บทที่ 9
การโปรแกรมภาษาไพทอนเบื้องต้น
Introduction to Python Programming

บทนำ

ภาษาไพทอน (Python) เป็นหนึ่งในภาษาโปรแกรมที่ได้รับความนิยมอย่างแพร่หลายในระดับโลก ทั้งในด้านการศึกษา การพัฒนาโปรแกรม การวิเคราะห์ข้อมูล ปัญญาประดิษฐ์ ระบบอัตโนมัติ และงานประยุกต์อื่น ๆ อีกมากมาย สาเหตุสำคัญที่ทำให้ภาษาไพทอนได้รับความนิยม คือ การมีรูปแบบคำสั่งที่อ่านง่าย เข้าใจง่าย และมีโครงสร้างที่เอื้อต่อการเรียนรู้สำหรับผู้เริ่มต้น ขณะเดียวกันก็มีศักยภาพเพียงพอสำหรับการประยุกต์ใช้ในงานที่มีความซับซ้อนสูง ภาษาไพทอนจึงเหมาะอย่างยิ่งสำหรับใช้เป็นเครื่องมือในการเรียนรู้พื้นฐานการเขียนโปรแกรมในรายวิชาวิทยาการคำนวณ

สำหรับผู้เรียนที่เริ่มต้นศึกษาการเขียนโปรแกรม ภาษาไพทอนช่วยให้สามารถทำความเข้าใจแนวคิดสำคัญต่าง ๆ ได้อย่างเป็นระบบ เช่น ตัวแปร ชนิดข้อมูล การรับและแสดงผลข้อมูล การใช้คำสั่งควบคุมแบบเรียงลำดับ ทางเลือก และทำซ้ำ ตลอดจนการประยุกต์ใช้โปรแกรมในการแก้ปัญหาอย่างง่าย การเรียนภาษาไพทอนจึงมิได้มุ่งเน้นเพียงการจำคำสั่งหรือไวยากรณ์ของภาษา แต่เป็นการพัฒนาความสามารถในการคิดเชิงตรรกะ การออกแบบลำดับขั้นตอนการทำงาน และการแปลงแนวคิดไปสู่คำสั่งที่คอมพิวเตอร์สามารถทำงานได้

ในบริบทของรายวิชาวิทยาการคำนวณสำหรับนักศึกษาครู การเรียนรู้ภาษาไพทอนมีความสำคัญในสองมิติ มิติแรก คือ การพัฒนาความเข้าใจพื้นฐานด้านการเขียนโปรแกรมในรูปแบบข้อความ ซึ่งเป็นขั้นต่อยอดจากการเรียนรู้ผังงาน การโปรแกรมแบบบล็อกคำสั่ง และการใช้เครื่องมืออย่าง Scratch หรือ Flowgorithm มิติที่สอง คือ การเตรียมความพร้อมให้นักศึกษาครูสามารถนำภาษาไพทอนไปประยุกต์ใช้ในการจัดการเรียนรู้สำหรับนักเรียนในอนาคต โดยเฉพาะการเลือกใช้เครื่องมือที่เหมาะสมและการออกแบบกิจกรรมที่ช่วยลดความซับซ้อนในการเรียนรู้

ในเอกสารประกอบการสอนนี้ ใช้ **Google Colab** เป็นเครื่องมือหลักในการรันโปรแกรมภาษาไพทอน เนื่องจากใช้งานสะดวก ไม่ต้องติดตั้งโปรแกรมเพิ่มเติม รองรับการทำงานผ่านเว็บเบราว์เซอร์ และเอื้อต่อการเรียนรู้ทั้งในห้องเรียนและนอกห้องเรียน นอกจากนี้ ยังใช้ **Python Tutor** เป็นเครื่องมือช่วยอธิบายลำดับขั้นตอนการทำงานของโปรแกรมภาษาไพทอน โดยเฉพาะการติดตามค่าของตัวแปร

และการทำงานของคำสั่งทีละขั้นตอน ซึ่งช่วยให้ผู้เรียนเข้าใจการทำงานภายในของโปรแกรมได้ง่ายขึ้น และลดความสับสนที่มักเกิดขึ้นในผู้เริ่มต้นเรียนเขียนโปรแกรม

บทนี้มุ่งอธิบายความรู้เบื้องต้นเกี่ยวกับภาษาไพทอน ตัวแปร ชนิดข้อมูล และการรับ–แสดงผลข้อมูล การใช้คำสั่งควบคุมแบบเรียงลำดับ ทางเลือก และทำซ้ำในภาษาไพทอน การเขียนโปรแกรมแก้ปัญหาอย่างง่าย ตลอดจนการออกแบบกิจกรรมการเรียนรู้เบื้องต้นสำหรับการสอนภาษาไพทอน เพื่อให้นักศึกษาครูสามารถใช้ภาษาไพทอนได้อย่างมีพื้นฐานที่มั่นคง และสามารถนำไปประยุกต์ใช้ในการจัดการเรียนรู้ได้อย่างเหมาะสม

9.1 ความรู้เบื้องต้นเกี่ยวกับภาษาไพทอน

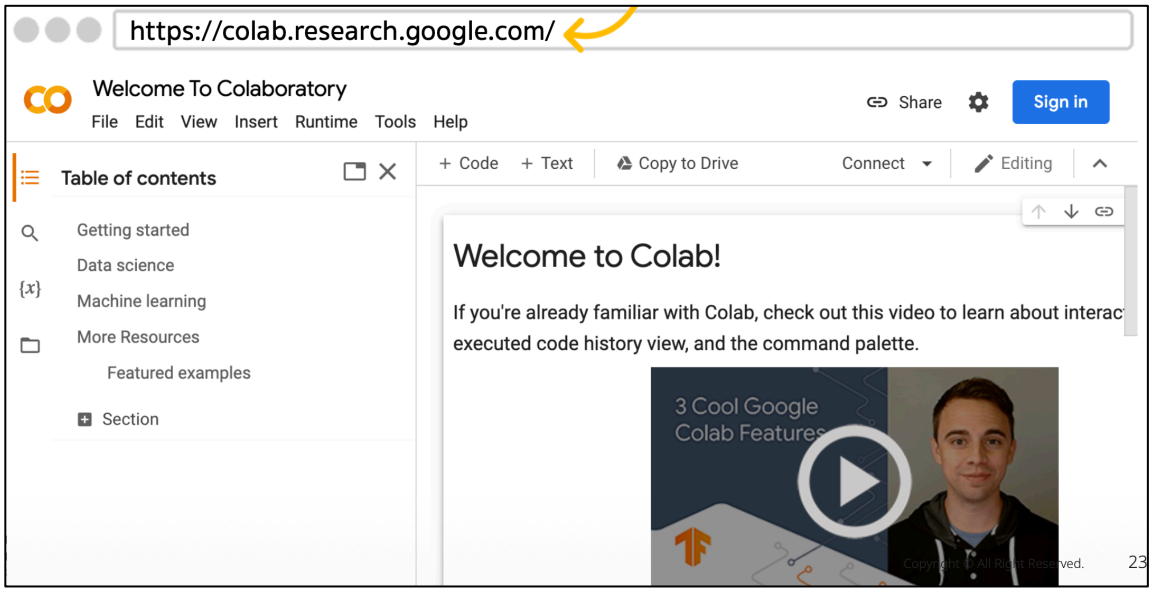
ภาษาไพทอนเป็นภาษาโปรแกรมระดับสูง (High-level Programming Language) ที่ถูกออกแบบมาให้มีรูปแบบคำสั่งอ่านง่าย คล้ายภาษามนุษย์ และเน้นความชัดเจนของโครงสร้างโปรแกรม ไพทอนได้รับการพัฒนาโดย Guido van Rossum และถูกนำมาใช้อย่างแพร่หลายในหลากหลายสาขา เนื่องจากเป็นภาษาที่มีความยืดหยุ่น รองรับการทำงานหลายรูปแบบ และมีชุมชนผู้ใช้งานขนาดใหญ่ที่ช่วยพัฒนาเครื่องมือและแหล่งเรียนรู้จำนวนมาก

