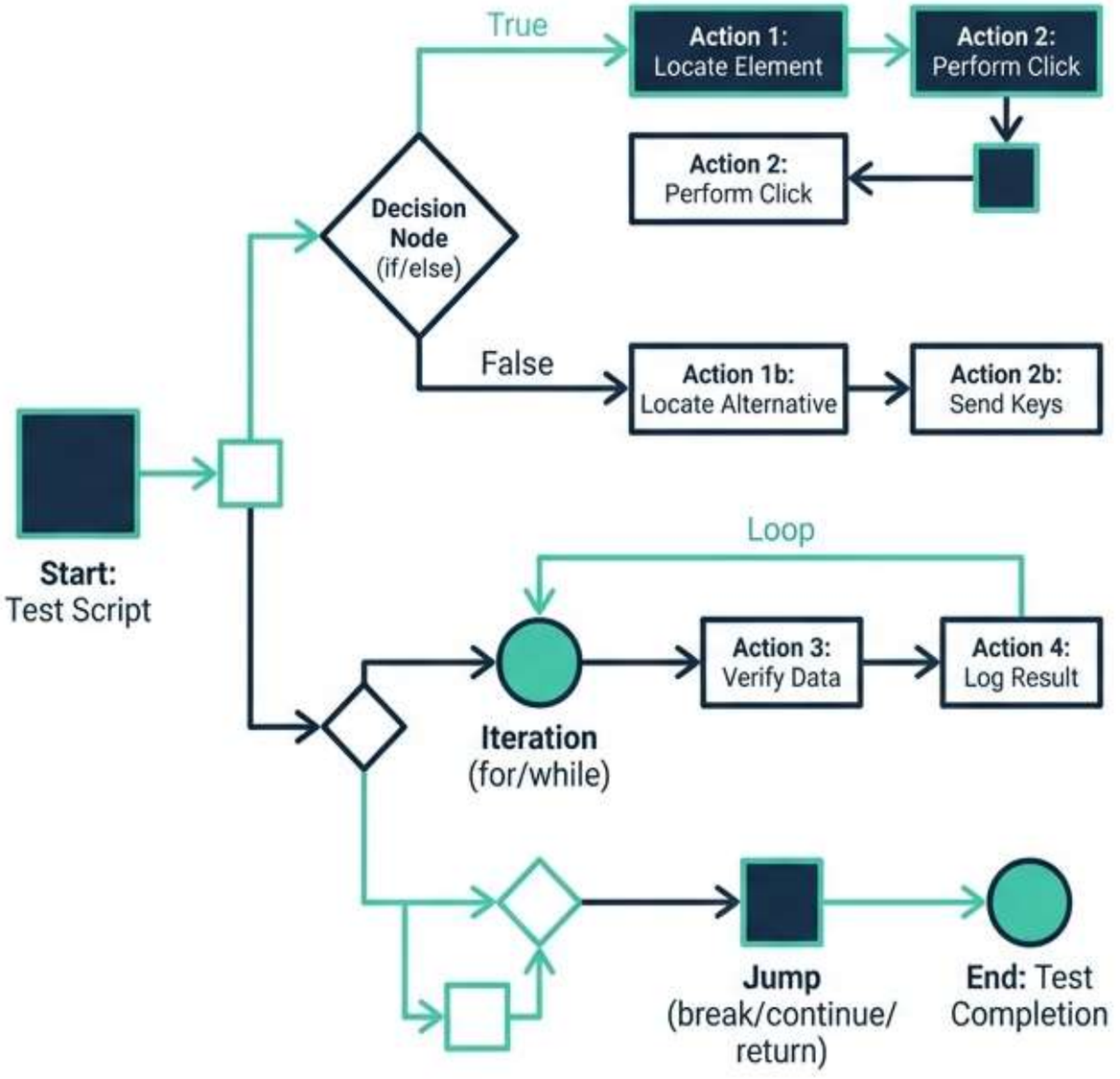
Testleaf
Always Ahead

Master Control Statements in Selenium WebDriver

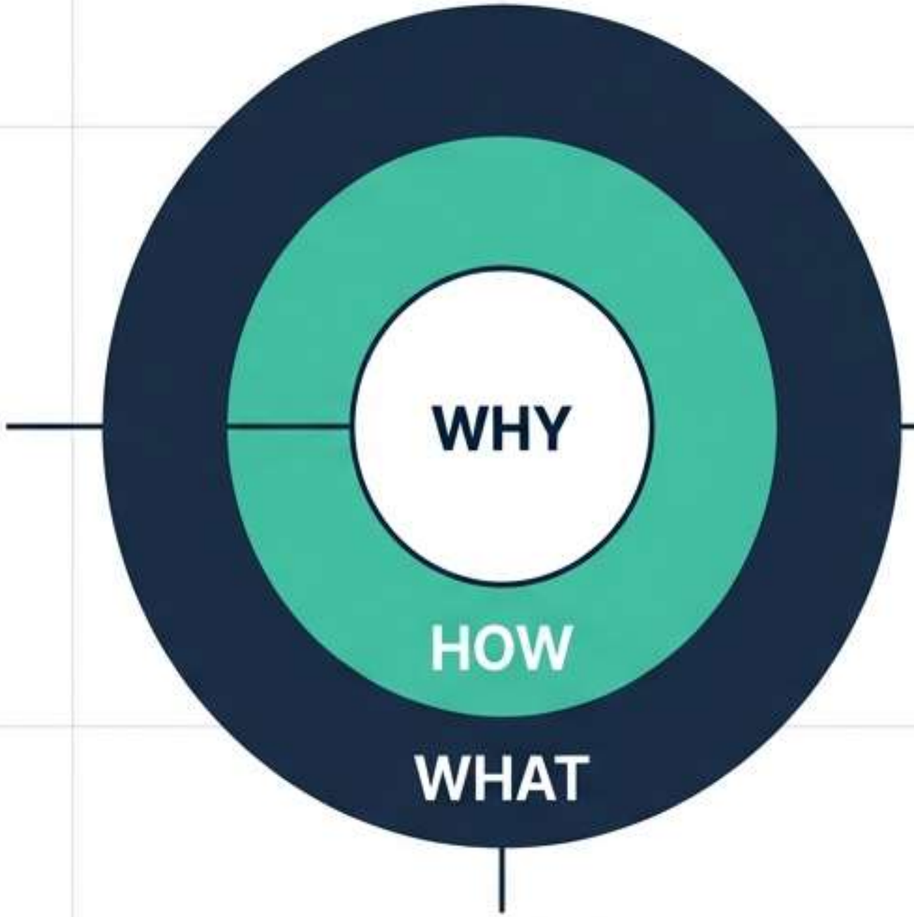
Optimizing Test Automation Logic through Selection, Iteration, and Jump Statements

Optimizing Test Automation Logic through Selection, Iteration, and Jump Statements



Corporate Training Series | Automation Fundamentals

The Golden Circle: Understanding Control Statements



To manage the flow of execution in the program. Without controls, scripts are linear and brittle.

By implementing Selection, Iteration, and Jump Statements.

