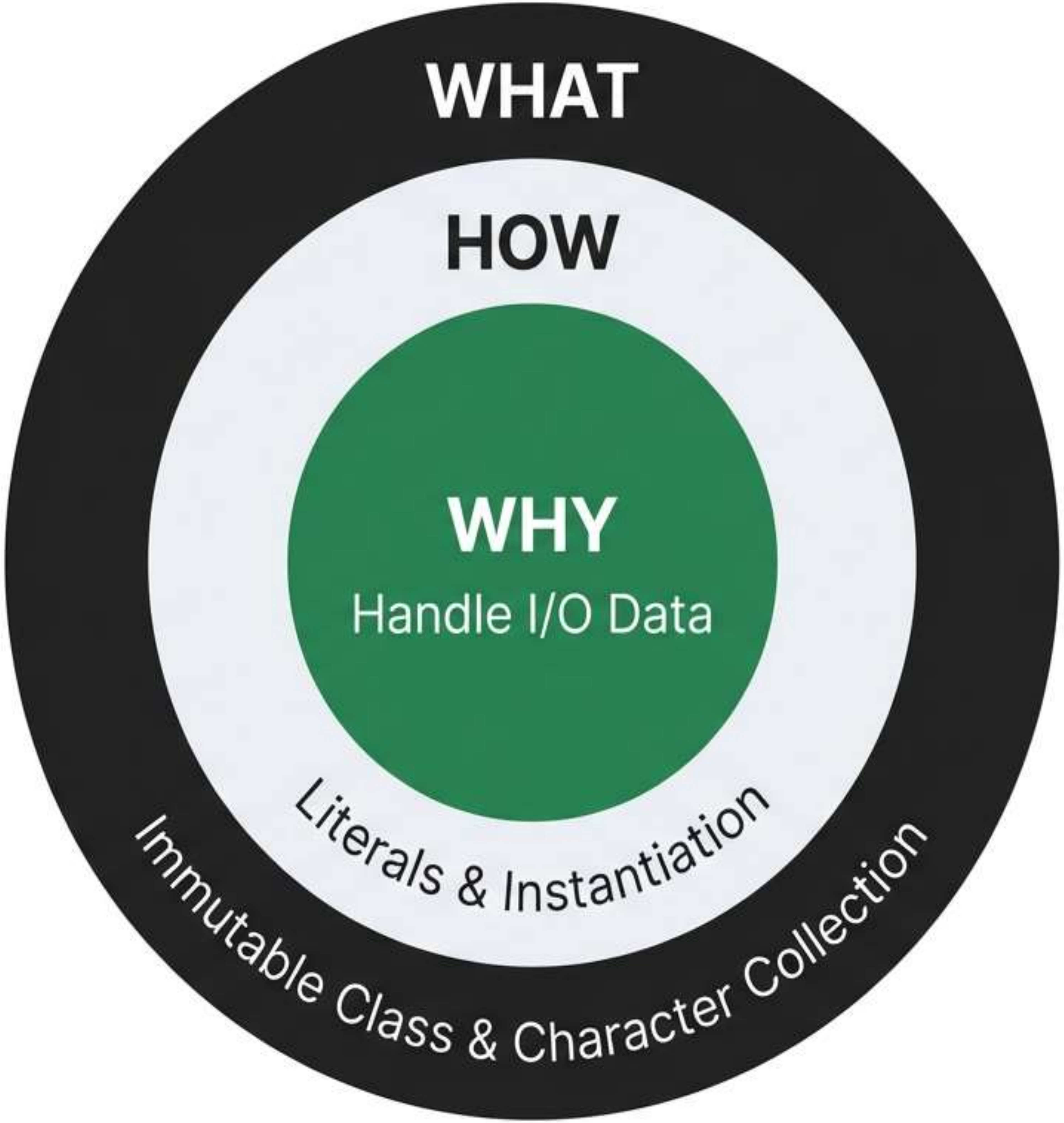
Testleaf
Always Ahead

Mastering Java Strings

The Core of Selenium Automation

From Memory Management to Manipulation Logic

The Golden Circle of Strings



WHAT

HOW