จุดเด่นสำคัญของภาษาไพทอน คือ ความเรียบง่ายของไวยากรณ์ เมื่อเปรียบเทียบกับภาษาโปรแกรมอื่น ผู้เรียนสามารถอ่านและทำความเข้าใจโค้ดได้ง่ายกว่า เพราะไพทอนไม่ใช้สัญลักษณ์ซับซ้อนมากเกินจำเป็น เช่น เครื่องหมายปีกกาในการกำหนดขอบเขตของคำสั่ง แต่ใช้การเยื้องบรรทัด (Indentation) เป็นส่วนสำคัญของโครงสร้างภาษา ลักษณะนี้ช่วยส่งเสริมให้ผู้เรียนเขียนโปรแกรมอย่างเป็นระเบียบและมองเห็นลำดับการทำงานได้ชัดเจน

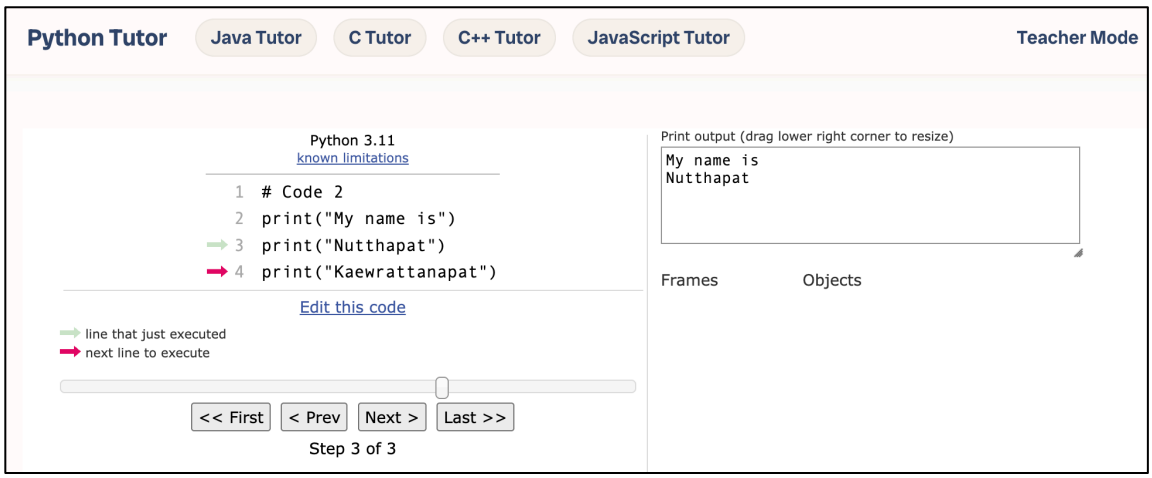
ภาษาไพทอนมีลักษณะเป็นภาษาตีความ (Interpreted Language) หมายถึง โปรแกรมจะถูกแปลและทำงานทีละบรรทัดหรือทีละส่วนในขณะรัน ทำให้ผู้เรียนสามารถทดลองคำสั่ง ตรวจสอบผลลัพธ์ และแก้ไขข้อผิดพลาดได้อย่างรวดเร็ว คุณลักษณะนี้เหมาะอย่างยิ่งต่อการเรียนการสอนในระดับเริ่มต้น เพราะช่วยให้ผู้เรียนเห็นผลของคำสั่งที่ตนเขียนทันที

ในบริบทการศึกษา ภาษาไพทอนได้รับความนิยมอย่างมากเพราะเหมาะสำหรับการเรียนรู้แนวคิดพื้นฐานทางการเขียนโปรแกรม เช่น ตัวแปร การคำนวณ การควบคุมลำดับคำสั่ง การตัดสินใจ และการทำซ้ำ อีกทั้งยังสามารถต่อยอดไปสู่การเรียนรู้ขั้นสูง เช่น การวิเคราะห์ข้อมูล ปัญญาประดิษฐ์ หรือการพัฒนาโปรแกรมประยุกต์ได้ในอนาคต

สำหรับเอกสารประกอบการสอนฉบับนี้ ผู้เขียนเลือกใช้ Google Colab เป็นเครื่องมือในการรันโปรแกรมภาษาไพทอน เนื่องจากผู้เรียนสามารถใช้งานได้สะดวกผ่านเว็บเบราว์เซอร์ โดยไม่จำเป็นต้องติดตั้งโปรแกรมลงในเครื่องคอมพิวเตอร์ ช่วยลดข้อจำกัดด้านอุปกรณ์และเวลาในการเตรียมความพร้อม อีกทั้งยังเหมาะกับการจัดการเรียนรู้ที่เน้นการทดลองและการปรับแก้โค้ดอย่างต่อเนื่อง



นอกจากนี้ การใช้ Python Tutor ยังมีบทบาทสำคัญในการช่วยให้ผู้เรียนเข้าใจการทำงานของภาษาไพทอนในระดับลึกขึ้น โดยเฉพาะในประเด็นที่ผู้เรียนมักสับสน เช่น การเปลี่ยนแปลงค่าของตัวแปร การทำงานของลูป หรือการตัดสินใจตามเงื่อนไข Python Tutor แสดงลำดับการทำงานของโปรแกรมทีละขั้นตอน ทำให้ผู้เรียนสามารถเห็นภาพการทำงานภายในได้อย่างเป็นระบบ



สำหรับนักศึกษาครู ความรู้เบื้องต้นเกี่ยวกับภาษาไพทอนไม่ควรจำกัดอยู่เพียงการรู้จักคำสั่ง แต่ควรครอบคลุมถึงความเข้าใจว่าทำไมไพทอนจึงเป็นภาษาที่เหมาะต่อการเรียนรู้เบื้องต้น และเครื่องมือใดบ้างที่ช่วยให้ผู้เรียนเข้าถึงการเขียนโปรแกรมได้ง่ายขึ้น ความเข้าใจเช่นนี้จะช่วยให้สามารถออกแบบการเรียนรู้ที่ลดความซับซ้อนและส่งเสริมความเข้าใจอย่างแท้จริง

กล่าวโดยสรุป ภาษาไพทอนเป็นภาษาโปรแกรมที่เหมาะสำหรับการเรียนรู้พื้นฐานการเขียนโปรแกรม เนื่องจากมีโครงสร้างที่เรียบง่าย อ่านง่าย และเอื้อต่อการทดลองปฏิบัติ เมื่อใช้ร่วมกับ Google Colab และ Python Tutor จะช่วยให้ผู้เรียนสามารถเรียนรู้ได้อย่างสะดวก เข้าใจง่าย และเห็นกระบวนการทำงานของโปรแกรมได้อย่างชัดเจน