Statements that define the flow of the program, moving beyond linear execution.

The Three Pillars of Control Flow

Selection (Decision)

Executes code blocks based on boolean criteria.

```
if
if-else
switch-case
```

Iteration (Loops)

Repeats a block of code until a condition is met.

```
for
for-each
while
do-while
```

Jump (Transfer)

Unconditionally transfers control to another part of the script.

```
break
continue
```

Deep Dive into Selection Statements

Testleaf
Always Ahead

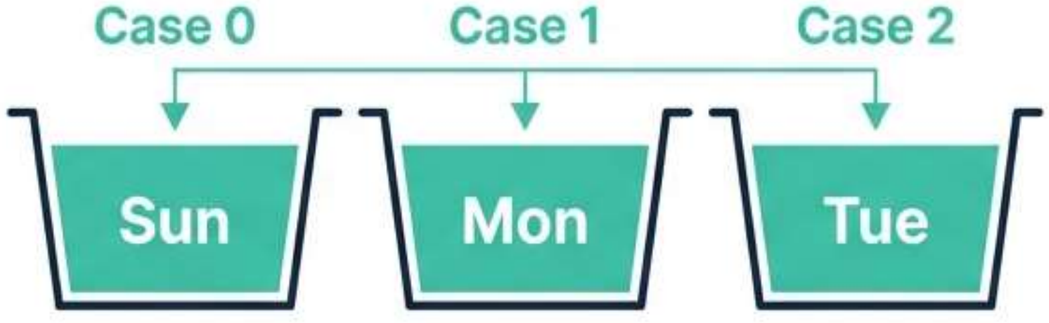
If / Else Logic

Best for binary checks or range-based conditions


```
if (income < 8925) {
  rate = 0.10;
}
```