WHY

Handle I/O Data

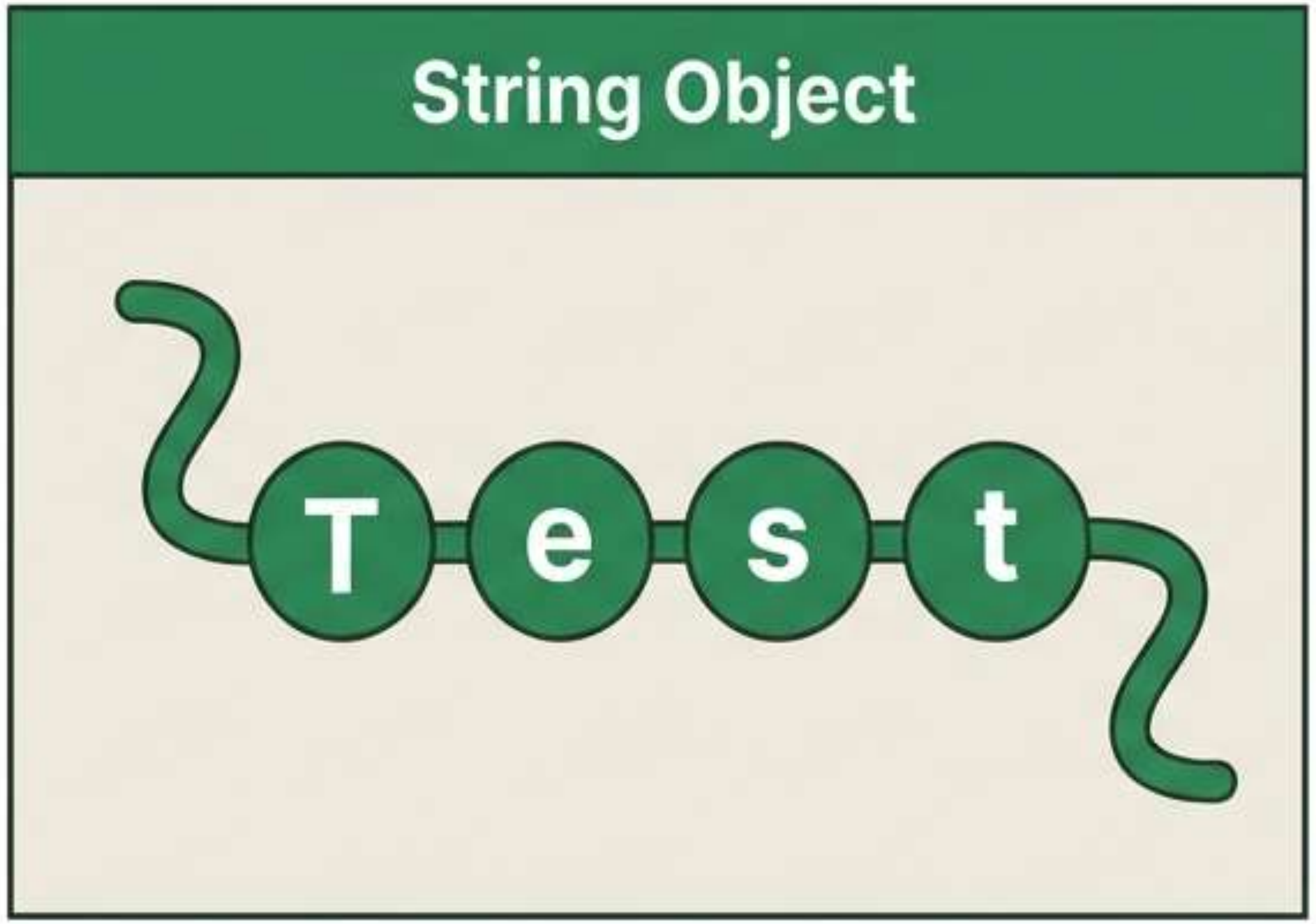
Literals & Instantiation

Immutable Class & Character Collection

What is a String

A String is a non-primitive datatype and a Class in Java.

