📚 Module 21: Basic Geometry - Complete Notes

🎯 What You'll Learn

Master **area and perimeter** calculations — bounding boxes, overlap detection, and UI layout testing.

📖 Concept Explained

Key Formulas

```
Rectangle:
    Area = length × width
    Perimeter = 2 × (length + width)

Square:
    Area = side²
    Perimeter = 4 × side
```

💻 Programming Connection

Code Examples

```python
# Example 1: Basic Calculations

def rectangle_area(length, width):
    return length * width

def rectangle_perimeter(length, width):
    return 2 * (length + width)

# 10x5 rectangle
print(f"Area: {rectangle_area(10, 5)}")       # 50
print(f"Perimeter: {rectangle_perimeter(10, 5)}")  # 30
```

```python
# Example 2: Bounding Box from Points

def bounding_box(points):
    """Find smallest rectangle containing all points"""
    min_x = min(p[0] for p in points)
    max_x = max(p[0] for p in points)
    min_y = min(p[1] for p in points)
    max_y = max(p[1] for p in points)

    return {
        "min": (min_x, min_y),
        "max": (max_x, max_y),
        "width": max_x - min_x,
        "height": max_y - min_y,
        "area": (max_x - min_x) * (max_y - min_y)
    }

points = [(1, 2), (4, 5), (3, 1)]
print(bounding_box(points))
```

```
# Example 3: Overlap Detection

def rectangles_overlap(r1, r2):
    """Check if two rectangles overlap
    Format: (x1, y1, x2, y2) - corners"""
    # No overlap if separated horizontally or vertically
    if r1[2] <= r2[0] or r2[2] <= r1[0]:
        return False  # Horizontal separation
    if r1[3] <= r2[1] or r2[3] <= r1[1]:
        return False  # Vertical separation
    return True

rect_a = (0, 0, 5, 5)
rect_b = (3, 3, 8, 8)
rect_c = (6, 0, 10, 4)

print(rectangles_overlap(rect_a, rect_b))  # True
print(rectangles_overlap(rect_a, rect_c))  # False
```

🧪 SDET/Testing Application

```
# SDET Scenario: Element Overlap Testing

def check_element_overlap(elem1, elem2):
    """Check if two UI elements overlap"""
    r1 = (elem1['x'], elem1['y'],
          elem1['x'] + elem1['width'],
          elem1['y'] + elem1['height'])
    r2 = (elem2['x'], elem2['y'],
          elem2['x'] + elem2['width'],
          elem2['y'] + elem2['height'])

    return rectangles_overlap(r1, r2)

button = {'x': 100, 'y': 50, 'width': 80, 'height': 30}
popup = {'x': 150, 'y': 40, 'width': 200, 'height': 150}
print(f"Overlap: {check_element_overlap(button, popup)}")  # True
```

🔑 Key Takeaways

✅ **Area = Width × Height**

✅ **Bounding box** — Smallest container for points

✅ **Overlap** — Check for separation in both axes

💾 *Save as:* `Module_21_Basic_Geometry.md`

Phase 4 Complete! 🚀