9.2 ตัวแปร ชนิดข้อมูล และการรับ–แสดงผลข้อมูล

ตัวแปร (Variable) เป็นแนวคิดพื้นฐานสำคัญของการเขียนโปรแกรม หมายถึง ชื่อที่ใช้แทนพื้นที่จัดเก็บข้อมูลในหน่วยความจำของคอมพิวเตอร์ เพื่อให้โปรแกรมสามารถนำข้อมูลนั้นไปใช้งานหรือปรับเปลี่ยนค่าได้ระหว่างการทำงาน ผู้เรียนสามารถเข้าใจตัวแปรได้ในลักษณะของ “กล่องเก็บข้อมูล” ที่มีชื่อกำกับไว้ เมื่อโปรแกรมต้องการเก็บค่าบางอย่าง เช่น คะแนน ชื่อ อายุ หรือผลลัพธ์จากการคำนวณ ก็จะนำข้อมูลเหล่านั้นไปเก็บไว้ในตัวแปร

การตั้งชื่อตัวแปรในภาษาไพทอนควรสื่อความหมาย ชัดเจน และสอดคล้องกับข้อมูลที่เก็บ เช่น score, name, total, average หรือ age การใช้ชื่อตัวแปรที่เหมาะสมจะช่วยให้โปรแกรมอ่านง่ายและเข้าใจได้สะดวกยิ่งขึ้น โดยเฉพาะในบริบทการเรียนการสอนที่ผู้เรียนยังต้องพัฒนาความเข้าใจเกี่ยวกับโครงสร้างของโปรแกรมอย่างเป็นระบบ

ชนิดข้อมูล (Data Type) หมายถึง ประเภทของข้อมูลที่เก็บอยู่ในตัวแปร ซึ่งภาษาไพทอนรองรับข้อมูลหลายชนิด แต่ในการเรียนรู้เบื้องต้นมักเน้นชนิดข้อมูลสำคัญ เช่น จำนวนเต็ม (Integer) จำนวนทศนิยม (Float) ข้อความ (String) และค่าความจริง (Boolean) การเข้าใจชนิดข้อมูลมีความสำคัญเพราะส่งผลต่อวิธีการประมวลผลและผลลัพธ์ของโปรแกรม เช่น การบวกตัวเลขย่อมต่างจากการเชื่อมข้อความ หรือการตรวจสอบเงื่อนไขย่อมเกี่ยวข้องกับค่าจริงและเท็จ

การรับข้อมูล (Input) เป็นกระบวนการที่โปรแกรมรับค่าจากผู้ใช้เพื่อนำไปประมวลผลต่อ ในภาษาไพทอนคำสั่งพื้นฐานที่ใช้รับข้อมูล คือ input() ตัวอย่างเช่น การรับชื่อหรือคะแนนจากผู้ใช้ อย่างไรก็ตาม ผู้เรียนควรเข้าใจด้วยว่า ข้อมูลที่รับผ่าน input() มักอยู่ในรูปข้อความ (String) ก่อน หากต้องการ

นำไปคำนวณเป็นตัวเลข จำเป็นต้องแปลงชนิดข้อมูลก่อน เช่น ใช้ int() สำหรับจำนวนเต็ม หรือ float() สำหรับจำนวนทศนิยม

การแสดงผลข้อมูล (Output) คือ กระบวนการนำข้อมูลหรือผลลัพธ์ออกมาแสดงให้ผู้ใช้เห็น ในภาษาไพทอนมักใช้คำสั่ง print() ในการแสดงข้อความ ตัวเลข หรือค่าของตัวแปร การแสดงผลอย่างเหมาะสมช่วยให้ผู้ใช้เข้าใจสิ่งที่โปรแกรมทำได้ชัดเจน เช่น การแสดงผลลัพธ์พร้อมข้อความอธิบาย หรือการแสดงค่าที่คำนวณได้ในรูปแบบที่อ่านง่าย

ตัวอย่างโปรแกรมเบื้องต้นเกี่ยวกับการรับและแสดงผลข้อมูล เช่น โปรแกรมรับชื่อนักเรียนแล้วแสดงข้อความต้อนรับ หรือโปรแกรมรับคะแนนสองวิชาแล้วคำนวณผลรวมและค่าเฉลี่ย กิจกรรมลักษณะนี้ช่วยให้ผู้เรียนเข้าใจบทบาทของตัวแปร การจัดเก็บข้อมูล การรับข้อมูลจากผู้ใช้ และการแสดงผลลัพธ์อย่างเป็นระบบ

สำหรับนักศึกษาครู การสอนเรื่องตัวแปรและชนิดข้อมูลควรใช้การอธิบายที่เชื่อมโยงกับสิ่งที่ผู้เรียนเข้าใจง่าย เช่น เปรียบตัวแปรเป็นกล่องหรือป้ายชื่อของข้อมูล และควรใช้ Python Tutor ช่วยแสดงให้เห็นว่าเมื่อโปรแกรมทำงาน ค่าของตัวแปรถูกสร้าง เปลี่ยนแปลง หรือถูกนำไปใช้อย่างไร เครื่องมือเช่นนี้ช่วยลดความเป็นนามธรรมของแนวคิดและทำให้ผู้เรียนเข้าใจการทำงานภายในของโปรแกรมได้ชัดเจนขึ้น

กล่าวโดยสรุป ตัวแปร ชนิดข้อมูล และการรับ–แสดงผลข้อมูล เป็นพื้นฐานสำคัญของการเขียนโปรแกรมภาษาไพทอน เพราะเป็นกลไกหลักในการจัดการข้อมูลที่โปรแกรมต้องใช้ การเข้าใจประเด็นเหล่านี้อย่างถูกต้องจะช่วยให้ผู้เรียนสามารถเขียนโปรแกรมแก้ปัญหาอย่างง่ายได้อย่างมีประสิทธิภาพ

9.3 การใช้คำสั่งควบคุมแบบเรียงลำดับ ทางเลือก และทำซ้ำในภาษาไพทอน

การเขียนโปรแกรมภาษาไพทอนมิได้อาศัยเพียงการกำหนดตัวแปรหรือการรับ–แสดงผลข้อมูลเท่านั้น แต่ยังต้องอาศัยคำสั่งควบคุม (Control Structures) เพื่อกำหนดลำดับและรูปแบบการทำงานของโปรแกรม คำสั่งควบคุมพื้นฐานที่สำคัญประกอบด้วย การทำงานแบบเรียงลำดับ การทำงานแบบทางเลือก และการทำงานแบบทำซ้ำ ซึ่งเป็นแนวคิดเดียวกับที่ผู้เรียนเคยศึกษาในรูปของผังงานและ Flowgorithm มาก่อน

9.3.1 การทำงานแบบเรียงลำดับ

การทำงานแบบเรียงลำดับ คือ การที่โปรแกรมดำเนินคำสั่งทีละคำสั่งจากบนลงล่างตามลำดับที่เขียนไว้ โดยไม่มีการข้ามขั้นตอนหรือการตัดสินใจเพิ่มเติม ตัวอย่างเช่น โปรแกรมรับคะแนนสองจำนวน

