.g., f(2) == 4 and f(5) == 10), you define general properties that must always hold true.

- **Generating Randomized Test Inputs**
  - The testing framework automatically generates a wide range of random inputs, covering more cases than manually written tests.
  - This helps discover edge cases that would be hard to anticipate manually.

- **Running Tests with Multiple Inputs**
  - The property is tested against hundreds or thousands of generated inputs.
  - If all tests pass, the property holds; if one fails, it means the property does not always apply.

# Pratical example

# Q&A

# That’s all folks

Doc hypothesis

Github repo