Key Property: IMMUTABLE. Once created, it cannot be changed.



Collection of Characters

Two ways to create String

Path 1: String Literal

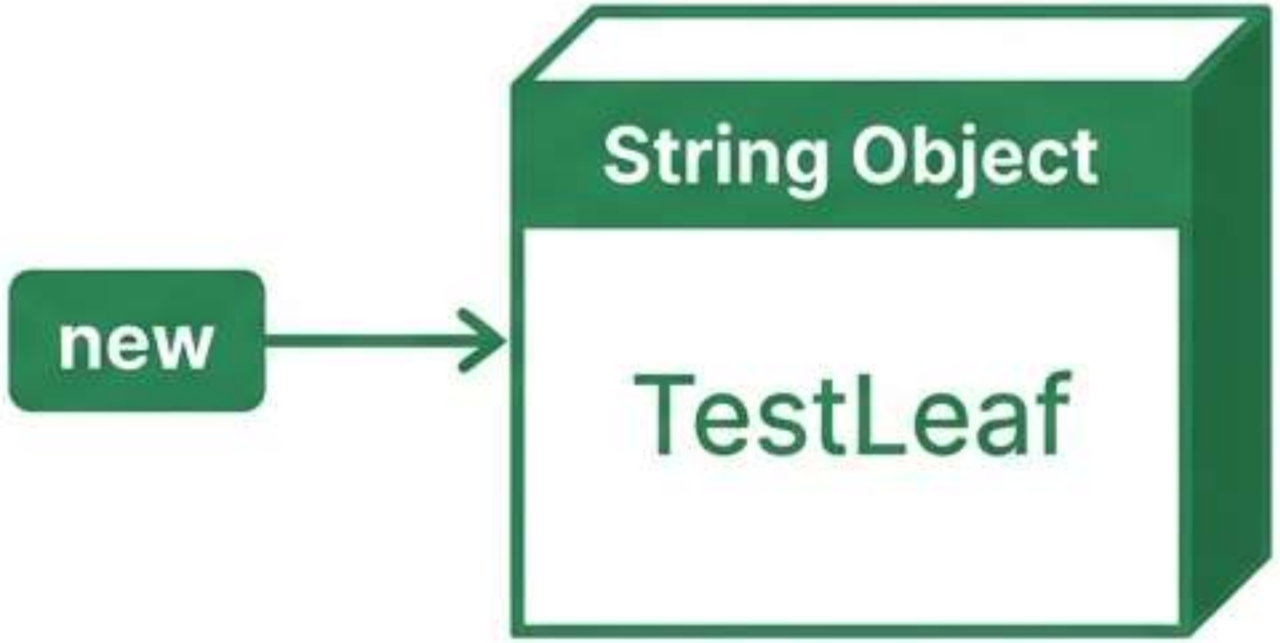
```java
String name = "TestLeaf";
```

Uses double quotes.