คำนวณผลรวม และแสดงผลลัพธ์ โปรแกรมลักษณะนี้ช่วยให้ผู้เรียนเข้าใจว่า ลำดับของคำสั่งมีผลโดยตรงต่อผลลัพธ์ หากสลับขั้นตอนอาจทำให้โปรแกรมทำงานผิดพลาดได้

9.3.2 การทำงานแบบทางเลือก

การทำงานแบบทางเลือกใช้เมื่อโปรแกรมต้องตัดสินใจภายใต้เงื่อนไขบางประการ ในภาษาไพทอนมักใช้คำสั่ง if, if-else หรือ if-elif-else เพื่อกำหนดแนวทางการทำงานที่แตกต่างกันตามผลของเงื่อนไข ตัวอย่างเช่น หากคะแนนมากกว่าหรือเท่ากับ 50 ให้แสดงผลว่า “ผ่าน” แต่หากน้อยกว่า 50 ให้แสดงผลว่า “ไม่ผ่าน” การเรียนรู้เรื่องนี้ช่วยให้ผู้เรียนเข้าใจความสัมพันธ์ระหว่างเงื่อนไขกับผลลัพธ์ของโปรแกรม

9.3.3 การทำงานแบบทำซ้ำ

การทำงานแบบทำซ้ำใช้เมื่อโปรแกรมต้องดำเนินคำสั่งเดิมหลายครั้ง ในภาษาไพทอนสามารถใช้คำสั่ง for และ while เพื่อควบคุมการทำงานซ้ำ เช่น การแสดงตัวเลข 1 ถึง 10 การหาผลรวมของข้อมูลหลายค่า หรือการรับค่าจนกว่าจะถูกต้อง การทำซ้ำช่วยลดความซ้ำซ้อนของโปรแกรมและทำให้การจัดการงานที่เกิดซ้ำเป็นไปอย่างมีประสิทธิภาพ

การสอนคำสั่งควบคุมในภาษาไพทอนควรเชื่อมโยงกับสิ่งที่ผู้เรียนเรียนมาก่อน เช่น ผังงานแบบเรียงลำดับ ทางเลือก และทำซ้ำ เพื่อให้ผู้เรียนเห็นว่า ภาษาไพทอนเป็นเพียงอีกวิธีหนึ่งในการถ่ายทอดแนวคิดเดียวกันจากรูปแบบภาพไปสู่รูปแบบข้อความ การเชื่อมโยงเช่นนี้จะช่วยลดความรู้สึกว่าการเขียนโค้ดเป็นเรื่องใหม่ที่แยกขาดจากความรู้เดิม

ในบริบทของการเรียนรู้ ผู้สอนสามารถใช้ Google Colab เพื่อให้ผู้เรียนทดลองเขียนและรันคำสั่งควบคุมแต่ละประเภทได้ทันที และใช้ Python Tutor เพื่อช่วยอธิบายลำดับการทำงานของโปรแกรม เช่น การตรวจสอบว่าเงื่อนไขเป็นจริงหรือเท็จอย่างไร การวนซ้ำเปลี่ยนค่าตัวแปรอย่างไร หรือเหตุใดโปรแกรมจึงให้ผลลัพธ์เช่นนั้น เครื่องมือทั้งสองนี้ช่วยให้การเรียนรู้คำสั่งควบคุมชัดเจนและตรวจสอบได้มากขึ้น

สำหรับนักศึกษาครู ความเข้าใจเรื่องคำสั่งควบคุมควรครอบคลุมทั้งการเขียนโปรแกรมและการอธิบายแก่ผู้เรียนในระดับการศึกษาขั้นพื้นฐาน ครูควรสามารถเลือกตัวอย่างง่าย ๆ ที่ใกล้ตัว เช่น การตรวจสอบผลสอบ การนับเลข หรือการทำกิจกรรมซ้ำ ๆ ในชีวิตประจำวัน แล้วค่อยเชื่อมไปสู่โค้ดในภาษาไพทอน เพื่อให้ผู้เรียนเห็นว่าคำสั่งควบคุมเป็นส่วนหนึ่งของการคิดอย่างมีระบบ ไม่ใช่เพียงรูปแบบคำสั่งที่ต้องท่องจำ

กล่าวโดยสรุป การใช้คำสั่งควบคุมแบบเรียงลำดับ ทางเลือก และทำซ้ำในภาษาไพทอน เป็นแกนหลักของการพัฒนาโปรแกรมที่มีความหมายและสามารถแก้ปัญหาได้อย่างเป็นระบบ ผู้เรียนที่เข้าใจแนวคิดนี้จะสามารถต่อยอดไปสู่การเขียนโปรแกรมที่ซับซ้อนขึ้นได้อย่างมั่นคง

9.4 การเขียนโปรแกรมแก้ปัญหาอย่างง่ายด้วยภาษาไพทอน

การเขียนโปรแกรมแก้ปัญหาอย่างง่ายด้วยภาษาไพทอนเป็นขั้นตอนที่ช่วยให้ผู้เรียนนำความรู้เรื่องตัวแปร ชนิดข้อมูล การรับ–แสดงผล และคำสั่งควบคุม มาประยุกต์ใช้ร่วมกันในการสร้างโปรแกรมที่ตอบโจทย์ปัญหาจริง การเรียนรู้ในส่วนนี้มีความสำคัญอย่างยิ่ง เพราะเป็นจุดที่ผู้เรียนเปลี่ยนจากการเรียนรู้คำสั่งแยกส่วน ไปสู่การใช้แนวคิดทางวิทยาการคำนวณเพื่อออกแบบวิธีแก้ปัญหาอย่างเป็นระบบ

กระบวนการเขียนโปรแกรมแก้ปัญหาอย่างง่ายควรเริ่มจากการวิเคราะห์โจทย์หรือสถานการณ์ก่อนว่า ต้องการข้อมูลนำเข้าอะไร ต้องประมวลผลอย่างไร และควรแสดงผลลัพธ์ในรูปแบบใด หลักการนี้สอดคล้องกับแนวคิด Input–Process–Output ที่ผู้เรียนได้ศึกษาในบทก่อนหน้า เมื่อเข้าใจโจทย์อย่างชัดเจนแล้ว จึงค่อยออกแบบอัลกอริทึมและแปลงเป็นโปรแกรมภาษาไพทอน

ตัวอย่างของปัญหาอย่างง่ายที่เหมาะกับการฝึกเขียนโปรแกรมในระดับเริ่มต้น ได้แก่ โปรแกรมคำนวณพื้นที่รูปเรขาคณิต โปรแกรมหาค่าเฉลี่ยคะแนน โปรแกรมตรวจสอบผลสอบ โปรแกรมแปลงหน่วยอุณหภูมิ โปรแกรมคำนวณส่วนลดสินค้า หรือโปรแกรมแสดงตารางสูตรคูณ ปัญหาเหล่านี้มีลักษณะไม่ซับซ้อนมากนัก แต่ช่วยให้ผู้เรียนฝึกใช้แนวคิดพื้นฐานครบถ้วน

