

แผนกิจกรรม: ทำความเข้าใจและประยุกต์ใช้โครงสร้าง **LinkedTree**

กิจกรรมที่ 1: แนวคิดของ **Linked Tree**

วัตถุประสงค์

เมื่อเสร็จสิ้นกิจกรรมนี้ นักศึกษาจะสามารถ:

- เข้าใจว่า **Linked Tree** จัดเก็บข้อมูลแบบลำดับชั้นได้อย่างไร
- ระบุองค์ประกอบสำคัญของ โครงสร้างข้อมูลต้นไม้
- วิเคราะห์ว่า ต้นไม้แตกต่างจากโครงสร้างข้อมูลอื่น ๆ อย่างไร

*ให้ศึกษาทบทวนและลงมือเขียนโค้ด **Python** จากการบรรยายในชั้นเรียนด้วยตนเอง

กิจกรรมที่ 2: การจัดโครงสร้างองค์กรบริษัท

วัตถุประสงค์

เมื่อเสร็จสิ้นกิจกรรมนี้ นักศึกษาจะสามารถ:

- สร้าง โครงสร้างองค์กรบริษัท โดยใช้ **LinkedTree**
- เพิ่มและดึงข้อมูลพนักงานได้แบบไดนามิก
- แสดงโครงสร้างลำดับชั้นโดยใช้ **Recursive Traversal**

คำแนะนำ

1. อ่านโจทย์ปัญหา:

- บริษัทมี **CEO** ที่อยู่บนสุดของลำดับชั้น
- CEO บริหาร **Manager** หลายคน
- แต่ละ **Manager** มีพนักงานอยู่ภายใต้การดูแล
- ต้องสร้างต้นไม้เพื่อแสดง โครงสร้างองค์กรนี้

2. เขียนโค้ด **Python** ตามตัวอย่างต่อไปนี้:

```
class CompanyHierarchy(LinkedTree):  
    def display_hierarchy(self, p, level=0):  
        print(" " * level + p.element())  
        for child in self.children(p):  
            self.display_hierarchy(child, level + 1)
```

ตัวอย่างการใช้งาน:

```
company = CompanyHierarchy()  
ceo = company.add_child(None, "CEO")  
manager1 = company.add_child(ceo, "Manager 1")
```

```
manager2 = company.add_child(ceo, "Manager 2")
employee1 = company.add_child(manager1, "Employee 1")
employee2 = company.add_child(manager1, "Employee 2")
company.display_hierarchy(company.root())
```

3. รันโค้ดและสังเกตผลลัพธ์ ควรแสดงผลดังนี้:

CEO

Manager 1

Employee 1

Employee 2

Manager 2

4. แก้ไขโค้ดเพื่อ:

- เพิ่มระดับ **Supervisor** เข้าไปในโครงสร้างองค์กร
- สร้างเมธอดสำหรับ ค้นหาพนักงานโดยใช้ชื่อ