

ความหมายของ codes ในโปรแกรม python ทางคณิตศาสตร์

นางสาวอนาดี มะหะหมัด

1 มีนาคม 2567

บทที่ 1

ที่มาและความสำคัญของ โปรแกรม python

1.1 ที่มา

Python ถูกคิดค้นครั้งแรกในปี 1989 โดย Guido van Rossum ชาวเนเธอร์แลนด์ ซึ่งสำเร็จการศึกษาระดับปริญญาโทสาขาคณิตศาสตร์และวิทยาการคอมพิวเตอร์จาก University of Amsterdam ประเทศเนเธอร์แลนด์ เขาเคยได้เข้าร่วมทำงานกับ Centrum Wiskunde and Informatica (CWI) ประเทศเนเธอร์แลนด์, National Institute of Standards and Technology (NIST) และ Corporation for National Research Initiatives (CNRI) ประเทศสหรัฐอเมริกา และได้ร่วมงานกับบริษัท Google ตั้งแต่ปี 2005 จนถึงปี 2012 และย้ายไปทำงาน ที่ Dropbox ในปี 2013 จนกระทั่งเกษียณตัวเองในเดือนตุลาคม ปี 2019 และได้กลับเข้ามาเริ่มงานอีกครั้งกับ Microsoft เมื่อปลายปี 2020 โดยมีเป้าหมายหลักก็คือการพัฒนาให้ Python ทำงานได้เร็วขึ้นอย่างน้อย 5 เท่าภายในระยะเวลา 4 ปี ทั้งนี้ที่มาของชื่อภาษาโปรแกรมไพธอน มา จากตอนที่ Van Rossum กำลังเริ่มพัฒนาภาษาโปรแกรมไพธอน เขาได้อ่านสคริปต์ของรายการทีวี “Monty Python’s Flying Circus” ซึ่งเป็นซีรีส์ตลกของช่อง BBC ในยุค 1970s ทำให้เขาได้คิดว่า เขาต้องการชื่อที่สั้น มีเอกลักษณ์ และค่อนข้างลึกลับ น่าค้นหา เขาจึงได้ตัดสินใจใช้ชื่อ Python สำหรับ ภาษาโปรแกรมที่เขากำลังพัฒนา ภาษาโปรแกรมไพธอนเป็นภาษาระดับสูงที่ได้รับการพัฒนาจากการผสมผสานความหลากหลายของ ภาษาอื่น ๆ เช่น ABC, Modula-3, Icon, Perl, Lisp, Smalltalk เป็นต้น นอกจากนี้ยังมีความสามารถ ในการจัดการหน่วยความจำแบบอัตโนมัติ รวม

ไปถึงการจัดการในเรื่องของตัวแปรที่สร้างขึ้นมาใช้งาน โดยไม่ต้องกำหนดชนิดข้อมูล เขาจึงพยายามสร้างภาษาขึ้นมาใหม่ที่เข้าใจง่ายและไม่ซับซ้อน จึงเป็นจุดเริ่มต้นให้เขาคิดค้นภาษา Python ขึ้นมานั่นเอง โดย Guido คัดแปลงรูปแบบภาษาบางอย่างมาจากภาษา ABC มาพัฒนาลงในภาษา Python เขาเริ่มเผยแพร่ Python 1.0 เวอร์ชันแรกในปี 1994 ซึ่งเป็นภาษาที่มีอายุมากกว่าภาษา Java ที่เกิดขึ้นในเวลาต่อมาในปี 1996[1]

1.2 ความสำคัญของโปรแกรม python

Python มีประโยชน์สำหรับการเขียนโค้ดฝั่งเซิร์ฟเวอร์ เป็นภาษาโปรแกรมพื้นฐานที่นำไปต่อยอดได้หลายรูปแบบ เรียกว่าอยู่ที่จะทำให้ทำอะไรมากกว่า เนื่องจากมีความยืดหยุ่นคล่องตัวสูง ทั้งยังมี Tools และ Library Support ฟรีเยอะ หาข้อมูลได้ง่าย แต่ที่นิยมนำไปใช้งานอย่างแพร่หลายเนื่องจากมีไลบรารีจำนวนมากที่ประกอบด้วยโค้ดที่เขียนไว้ล่วงหน้าสำหรับฟังก์ชันแบ็คเอนด์ที่ซับซ้อน นักพัฒนายังใช้เฟรมเวิร์ก Python ที่หลากหลายซึ่งมีเครื่องมือที่จำเป็นทั้งหมดเพื่อสร้างเว็บแอปพลิเคชันได้เร็วขึ้นและง่ายขึ้นอีกด้วย ตัวอย่างเช่น นักพัฒนาสามารถสร้างโครงสร้างเว็บแอปพลิเคชันได้ภายในไม่กี่วินาที เนื่องจากไม่จำเป็นต้องเขียนขึ้นใหม่ทั้งหมด จากนั้นนักพัฒนาสามารถทดสอบได้โดยใช้เครื่องมือทดสอบของเฟรมเวิร์ก โดยไม่ต้องพึ่งพาเครื่องมือทดสอบภายนอก[2]