ตัวอย่างเช่น หากโจทย์กำหนดให้เขียนโปรแกรมรับคะแนน 3 วิชา แล้วหาค่าเฉลี่ยและแสดงผลว่าผ่านหรือไม่ ผู้เรียนต้องวิเคราะห์ว่า ข้อมูลนำเข้าคือคะแนน 3 ค่า กระบวนการคือการหาผลรวมและหารด้วย 3 และผลลัพธ์คือค่าเฉลี่ยพร้อมข้อความว่า “ผ่าน” หรือ “ไม่ผ่าน” จากนั้นจึงเลือกใช้ตัวแปร คำสั่งรับข้อมูล คำสั่งคำนวณ และคำสั่งเงื่อนไขให้เหมาะสม กระบวนการเช่นนี้ช่วยให้ผู้เรียนเห็นความเชื่อมโยงระหว่างโจทย์กับโค้ดอย่างเป็นขั้นตอน

อีกตัวอย่างหนึ่ง เช่น โปรแกรมแสดงตัวเลข 1 ถึง 10 และหาผลรวมของตัวเลขทั้งหมด ผู้เรียนต้องใช้แนวคิดเรื่องการทำซ้ำและตัวแปรสะสมผลร่วมกัน การฝึกโจทย์ลักษณะนี้ช่วยให้เข้าใจว่าการทำซ้ำมิใช่เพียงการแสดงผลหลายครั้ง แต่ยังใช้เพื่อจัดการปัญหาที่เกี่ยวข้องกับข้อมูลหลายค่าได้ด้วย

ในการเรียนการสอน ผู้เขียนใช้ Google Colab เป็นพื้นที่ให้ผู้เรียนทดลองเขียนโปรแกรม แก้ไข และรันโปรแกรมได้สะดวก ขณะเดียวกัน ใช้ Python Tutor เพื่อช่วยให้ผู้เรียนติดตามลำดับการทำงานของโปรแกรมทีละขั้น โดยเฉพาะในโจทย์ที่เกี่ยวข้องกับการเปลี่ยนแปลงค่าของตัวแปรหรือการทำงาน

ของลูป เครื่องมือทั้งสองนี้ช่วยให้ผู้เรียนไม่เพียงเห็นผลลัพธ์สุดท้าย แต่เข้าใจเส้นทางที่นำไปสู่ผลลัพธ์นั้นด้วย

สำหรับนักศึกษาครู การเขียนโปรแกรมแก้ปัญหาอย่างง่ายควรถูกมองเป็นทั้งการฝึกทักษะทางวิทยาการคำนวณและการเตรียมความพร้อมสำหรับการสอน กล่าวคือ นักศึกษาครูควรสามารถคัดเลือกโจทย์ที่เหมาะสมกับวัยของผู้เรียน อธิบายวิธีวิเคราะห์โจทย์ได้อย่างชัดเจน และใช้เครื่องมืออย่าง Google Colab หรือ Python Tutor เพื่อช่วยลดความซับซ้อนในการเรียนรู้ของนักเรียน

กล่าวโดยสรุป การเขียนโปรแกรมแก้ปัญหาอย่างง่ายด้วยภาษาไพทอนเป็นกระบวนการสำคัญที่ช่วยให้ผู้เรียนบูรณาการความรู้พื้นฐานต่าง ๆ เข้าด้วยกัน และพัฒนาความสามารถในการวิเคราะห์ปัญหา ออกแบบวิธีการ และสร้างโปรแกรมที่มีความหมายในระดับเบื้องต้น

9.5 การออกแบบกิจกรรมการเรียนรู้เบื้องต้นสำหรับการสอนภาษาไพทอน

การออกแบบกิจกรรมการเรียนรู้เบื้องต้นสำหรับการสอนภาษาไพทอนควรคำนึงถึงธรรมชาติของผู้เรียนระดับเริ่มต้น ซึ่งมักยังไม่คุ้นเคยกับการเขียนโปรแกรมแบบข้อความ ผู้สอนจึงต้องเลือกแนวทางที่ช่วยลดความซับซ้อนของภาษา เชื่อมโยงเนื้อหากับประสบการณ์เดิม และเปิดโอกาสให้ผู้เรียนได้ทดลองปฏิบัติจริงอย่างต่อเนื่อง การเรียนรู้ภาษาไพทอนจะเกิดประสิทธิภาพสูงสุดเมื่อผู้เรียนไม่เพียงมองเห็นคำสั่ง แต่เข้าใจเหตุผลของคำสั่งและสามารถนำไปใช้แก้ปัญหาได้

หลักการสำคัญประการแรก คือ ควรเริ่มจากกิจกรรมที่ง่าย ชัดเจน และให้ผลลัพธ์รวดเร็ว เช่น การใช้ print() แสดงข้อความ การสร้างตัวแปรเก็บชื่อหรืออายุ หรือการรับข้อมูลอย่างง่ายผ่าน input() กิจกรรมเหล่านี้ช่วยสร้างความคุ้นเคยกับรูปแบบของภาษาไพทอน และทำให้ผู้เรียนเกิดความมั่นใจตั้งแต่เริ่มต้น

หลักการประการที่สอง คือ ควรเชื่อมโยงภาษาไพทอนกับแนวคิดที่ผู้เรียนเคยเรียนมาก่อน เช่น ผังงาน โครงสร้างแบบเรียงลำดับ แบบทางเลือก และแบบทำซ้ำ การให้ผู้เรียนเปรียบเทียบผังงานกับโค้ดไพทอน หรือเปรียบเทียบบล็อกคำสั่งใน Scratch กับคำสั่งในภาษาไพทอน จะช่วยให้ผู้เรียนมองเห็นความต่อเนื่องขององค์ความรู้และลดความรู้สึกว่าไพทอนเป็นเรื่องใหม่ที่ยากเกินไป

หลักการประการที่สาม คือ ควรใช้เครื่องมือที่สนับสนุนการเรียนรู้อย่างเหมาะสม ในบริบทนี้ Google Colab เป็นเครื่องมือที่มีประโยชน์มาก เพราะผู้เรียนสามารถเปิดใช้งานผ่านเว็บเบราว์เซอร์ได้ทันที เขียนโค้ด รันโปรแกรม และบันทึกงานได้สะดวก ส่วน Python Tutor เป็นเครื่องมือช่วยวิเคราะห์

การทำงานของโปรแกรมทีละขั้นตอน ซึ่งเหมาะอย่างยิ่งสำหรับการอธิบายตัวแปร เงื่อนไข และการทำซ้ำในระดับเริ่มต้น

รูปแบบกิจกรรมการเรียนรู้ที่เหมาะสมสำหรับการสอนภาษาไพทอนเบื้องต้น อาจประกอบด้วย

1) กิจกรรมทดลองเขียนและรันคำสั่งสั้น ๆ

เช่น การแสดงข้อความ การคำนวณอย่างง่าย หรือการรับข้อมูลแล้วแสดงผล ช่วยให้ผู้เรียนคุ้นเคยกับสภาพแวดล้อมของภาษา

2) กิจกรรมวิเคราะห์การทำงานของโปรแกรม

โดยใช้ Python Tutor ให้ผู้เรียนติดตามค่าของตัวแปรและลำดับการทำงานทีละขั้นตอน วิธีนี้ช่วยให้เข้าใจการทำงานภายในของโปรแกรมอย่างชัดเจน