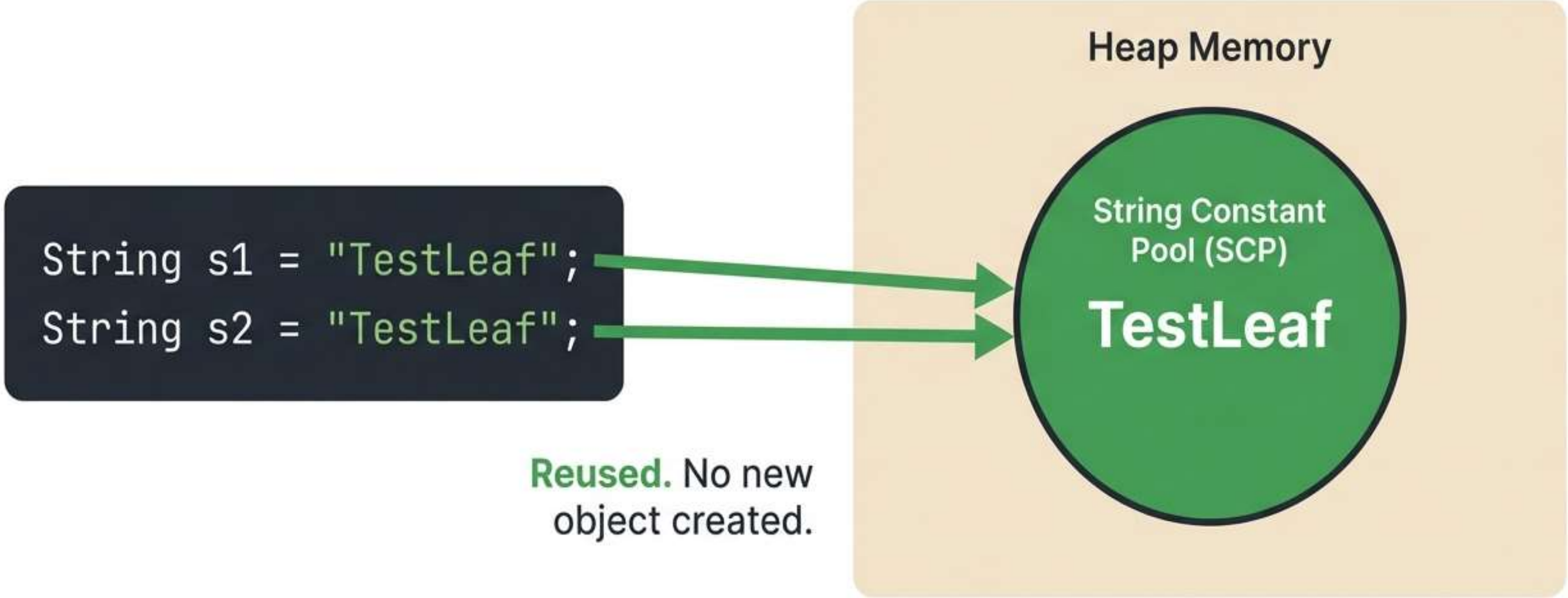
Path 2: Instantiation

```java
String name = new String("TestLeaf");
```

Uses the ‘new’ keyword to force object creation.



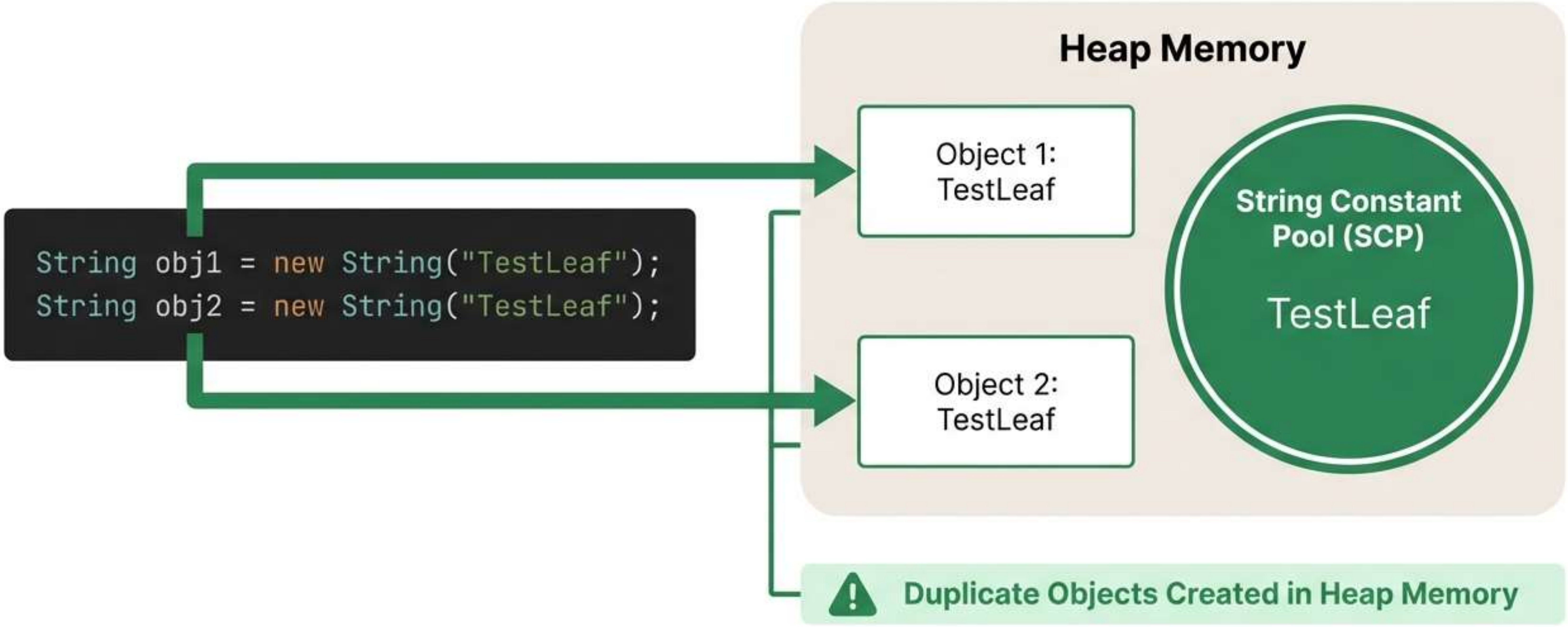
The String Constant Pool (Efficiency,



```
String s1 = "TestLeaf";
String s2 = "TestLeaf";
```

