# 📚 Module 18: Absolute Values - Complete Notes

## 🎯 What You'll Learn

In this module, you'll master **absolute values** — distance from zero, error margins, and tolerance checking.

---

## 📖 Concept Explained (Like a YouTube Video)

### The Basics

**Absolute value** is the distance from zero — always positive.

```
|5| = 5    (5 units from zero)
|-5| = 5   (also 5 units from zero)
|0| = 0    (zero units from zero)

Key: |x| removes the sign, keeps magnitude
```

### Visual Understanding

```
Number line:
       -5        0        +5
    ───┼─────────┼────────┼───
       |<─ 5 ─>|<─ 5 ─>|

Both -5 and +5 are distance 5 from zero
```

### Distance Between Values

```
Distance from A to B = |A - B|

Example: Distance between 3 and 7
|3 - 7| = |-4| = 4
|7 - 3| = |4| = 4  (same!)

Order doesn't matter with absolute value!
```

---

## 💻 Programming Connection

### Code Examples

```python
# Example 1: Basic Absolute Value

# Built-in abs() function
print(abs(5))     # 5
print(abs(-5))    # 5
print(abs(0))     # 0
print(abs(-3.7))  # 3.7
```

```python
# Manual implementation
def absolute(x):
    return x if x >= 0 else -x
```

```python
# Example 2: Distance Between Values

def distance(a, b):
    """Distance between two values (always positive)"""
    return abs(a - b)

print(distance(10, 3))   # 7
print(distance(3, 10))   # 7 (order doesn't matter)
print(distance(-5, 5))   # 10
```

```python
# Example 3: Tolerance Checking (THE BIG ONE)

def within_tolerance(actual, expected, tolerance):
    """Check if actual is within tolerance of expected"""
    return abs(actual - expected) <= tolerance

# Is 98 within ±5 of 100?
print(within_tolerance(98, 100, 5))    # True (diff = 2)
print(within_tolerance(94, 100, 5))    # False (diff = 6)

# Floating point comparison
def almost_equal(a, b, epsilon=1e-9):
    return abs(a - b) < epsilon

print(almost_equal(0.1 + 0.2, 0.3))  # True!
```

```python
# Example 4: Error Margin Calculation

def calculate_error(measured, actual):
    """Calculate absolute and percentage error"""
    abs_error = abs(measured - actual)
    pct_error = (abs_error / abs(actual)) * 100 if actual != 0 else 0

    return {
        "measured": measured,
        "actual": actual,
        "absolute_error": abs_error,
        "percentage_error": round(pct_error, 2)
    }

result = calculate_error(98, 100)
print(result)  # {'absolute_error': 2, 'percentage_error': 2.0}
```

```
# Example 5: Find Closest Value

def find_closest(target, values):
    """Find value closest to target"""
    return min(values, key=lambda v: abs(v - target))

values = [10, 25, 47, 82, 95]
print(find_closest(50, values))  # 47 (closest to 50)
print(find_closest(80, values))  # 82 (closest to 80)
```

---

## 🧪 SDET/Testing Application

```
# SDET Scenario: Validate With Tolerance

def validate_value(actual, expected, tolerance_pct):
    """Validate value within percentage tolerance"""
    tolerance = expected * tolerance_pct / 100
    diff = abs(actual - expected)

    return {
        "actual": actual,
        "expected": expected,
        "difference": diff,
        "tolerance": tolerance,
        "passed": diff <= tolerance,
        "status": "✅ PASS" if diff <= tolerance else "❌ FAIL"
    }

# Response time within 10% of baseline
result = validate_value(108, 100, 10)
print(f"Status: {result['status']}")  # ✅ PASS (diff=8, tolerance=10)
```

```
# SDET Scenario: Performance Deviation Check

def check_deviation(baseline_times, current_time, max_deviation_pct):
    """Check if current time deviates too much from baseline avg"""
    baseline_avg = sum(baseline_times) / len(baseline_times)
    deviation = abs(current_time - baseline_avg)
    max_allowed = baseline_avg * max_deviation_pct / 100

    return {
        "baseline_avg": round(baseline_avg, 2),
        "current": current_time,
        "deviation": round(deviation, 2),
        "max_allowed": round(max_allowed, 2),
        "within_bounds": deviation <= max_allowed
    }
```

```
baseline = [100, 105, 98, 102, 95]
result = check_deviation(baseline, 115, 10)
print(result)
```

---

## 🎓 Practice Problems

### Problem 1: Easy 🟢

**Challenge**: What is |–7 – 3|?

### Problem 2: Medium 🟡

**Scenario**: Expected response time is 200ms. Acceptable tolerance is ±15%.
**Challenge**: What's the valid range? Is 225ms acceptable?

### Problem 3: Application 🔴

**Scenario**: You have sensor readings: [98.5, 101.2, 99.8, 100.5]. Target is 100.
**Challenge**: Find which reading is closest to target and which is furthest.

---

## ⚠️ Common Mistakes

❌ **Mistake 1**: Thinking |a - b| ≠ |b - a|
✅ **Fix**: They're always equal! Distance is symmetric.

❌ **Mistake 2**: Using signed difference for error
✅ **Fix**: Always use abs() for magnitude of error

---

## 🔑 Key Takeaways

✅ **Absolute = Distance from zero** — Always non-negative
✅ **|a - b| = Distance between a and b**
✅ **Tolerance check**: `abs(actual - expected) <= tolerance`
✅ **Order doesn't matter**: `abs(a - b) == abs(b - a)`

---

💾 *Save as:* *`Module_18_Absolute_Values.md`*

**Phase 3 Complete!** 🚀