3) กิจกรรมแก้ไขโปรแกรมที่ผิดพลาด

ให้ผู้เรียนวิเคราะห์ข้อผิดพลาดของโค้ดที่กำหนด เช่น ลืมแปลงชนิดข้อมูล เขียนเงื่อนไขผิด หรือเยื้องบรรทัดไม่ถูกต้อง กิจกรรมลักษณะนี้ช่วยพัฒนาทักษะการตรวจสอบและแก้ปัญหา

4) กิจกรรมออกแบบโปรแกรมจากสถานการณ์จริง

เช่น โปรแกรมคำนวณค่าใช้จ่าย โปรแกรมตัดเกรด หรือโปรแกรมคำนวณคะแนนรวม กิจกรรมเหล่านี้ช่วยให้ผู้เรียนเห็นความหมายของภาษาไพทอนในฐานะเครื่องมือแก้ปัญหา

สำหรับนักศึกษาครู การออกแบบกิจกรรมการสอนภาษาไพทอนควรเชื่อมโยงกับการจัดทำแผนการจัดการเรียนรู้ เช่น การกำหนดผลลัพธ์การเรียนรู้ การเลือกกิจกรรมที่เหมาะสมกับระดับของผู้เรียน การจัดลำดับเนื้อหา การเลือกใช้ Google Colab และ Python Tutor ให้สอดคล้องกับจุดประสงค์ และการกำหนดเกณฑ์ประเมินที่สะท้อนทั้งความรู้ ทักษะ และความสามารถในการคิดวิเคราะห์ของผู้เรียน

ในด้านการประเมินผล ครูควรใช้ทั้งการประเมินจากชิ้นงาน การสังเกตกระบวนการทำงาน การให้ผู้เรียนอธิบายโค้ดของตนเอง และการวิเคราะห์ลำดับการทำงานของโปรแกรม การประเมินลักษณะนี้จะช่วยให้เห็นว่าผู้เรียนเข้าใจจริงหรือเพียงคัดลอกคำสั่งตามตัวอย่างเท่านั้น

กล่าวโดยสรุป การออกแบบกิจกรรมการเรียนรู้เบื้องต้นสำหรับการสอนภาษาไพทอนควรมุ่งเน้นความเข้าใจเป็นลำดับขั้น ลดความซับซ้อนด้วยเครื่องมือที่เหมาะสม และเปิดโอกาสให้ผู้เรียนได้ลงมือปฏิบัติ วิเคราะห์ และสะท้อนผลการเรียนรู้ของตนเองอย่างต่อเนื่อง

บทสรุปประจำบทที่ 9

ภาษาไพทอนเป็นภาษาโปรแกรมที่เหมาะสำหรับการเรียนรู้พื้นฐานการเขียนโปรแกรม เนื่องจากมีรูปแบบคำสั่งที่เรียบง่าย อ่านง่าย และเอื้อต่อการพัฒนาทักษะการคิดเชิงตรรกะและการแก้ปัญหาอย่างเป็นระบบ ในบทนี้ได้อธิบายความรู้เบื้องต้นเกี่ยวกับภาษาไพทอน แนวคิดเรื่องตัวแปร ชนิดข้อมูล การรับ–แสดงผลข้อมูล การใช้คำสั่งควบคุมแบบเรียงลำดับ ทางเลือก และทำซ้ำ ตลอดจนการเขียนโปรแกรมแก้ปัญหาอย่างง่ายด้วยภาษาไพทอน

นอกจากนี้ ยังได้เน้นการใช้ Google Colab เป็นเครื่องมือในการรันโปรแกรม เพื่อให้ผู้เรียนเข้าถึงการใช้งานได้สะดวกโดยไม่ต้องติดตั้งซอฟต์แวร์เพิ่มเติม และใช้ Python Tutor เป็นเครื่องมือช่วยอธิบายลำดับขั้นตอนการทำงานของโปรแกรม ซึ่งช่วยให้ผู้เรียนเข้าใจการเปลี่ยนแปลงค่าของตัวแปรและการทำงานของโปรแกรมได้ชัดเจนมากขึ้น สำหรับนักศึกษาครู การเรียนรู้ภาษาไพทอนควรเชื่อมโยงทั้งกับการพัฒนาทักษะการเขียนโปรแกรมของตนเอง และกับการออกแบบกิจกรรมการเรียนรู้ที่เหมาะสมสำหรับผู้เรียนในระดับการศึกษาขั้นพื้นฐาน เพื่อให้การสอนภาษาไพทอนเป็นไปอย่างมีความหมาย เข้าใจง่าย และส่งเสริมทักษะวิทยาการคำนวณได้อย่างแท้จริง

คำถามท้ายบท

1. จงอธิบายลักษณะเด่นของภาษาไพทอน และเหตุใดภาษาไพทอนจึงเหมาะกับผู้เริ่มต้นเรียนเขียนโปรแกรม
2. Google Colab มีบทบาทอย่างไรในการสนับสนุนการเรียนรู้ภาษาไพทอน
3. Python Tutor มีประโยชน์อย่างไรต่อการทำความเข้าใจลำดับการทำงานของโปรแกรมภาษาไพทอน
4. จงอธิบายความหมายของตัวแปรและชนิดข้อมูลในภาษาไพทอน พร้อมยกตัวอย่าง
5. เพราะเหตุใดจึงต้องแปลงชนิดข้อมูลเมื่อรับค่าจาก input() ในบางกรณี
6. จงอธิบายความแตกต่างระหว่างโครงสร้างควบคุมแบบเรียงลำดับ ทางเลือก และทำซ้ำในภาษาไพทอน
7. หากต้องการเขียนโปรแกรมตรวจสอบผลสอบจากคะแนน ผู้เรียนควรใช้คำสั่งควบคุมประเภทใด เพราะเหตุใด
8. จงยกตัวอย่างโปรแกรมแก้ปัญหาอย่างง่ายด้วยภาษาไพทอนอย่างน้อย 2 ตัวอย่าง พร้อมอธิบายแนวคิดในการออกแบบ

9. หากท่านเป็นครูระดับมัธยมศึกษา ท่านจะออกแบบกิจกรรมการเรียนรู้ภาษาไพทอนโดยใช้ Google Colab อย่างไร
10. หากท่านเป็นครูผู้สอน ท่านจะใช้ Python Tutor เพื่อช่วยลดความสับสนของผู้เรียนเกี่ยวกับตัวแปรและลูปได้อย่างไร

เอกสารอ้างอิง

กระทรวงศึกษาธิการ. (2560). *ตัวชี้วัดและสาระการเรียนรู้แกนกลาง กลุ่มสาระการเรียนรู้วิทยาศาสตร์ (ฉบับปรับปรุง พ.ศ. 2560) ตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช 2551*. กรุงเทพมหานคร: ชุมนุมสหกรณ์การเกษตรแห่งประเทศไทย.

Downey, A. (2015). *Think Python: How to think like a computer scientist* (2nd ed.). Needham, MA: Green Tea Press.

Google. (2024). *Google Colaboratory*. Retrieved from the official Google Colab website.

Guo, P. J. (2013). Online Python Tutor: Embeddable web-based program visualization for CS education. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (pp. 579–584).

