

Common Control Flow Patterns

Pattern 1: Guard Clause

```
def buy_stock(price, balance):  
    # Guard: Check preconditions  
    if price <= 0:  
        return "Invalid price"  
    if balance < price:  
        return "Insufficient funds"  
  
    # Main logic  
    execute_buy(price)
```

Pattern 2: Accumulator

```
prices = [150, 165, 148, 172]  
total = 0 # Accumulator  
  
for price in prices:  
    total += price  
  
average = total / len(prices)  
print(f"Avg: ${average:.2f}")
```

Pattern 3: Search & Break

```
prices = [150, 165, 148, 172]  
found = False  
  
for price in prices:  
    if price < 150:  
        print(f"Found: ${price}")  
        found = True  
        break # Stop searching
```

Pattern 4: Filter Pattern

```
all_prices = [150, 165, 148, 172]  
high_prices = [] # Filtered list  
  
for price in all_prices:  
    if price > 160:  
        high_prices.append(price)  
  
# Result: [165, 172]
```

Pattern 5: Counter

```
prices = [150, 165, 148, 172, 145]  
count_low = 0 # Counter  
  
for price in prices:  
    if price < 150:  
        count_low += 1  
  
print(f"{count_low} low prices")
```

Pattern 6: Find Min/Max

```
prices = [150, 165, 148, 172]  
max_price = prices[0] # Initialize  
  
for price in prices:  
    if price > max_price:  
        max_price = price  
  
print(f"Max: ${max_price}")
```