Switch / Case Logic

Best for multiple specific value checks (Fixed Data)

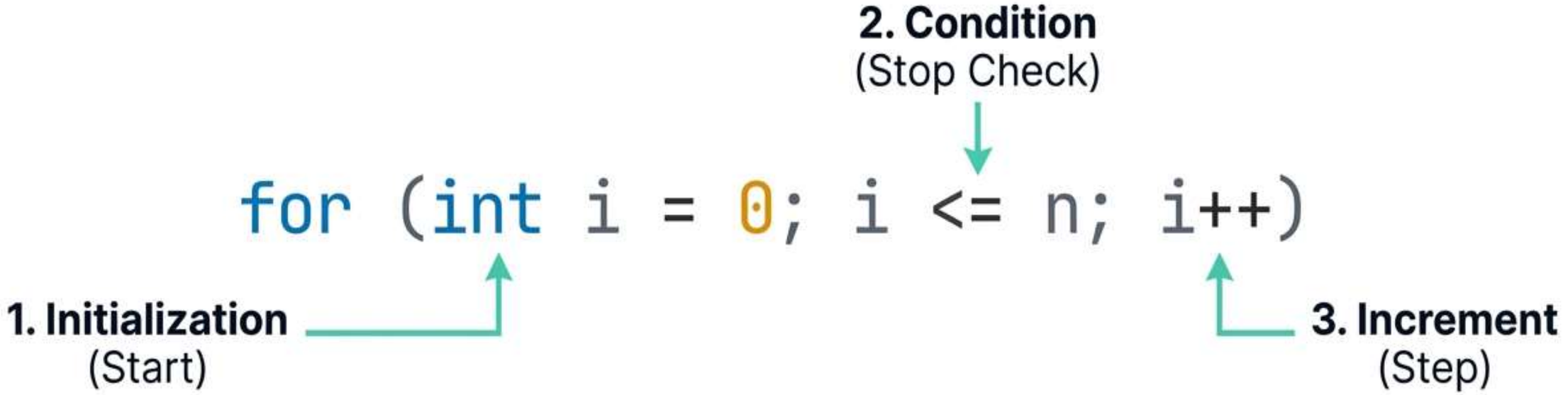



```
switch (day) {
  case 0: print('Sun'); break;
}
```

Critical Distinction: Use 'if-else' for boolean complexity; use 'switch' for readable fixed-value selection

Testleaf
Always Ahead

Deep Dive into Iteration & Loops



For Loop

Used when iteration count is known (e.g., table rows).

While / Do-While

Used when count is dynamic (e.g., waiting for element).