Python Software Foundation. (2024). *Python documentation*. Retrieved from the official Python documentation website.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

เอกสารแนบ

ภาษาไพทอน (Python) เป็นภาษาการเขียนโปรแกรมที่ได้รับความนิยมอย่างมากในปัจจุบัน เนื่องจากมีรูปแบบคำสั่งที่อ่านง่าย เขียนง่าย และเหมาะสำหรับผู้เริ่มต้นเรียนรู้การเขียนโปรแกรม อีกทั้งยังสามารถประยุกต์ใช้ได้หลากหลาย ทั้งด้านการคำนวณ การพัฒนาเว็บไซต์ การวิเคราะห์ข้อมูล ปัญญาประดิษฐ์ และงานอัตโนมัติต่าง ๆ

ในการเรียนรู้ภาษาไพทอนสำหรับผู้เริ่มต้น สิ่งสำคัญคือการทำความเข้าใจ **Syntax** หรือรูปแบบการเขียนคำสั่งให้ถูกต้อง เพราะคอมพิวเตอร์จะประมวลผลตามโครงสร้างของคำสั่งอย่างเคร่งครัด หากเขียนผิด แม้เพียงเล็กน้อย โปรแกรมอาจไม่สามารถทำงานได้ ดังนั้น เอกสารแนบฉบับนี้จึงจัดทำขึ้นเพื่อให้นักศึกษาได้ศึกษาไวยากรณ์พื้นฐานของภาษาไพทอน พร้อมตัวอย่างโจทย์และโค้ดประกอบ เพื่อใช้เป็นแนวทางในการฝึกปฏิบัติและพัฒนาทักษะการเขียนโปรแกรมอย่างเป็นขั้นตอน

1. ความรู้พื้นฐานเกี่ยวกับ Syntax ของภาษาไพทอน

1.1 การแสดงผลข้อมูลด้วย print()

คำสั่ง print() ใช้สำหรับแสดงข้อความหรือค่าตัวแปรออกทางหน้าจอ

รูปแบบคำสั่ง

print("ข้อความที่ต้องการแสดง")

ตัวอย่าง

```
print("Hello, Python")
print("ยินดีต้อนรับสู่การเขียนโปรแกรม")
```

ผลลัพธ์

```
Hello, Python
ยินดีต้อนรับสู่การเขียนโปรแกรม
```

1.2 การรับข้อมูลด้วย input()

คำสั่ง input() ใช้สำหรับรับข้อมูลจากผู้ใช้ทางแป้นพิมพ์

รูปแบบคำสั่ง

ตัวแปร = input("ข้อความแจ้งผู้ใช้: ")

ตัวอย่าง

```
name = input("กรุณาป้อนชื่อของคุณ: ")
print("สวัสดี", name)
```

หมายเหตุ

ค่าที่รับด้วย input() จะเป็นชนิดข้อความ (string) โดยอัตโนมัติ

1.3 ตัวแปร (Variable)

ตัวแปรใช้สำหรับเก็บข้อมูล เช่น ชื่อ อายุ คะแนน หรือผลลัพธ์จากการคำนวณ

ตัวอย่าง

```
name = "Nutthapat"
age = 40
score = 85.5
```

การตั้งชื่อตัวแปร

- ควรสื่อความหมาย
- ห้ามขึ้นต้นด้วยตัวเลข
- ห้ามเว้นวรรค
- สามารถใช้ตัวอักษร ตัวเลข และเครื่องหมาย _

ตัวอย่างชื่อตัวแปรที่ถูกต้อง

```
student_name = "Somchai"
score1 = 90
```

1.4 ชนิดข้อมูลพื้นฐาน

ภาษาไพทอนมีชนิดข้อมูลพื้นฐานที่พบบ่อย ดังนี้

ชนิดข้อมูล	ความหมาย	ตัวอย่าง
int	จำนวนเต็ม	10, 25, -5
float	จำนวนทศนิยม	3.14, 99.5
str	ข้อความ	"Hello", "Python"
bool	ค่าความจริง	True, False

ตัวอย่าง

```
x = 10
y = 5.5
name = "Python"
is_pass = True
```

1.5 การแปลงชนิดข้อมูล

บางครั้งจำเป็นต้องแปลงข้อมูลจากข้อความให้เป็นตัวเลขก่อนนำไปคำนวณ

ตัวอย่าง

```
age = int(input("กรุณาป้อนอายุ: "))
weight = float(input("กรุณาป้อนน้ำหนัก: "))
```

คำสั่งที่ใช้บ่อย

- int() แปลงเป็นจำนวนเต็ม
- float() แปลงเป็นจำนวนทศนิยม
- str() แปลงเป็นข้อความ

1.6 การคำนวณทางคณิตศาสตร์

เครื่องหมาย	ความหมาย	ตัวอย่าง
+	บวก	a + b
-	ลบ	a - b
*	คูณ	a * b
/	หาร	a / b
//	หารเอาส่วน	a // b
%	หารเอาเศษ	a % b
**	ยกกำลัง	a ** b

ตัวอย่าง

```
num1 = 10
num2 = 3
print(num1 + num2)
print(num1 - num2)
print(num1 * num2)
print(num1 / num2)
print(num1 // num2)
print(num1 % num2)
print(num1 ** num2)
```

1.7 การเปรียบเทียบ

ใช้เปรียบเทียบค่าระหว่างข้อมูล 2 ค่า โดยผลลัพธ์จะเป็น True หรือ False

เครื่องหมาย	ความหมาย
==	เท่ากับ
!=	ไม่เท่ากับ
>	มากกว่า
<	น้อยกว่า
>=	มากกว่าหรือเท่ากับ
<=	น้อยกว่าหรือเท่ากับ

ตัวอย่าง

```
x = 10
y = 20
print(x < y)
print(x == y)
```

1.8 การตัดสินใจด้วย if, elif, else

ใช้สำหรับตรวจสอบเงื่อนไขและเลือกทำงานตามเงื่อนไขที่กำหนด

รูปแบบคำสั่ง

```
if เงื่อนไข:
    คำสั่ง
elif เงื่อนไข:
    คำสั่ง
else:
    คำสั่ง
```

ตัวอย่าง

```
score = int(input("กรุณาป้อนคะแนน: "))
if score >= 80:
    print("เกรด A")
elif score >= 70:
    print("เกรด B")
elif score >= 60:
    print("เกรด C")
else:
    print("ควรพัฒนา")
```

1.9 การทำซ้ำด้วย for

ใช้เมื่อต้องการทำงานซ้ำตามจำนวนรอบที่กำหนด

ตัวอย่าง

```
for i in range(5):
    print("รอบที่", i)
```

ผลลัพธ์

```
รอบที่ 0
รอบที่ 1
```

รอบที่ 2

รอบที่ 3

รอบที่ 4

1.10 การทำซ้ำด้วย while

ใช้เมื่อต้องการทำงานซ้ำตราบใดที่เงื่อนไขยังเป็นจริง

ตัวอย่าง

```
count = 1
while count <= 5:
    print("จำนวน", count)
    count += 1
```

1.11 ความสำคัญของการย่อหน้า (Indentation)

ภาษาไพทอนใช้การย่อหน้าเพื่อกำหนดขอบเขตของคำสั่ง เช่น ภายใน if หรือภายในลูป หากย่อหน้าไม่ถูกต้อง โปรแกรมจะเกิดข้อผิดพลาด

