Lesson 03 Summary: Control Flow

Data Science with Python – Key Concepts

Data Science Program

Control Flow Overview

Control Flow Summary

Common Patterns:

Accumulator: `total = 0; for x: total += x`

Filter: `[x for x in items if cond]`

Search: `for x: if match: break`

Control flow directs program execution based on conditions

If-Elif-Else – Key Concepts

Conditional execution based on boolean expressions

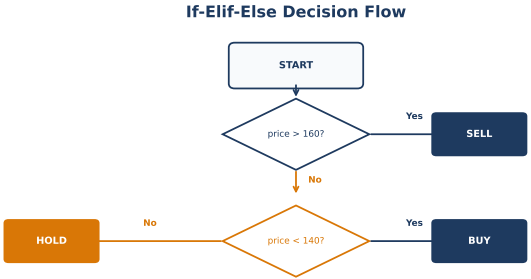
- **if:** First condition to check
- **elif:** Additional conditions (optional, multiple allowed)
- **else:** Fallback when all conditions False (optional)

Trading Example:

```
if price > 160:  action = "SELL"
elif price < 140:  action = "BUY"
else:  action = "HOLD"
```

Conditions evaluated top-to-bottom; first True wins

Decision Flow Diagram



Diamond = condition — Rectangle = action — Arrows = flow

For Loops – Key Concepts

Iterate over sequences (lists, ranges, strings)

- **Syntax:** `for item in sequence:`
- **range():** `range(5)` → 0,1,2,3,4
- **enumerate():** Get index and value together

Examples:

```
for price in prices:  total += price
for i, p in enumerate(prices):  print(f"Day {i}:  {p}")
```

For loops: known iterations, automatic advancement

While Loops – Key Concepts

Repeat until condition becomes False

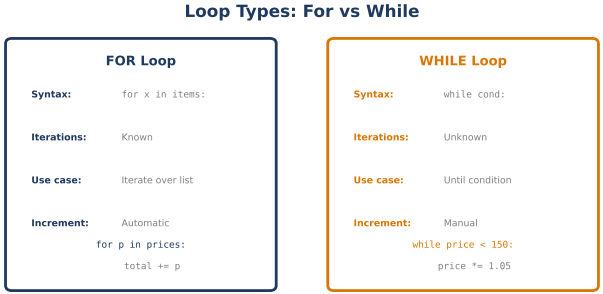
- **Syntax:** `while condition:`
- **Unknown iterations:** Runs until condition fails
- **Manual increment:** You control loop variable

Finance Example:

```
while balance > 5000:
    balance *= 0.95 # 5% loss each iteration
    years += 1
```

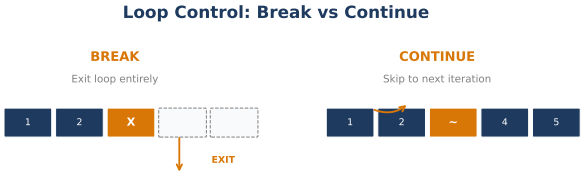
Warning: ensure condition eventually becomes False!

Loop Types Comparison



FOR when you know iterations, WHILE when waiting for condition

Break and Continue



break = exit loop — continue = skip to next iteration

Common Patterns

Essential loop patterns for data analysis:

1. Accumulator: Sum, count, average

```
total = 0
for price in prices:  total += price
```

2. Filter: Select matching items

```
high = [p for p in prices if p > 150]
```

3. Search: Find first match

```
for p in prices:
    if p > 200:  break
```

Combine patterns for complex trading logic

Quick Reference

Essential Syntax:

Structure	Syntax
Conditional	`if cond: ... elif cond: ... else: ...`
For loop	`for x in items: ...`
While loop	`while cond: ...`
Range	`range(start, stop, step)`
Enumerate	`for i, x in enumerate(items):`
Break	`break` (exit loop)
Continue	`continue` (skip iteration)

Remember: Colon after condition/loop — Indentation matters!

Master control flow for trading system logic