Reused. No new object created.

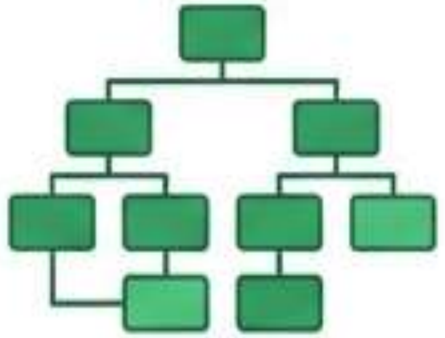
The Heap Memory (Redundancy)



```
String obj1 = new String("TestLeaf");
String obj2 = new String("TestLeaf");
```

Duplicate Objects Created in Heap Memory

Literal vs. Object: The Memory Showdown

	String Literal	String Object
Syntax	`""`	`new String()`
Storage	String Constant Pool (efficient) 	Heap Memory (redundant) 
Behavior	Reuses existing objects	Always creates new object

"Literal and Object differs based on the memory storage."

The Tester’s Toolkit: Validating Content

`equals()`

Compares exact content. Case Sensitive.

`equalsIgnoreCase()`

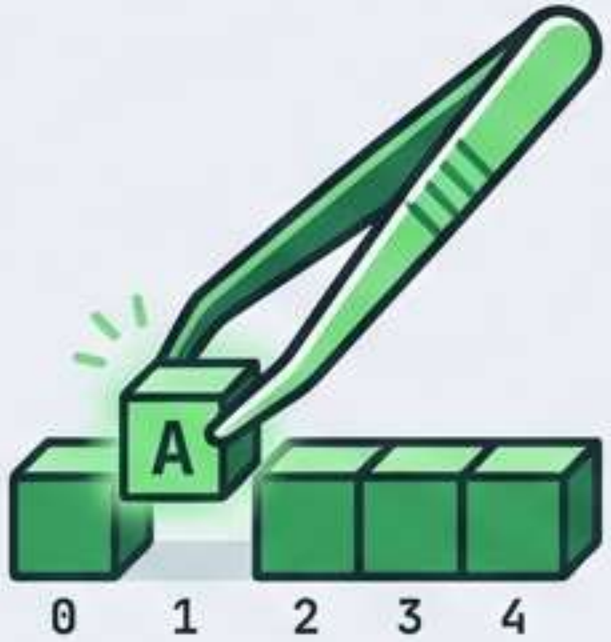
Compares content ignoring case (e.g., 'Pass' vs 'pass').

`contains()`

Checks if a sequence exists within the string.