บทที่ 2

codes ที่มีความสัมพันธ์ที่จะ สามารถไปสร้างแบบจำลองทาง คณิตศาสตร์

1. คำสั่ง `abs()` ใช้สำหรับหาค่าจำนวนเต็มบวก
2. คำสั่ง `float()` ใช้ในกรณีที่ต้องการแปลงชนิดข้อมูลจำนวนเต็ม ให้เป็นชนิดข้อมูลจำนวนทศนิยม
3. คำสั่ง `complex()` ใช้ในกรณีที่ต้องการแปลงเป็นชนิดข้อมูลจำนวนเชิงซ้อนให้เรียกใช้งาน
4. คำสั่ง `int()` ใช้ในกรณีที่ต้องการแปลงเป็นชนิดข้อมูลจำนวนเต็มให้เรียกใช้งาน
5. คำสั่ง `math.modf(x)` แยกจำนวนเต็มและเศษของจำนวนจริงออกจากกัน `x` คือจำนวนจริง
6. คำสั่ง `isalnum()` คืนค่า `True` ถ้าสตริงประกอบด้วยตัวอักษรปนกับตัวเลข หรือแค่ตัวอักษรตัวเลขอย่างเดียวหนึ่งเพียงอย่างเดียวถ้ามีสัญลักษณ์อื่น หรือช่องว่างจะคืนค่า `False`
7. คำสั่ง `isalpha()` คืนค่า `True` ถ้าสตริงประกอบด้วยตัวอักษรทั้งหมด ถ้ามีสัญลักษณ์ ตัวเลข หรือช่องว่างจะคืนค่า `False`
8. คำสั่ง `isdigit()` คืนค่า `True` ถ้าสตริงประกอบด้วยตัวเลขทั้งหมด ถ้ามีตัวอักษรสัญลักษณ์ หรือช่องว่าง จะคืนค่า `False`

9. คำสั่ง `isdecimal()` ใช้สำหรับตรวจสอบตัวเลขแสดงผลลัพธ์เหมือน
เมธอด `isdigit()`
10. คำสั่ง `isnumber()` ใช้สำหรับตรวจสอบตัวเลข แสดงผลลัพธ์เหมือน
เมธอด `isdigit()`
11. คำสั่ง `append()` เพิ่มข้อมูลต่อท้ายลิสต์
12. คำสั่ง `len()` ฟังก์ชันที่ใช้แสดงจำนวนข้อมูลที่มีอยู่ในลิสต์
13. คำสั่ง `math.sqrt(x)` ค่ารากที่สองของ x คือ จำนวนจริง ที่ไม่ต่ำกว่า 0
14. คำสั่ง `math.copysign(x,y)` สลับค่าของ x เป็นจำนวนจริงและใส่
เครื่องหมายตามค่าของ y คือ จำนวนเต็มหรือจำนวนจริง y คือ จำนวนเต็ม
หรือจำนวนจริง
15. คำสั่ง `divmod()` จะคืนค่าผลหารและเศษจากการหารออกมา
16. คำสั่ง `enumerate()` ใช้สำหรับกำหนดหมายเลขแบบเรียงลำดับ
17. คำสั่ง `range()` ใช้สำหรับสร้างกลุ่มข้อมูลตัวเลขจำนวนเต็ม
18. คำสั่ง `round()` ใช้สำหรับปัดเศษจำนวนทศนิยม
19. คำสั่ง `math.exp(x)` ค่า e ยกกำลัง x
20. คำสั่ง `math.floor(x)` หาจำนวนเต็มที่มีค่ามากที่สุด
21. คำสั่ง `math.factorial(x)` หาค่า factorial ของ x
22. คำสั่ง `math.gcd(x,y)` หาค่าตัวหารร่วมมากที่มีค่ามากที่สุด
23. คำสั่ง `math.fmod(x,y)` หาค่าเศษจากการหาร
24. คำสั่ง `math.fsum(x)` หาค่าผลรวมจากชนิดข้อมูลแบบเรียงลำดับ

Bibliography

- [1] Ratthaprom Promkam. *Programming Fundamentals with python*, pp. 7–8. DOI: https://epsilonxe.github.io/RMUTT_09114204/materials/lecture_python.pdf.
- [2] Ratthaprom Promkam. *Programming Fundamentals with python*, pp. 12–13. DOI: https://epsilonxe.github.io/RMUTT_09114204/materials/lecture_python.pdf.