ตัวอย่างที่ถูกต้อง

```
score = 80
if score >= 50:
    print("ผ่าน")
```

ตัวอย่างที่ผิด

```
score = 80
if score >= 50:
print("ผ่าน")
```

2. ตัวอย่างโจทย์พร้อมโค้ด

โจทย์ที่ 1 การแสดงข้อความทักทาย

โจทย์ จงเขียนโปรแกรมเพื่อแสดงข้อความว่า “ยินดีต้อนรับสู่ภาษาไพทอน”

โค้ดตัวอย่าง

```
print("ยินดีต้อนรับสู่ภาษาไพทอน")
```

โจทย์ที่ 2 รับชื่อและแสดงผล

โจทย์ จงเขียนโปรแกรมรับชื่อจากผู้ใช้ แล้วแสดงข้อความทักทาย

โค้ดตัวอย่าง

```
name = input("กรุณาป้อนชื่อ: ")
print("สวัสดี", name)
```

โจทย์ที่ 3 การหาผลบวกของจำนวน 2 จำนวน

โจทย์ จงเขียนโปรแกรมรับค่าจำนวนเต็ม 2 จำนวน แล้วแสดงผลรวม

โค้ดตัวอย่าง

```
num1 = int(input("ป้อนจำนวนที่ 1: "))
num2 = int(input("ป้อนจำนวนที่ 2: "))
result = num1 + num2
print("ผลรวม =", result)
```

โจทย์ที่ 4 การหาพื้นที่สี่เหลี่ยมผืนผ้า

โจทย์ จงเขียนโปรแกรมรับค่าความกว้างและความยาว แล้วคำนวณหาพื้นที่สี่เหลี่ยมผืนผ้า

โค้ดตัวอย่าง

```
width = float(input("ป้อนความกว้าง: "))
length = float(input("ป้อนความยาว: "))
area = width * length
print("พื้นที่สี่เหลี่ยมผืนผ้า =", area)
```

โจทย์ที่ 5 ตรวจสอบเลขคู่หรือเลขคี่

โจทย์ จงเขียนโปรแกรมรับจำนวนเต็ม 1 จำนวน แล้วตรวจสอบว่าเป็นเลขคู่หรือเลขคี่

โค้ดตัวอย่าง

```
number = int(input("ป้อนจำนวนเต็ม: "))
if number % 2 == 0:
    print("เป็นเลขคู่")
else:
    print("เป็นเลขคี่")
```

โจทย์ที่ 6 ตรวจสอบคะแนนสอบ

โจทย์ จงเขียนโปรแกรมรับคะแนนสอบ แล้วแสดงผลว่าผ่านหรือไม่ผ่าน โดยกำหนดให้ 50 คะแนนขึ้นไปถือว่าผ่าน

โค้ดตัวอย่าง

```
score = int(input("ป้อนคะแนน: "))
if score >= 50:
    print("ผ่าน")
else:
    print("ไม่ผ่าน")
```

โจทย์ที่ 7 แสดงตัวเลข 1 ถึง 10 ด้วย for

โจทย์ จงเขียนโปรแกรมแสดงตัวเลขตั้งแต่ 1 ถึง 10

โค้ดตัวอย่าง

```
for i in range(1, 11):
    print(i)
```

โจทย์ที่ 8 หาผลรวมของตัวเลข 1 ถึง 5

โจทย์ จงเขียนโปรแกรมหาผลรวมของตัวเลขตั้งแต่ 1 ถึง 5

โค้ดตัวอย่าง

```
sum_num = 0
for i in range(1, 6):
    sum_num += i

print("ผลรวม =", sum_num)
```

โจทย์ที่ 9 แม่สูตรคูณ

โจทย์ จงเขียนโปรแกรมรับตัวเลข 1 จำนวน แล้วแสดงสูตรคูณแม่ของจำนวนนั้นตั้งแต่ 1 ถึง 12

โค้ดตัวอย่าง

```
number = int(input("ป้อนแม่สูตรคูณ: "))

for i in range(1, 13):
    print(number, "x", i, "=", number * i)
```

โจทย์ที่ 10 นับถอยหลังด้วย while

โจทย์ จงเขียนโปรแกรมนับถอยหลังจาก 5 ถึง 1

โค้ดตัวอย่าง

```
count = 5

while count >= 1:
    print(count)
    count -= 1
```

3. แบบฝึกหัดเพิ่มเติมสำหรับฝึกปฏิบัติ

1. เขียนโปรแกรมรับอายุของผู้ใช้ แล้วแสดงผลว่าอีก 5 ปี จะมีอายุเท่าไร
2. เขียนโปรแกรมรับรัศมีของวงกลม แล้วคำนวณหาพื้นที่วงกลม
3. เขียนโปรแกรมรับจำนวนเต็ม 1 จำนวน แล้วตรวจสอบว่าเป็นจำนวนบวก จำนวนลบ หรือศูนย์
4. เขียนโปรแกรมแสดงเลขคู่ตั้งแต่ 2 ถึง 20
5. เขียนโปรแกรมหาผลคูณของตัวเลขตั้งแต่ 1 ถึง 5

4. สรุป

การเขียนโปรแกรมภาษาไพทอนสำหรับผู้เริ่มต้นควรเริ่มจากการเรียนรู้ Syntax พื้นฐาน ได้แก่ การแสดงผล การรับข้อมูล ตัวแปร ชนิดข้อมูล การคำนวณ การเปรียบเทียบ การตัดสินใจ และการทำซ้ำ เมื่อเข้าใจหลักการเหล่านี้แล้ว ผู้เรียนจะสามารถพัฒนาโปรแกรมที่มีความซับซ้อนมากขึ้นได้ต่อไป

การฝึกเขียนโปรแกรมจากโจทย์ง่าย ๆ พร้อมดูตัวอย่างโค้ด จะช่วยให้เกิดความเข้าใจทั้งในเชิงโครงสร้างคำสั่งและการประยุกต์ใช้จริง ดังนั้น ผู้เรียนควรฝึกทดลองแก้ไขโค้ด เปลี่ยนค่า และสังเกตผลลัพธ์ เพื่อสร้างความเข้าใจอย่างลึกซึ้งและพัฒนาทักษะการคิดเชิงตรรกะควบคู่กันไป

5. ข้อเสนอแนะในการเรียนรู้เพิ่มเติม

- ฝึกพิมพ์โค้ดด้วยตนเองแทนการคัดลอก
- ทดลองเปลี่ยนค่าตัวแปรและเงื่อนไขเพื่อดูผลลัพธ์
- ตรวจสอบการย่อหน้าให้ถูกต้องทุกครั้ง
- ฝึกอ่านข้อผิดพลาด (Error Message) เพื่อเรียนรู้การแก้ปัญหา
- เริ่มจากโจทย์ง่าย แล้วค่อยเพิ่มความซับซ้อน

เอกสารแนบฉบับนี้สามารถใช้เป็นพื้นฐานสำหรับการเรียนรู้ภาษาไพทอนในบทเริ่มต้น และใช้ต่อยอดสู่การเขียนโปรแกรมในระดับที่สูงขึ้นได้ต่อไป