The Tester’s Toolkit: Extracting & Analyzing

charAt(index)

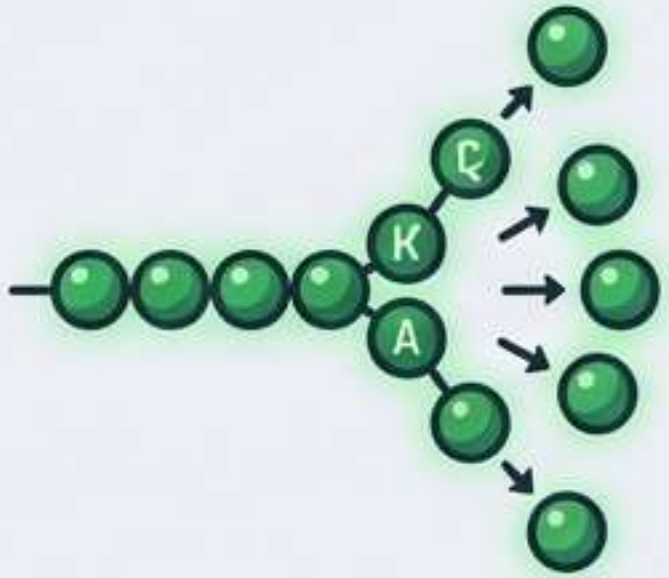


Returns character at specific position.

subString()

Retrieves a sequence based on index.

toCharArray()



Converts String into an array of characters. Vital for loops.

The Tester’s Toolkit: Transforming Data

toLowerCase()

ABC → abc

Converts all characters to lowercase.

toUpperCase()

abc → ABC

Converts all characters to uppercase.

split()

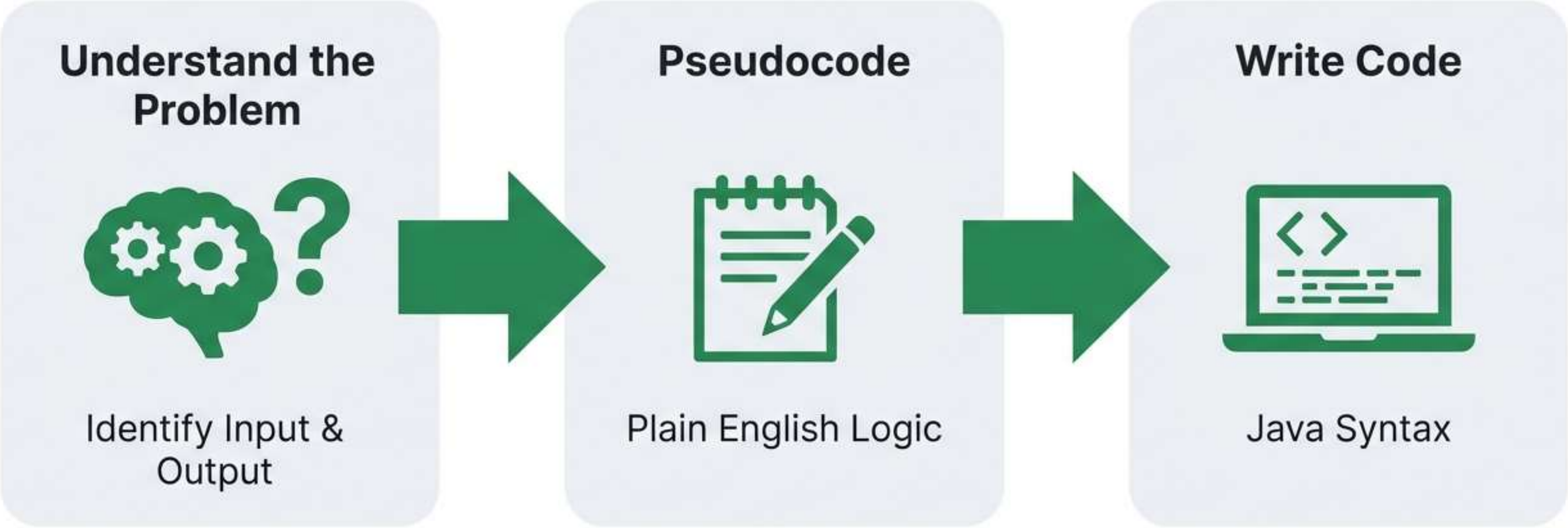
Splits a string into an array of substrings.

replaceAll()

Price: $99 ⟳ Price: 99

Replaces all occurrences of a specified sequence.

