📚 Module 16: Substitution - Complete Notes

🎯 What You'll Learn

In this module, you'll master **substitution** — replacing variables with values to evaluate expressions. This is how all templates and parameterization work.

📖 Concept Explained (Like a YouTube Video)

The Basics

Substitution means replacing a variable with its actual value.

```
Expression: 2x + 3y
Substitute: x = 5, y = 2

Result: 2(5) + 3(2) = 10 + 6 = 16
```

💻 Programming Connection

Code Examples

```python
# Example 1: Function Parameters ARE Substitution

def calculate_area(length, width):
    """Substitute length and width into formula"""
    return length * width

# Substituting: length=5, width=3
print(calculate_area(5, 3))  # 15
```

```python
# Example 2: String Template Substitution

def build_message(name, action, item):
    """Substitute values into template"""
    template = "User {name} {action} {item}"
    return template.format(name=name, action=action, item=item)

print(build_message("John", "purchased", "Widget"))
# "User John purchased Widget"

# F-string substitution
name = "John"
action = "purchased"
print(f"User {name} {action} an item")
```

```python
# Example 3: Configuration Substitution

def build_url(base, version, endpoint, params=None):
    """Build URL by substituting components"""
    url = f"{base}/{version}/{endpoint}"
    if params:
        query = "&".join(f"{k}={v}" for k, v in params.items())
        url += f"?{query}"
    return url

result = build_url(
    base="https://api.example.com",
    version="v1",
    endpoint="users",
    params={"page": 1, "limit": 10}
)
print(result)
# https://api.example.com/v1/users?page=1&limit=10
```

```python
# Example 4: Formula with Multiple Substitutions

def evaluate_formula(**kwargs):
    """Evaluate quadratic: ax² + bx + c for given x"""
    a = kwargs.get('a', 0)
    b = kwargs.get('b', 0)
    c = kwargs.get('c', 0)
    x = kwargs.get('x', 0)

    return a * (x ** 2) + b * x + c

# Substitute: a=1, b=-5, c=6, x=2
result = evaluate_formula(a=1, b=-5, c=6, x=2)
print(result)  # 1(4) + -5(2) + 6 = 0
```

🧪 SDET/Testing Application

```python
# SDET Scenario: Test Data Templates

def create_test_data(template, **substitutions):
    """Generate test data by substituting values into template"""
    result = {}
    for key, value in template.items():
        if isinstance(value, str) and '{' in value:
            result[key] = value.format(**substitutions)
        else:
            result[key] = value
    return result
```

```
template = {
    "email": "{name}@{domain}",
    "username": "{name}_{id}",
    "password": "Test123!"
}

test_data = create_test_data(template, name="john", domain="test.com", id="001")
print(test_data)
# {'email': 'john@test.com', 'username': 'john_001', 'password': 'Test123!'}
```

```
# SDET Scenario: API Request Template

def make_request(template, **params):
    """Build API request from template"""
    return {
        "url": template["url"].format(**params),
        "method": template["method"],
        "headers": template.get("headers", {}),
        "body": {k: v.format(**params) if isinstance(v, str) else v
                 for k, v in template.get("body", {}).items()}
    }

template = {
    "url": "{base_url}/users/{user_id}",
    "method": "PUT",
    "body": {"name": "{name}", "email": "{email}"}
}

request = make_request(
    template,
    base_url="https://api.test.com",
    user_id="123",
    name="John Doe",
    email="john@test.com"
)
print(request)
```

🎓 Practice Problems

Problem 1: Easy 🟢

Challenge: Given `f(x) = x² + 2x + 1`, find f(3).

Problem 2: Medium 🟡

Challenge: Create a function that substitutes values into `"Hello, {name}! You have {count} messages."`

Problem 3: Application 🔴

Scenario: API endpoint template: `"/api/{version}/users/{user_id}/orders/{order_id}"`
Challenge: Create a function to build the full URL with substitutions.

🔑 Key Takeaways

✅ **Substitution = Replace with value**
✅ **Functions are substitution machines**
✅ **Templates use substitution for flexibility**
✅ **Evaluate step by step** — Substitute, then calculate

💾 *Save as:* `Module_16_Substitution.md`