For-Each

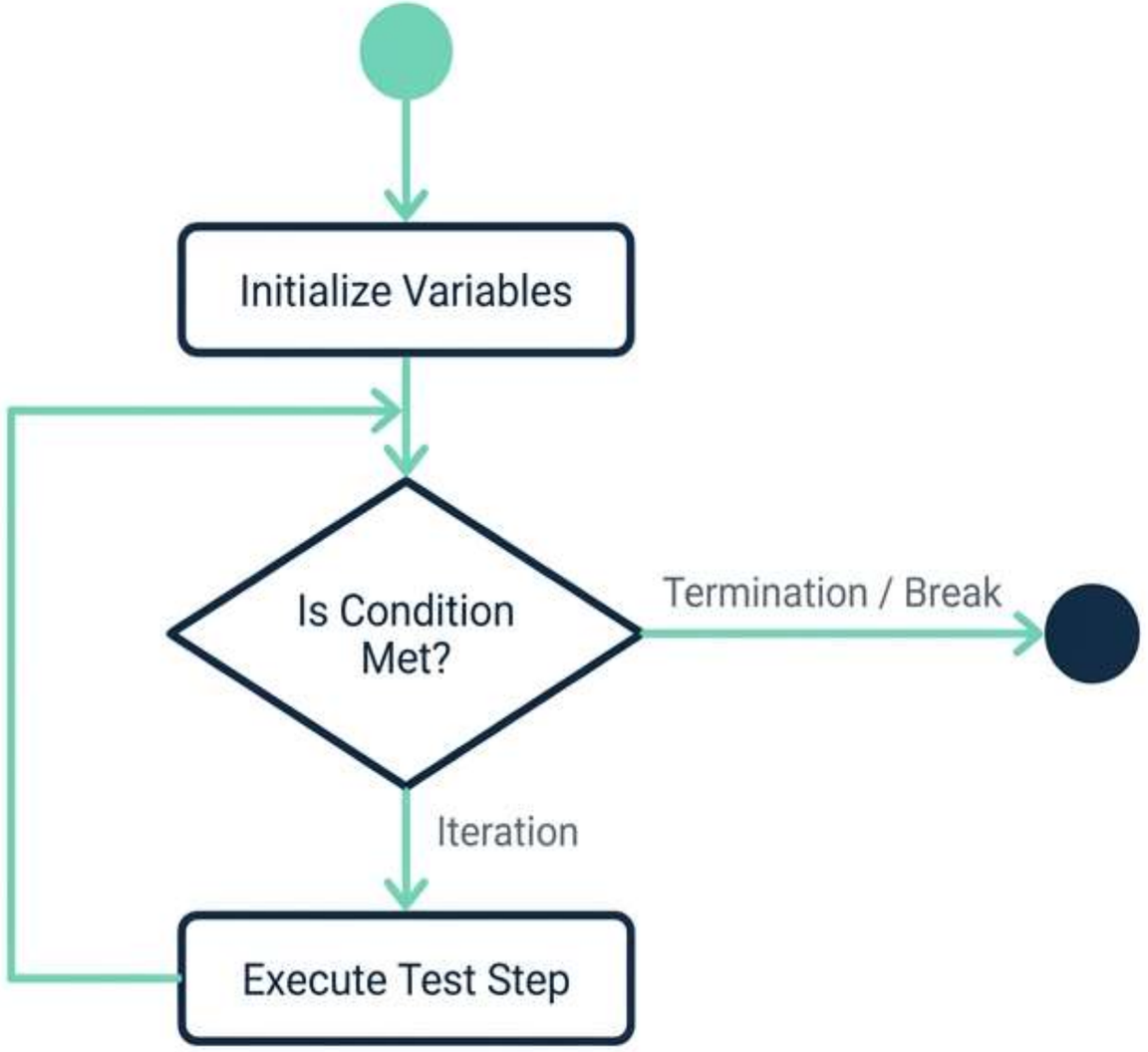
Specialized for traversing Arrays or Collections.

Execution Logic Architecture

Sequential Flow: Default top-to-bottom.

Conditional Branching: Flow splits on data validation.

Looping: Flow cycles to process collections.



Testleaf
Always Ahead

Applying Logic to Selenium Automation

Multi-Browser Testing (Switch Case)

Input: 'Chrome', 'Edge', or 'Firefox'.

```
switch(browser) {
  case 'Chrome':
   driver = new ChromeDriver();
}
```

Test Reporting (If/Else)

Validating test outcomes.

```
if(actual.equals(expected))
  log('PASS');
else
  log('FAIL');
```

Data Tables (For Loop)

Parsing search results.

```
for(WebElement row : rows) {
  text = row.getText();
}
```

Coding Standards for Control Statements

Testleaf
Always Ahead

✓ Best Practices	✗ Avoid
Use Switch for fixed values (Enums)	Chaining 10+ “else-if” statements
Minimize Nesting (fail fast/early return)	Deeply nested “Spaghetti Code”
Always include a “default” case	Leaving unexpected inputs unhandled
Ensure loop variables update	Infinite loops that hang execution

Ace the Technical Interview

Testleaf
Always Ahead

Q: When should you use a `while` loop vs. a `do-while` loop?

A: Use `do-while` when the code block **must** execute at least once before checking the condition.

Q: Explain the difference between `break` and `continue`.

A: `break` exits the loop entirely; `continue` skips the current iteration and jumps to the next.

Q: Can you use a switch statement with Strings?

A: Yes, starting from Java 7, String objects are supported in switch cases.

Session Recap

Testleaf
Always Ahead

1

Selection

Use `if/else` for boolean logic and `switch` for fixed values to improve readability.

2

Iteration

Use `for` loops when the count is fixed and `while` loops for dynamic wait conditions.

3

Jump

Use `break` to exit loops early and `continue` to skip irrelevant data.

4

Stability

Clean logic leads to stable, maintainable, and debuggable test suites.