The 3-Step Problem Solving Framework



Understand the Problem

Identify Input & Output

Pseudocode

Plain English Logic

Write Code

Java Syntax

Before writing the code – follow the 3 step process

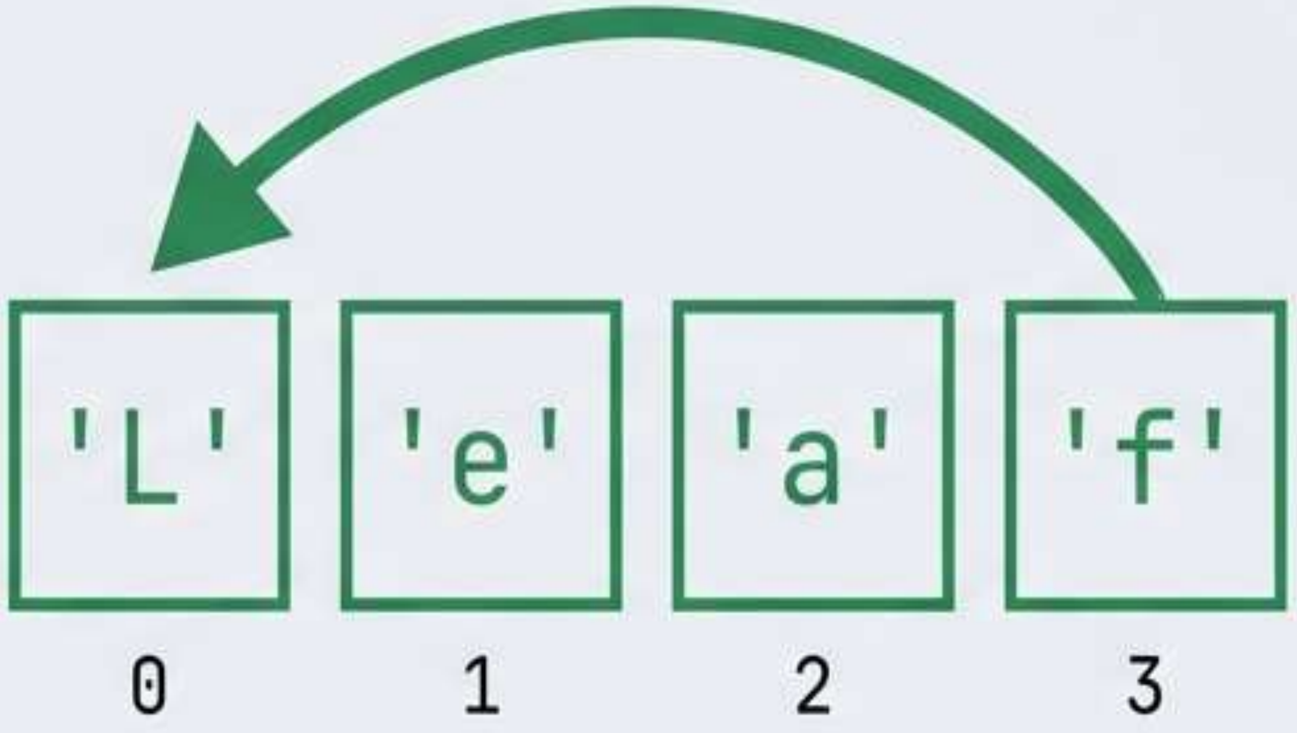
Applied Logic: Reversing a String

Testleaf
Always Ahead

The Problem

- **Goal:** Print characters backwards.

The Logic Visualization



Pseudocode

1. Convert String to Char Array (`toCharArray`)
2. Iterate Loop backwards (from `length-1` to `0`)
3. Print each character

Summary & Best Practices

- ✓ Strings are immutable objects.
- ✓ Prefer Literals for memory efficiency (String Constant Pool).
- ✓ Master the toolkits: Validation, Extraction, Transformation.
- ✓ Always plan logic (Pseudocode) before coding.