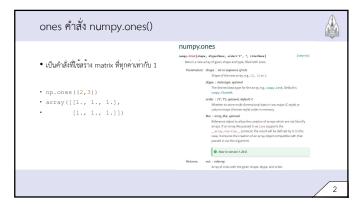
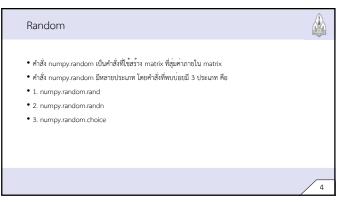


4



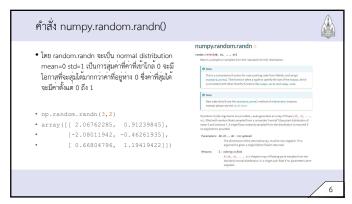
Matrix Operation (scalar multiplication) • scalar multiplication คือ การคูณค่าคงที่ 1 ค่าที่เป็นเลขตัวเดียวคูณเข้าใน matrix เช่น • M one = np.ones((2,3)) • มุมมองค่าภายใน • array([[1., 1., 1.], [1., 1., 1.]]) • 2*M one • ผลลัพธ์จะได้ • array([[2., 2., 2.], [2., 2., 2.]])

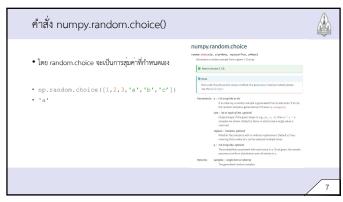
3

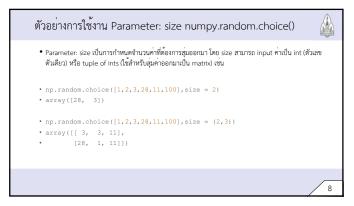




5 6





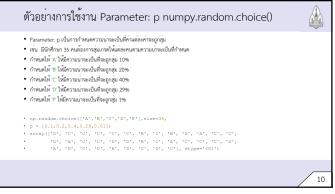


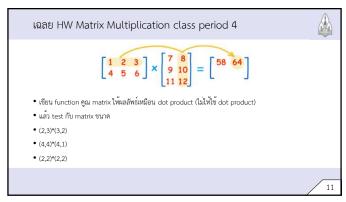
ตัวอยางการใช้งาน Parameter: replace numpy.random.choice()

• Parameter: replace การใส่คืน เป็นการกำหนดว่า จะให้นำค่าที่สุ่มออกมาใส่คืนหรือไม่ใส่คืน คือ การสุ่มค่า ช้ำหรือไม่เข้า
• โดยค่า default ของ replace = True คือการสุ่มแบบใส่คืนมีค่าข้ำกันได้ ถ้าต้องการสุ่มแบบไม่ใส่คืน คือให้ไม่ ค่าข้ำกัน ให้กำหนด replace = Fasle

• np.random.choice([1,2,3,28,11,100], size = (2,3), replace=False)
• array([[3, 11, 2],
[100, 28, 1]])

9 10





```
      การสร้างฟังก์ชั่น คูณ matrix

      • def mat_mul(A,B):

      • C = np.zeros((A.shape[0],B.shape[1]))

      • for r_a in range(A.shape[0]):

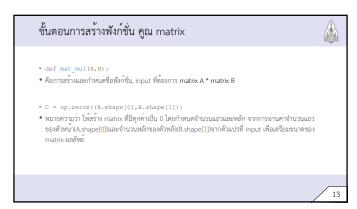
      • for c_b in range(B.shape[1]):

      • for every_element in range(A.shape[1]):

      • C[r_a,c_b] = C[r_a,c_b] + (A[r_a,every_element] *

      B[every_element,c_b])

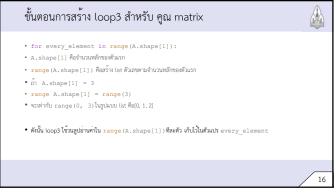
      • return C
```





ขั้นตอนการสร้าง loop2 สำหรับ คูณ matrix
 for c_b in range (B.shape[1]):
 B.shape[1] คือจำนวนหลักของตัวหลัง
 range (B.shape[1]) คือสาวา แร่ ตัวเลขตามจำนวนหลักของตัวหลัง
 ถ้า B.shape[1] = 2
 range B.shape[1] = range (2)
 จะเท่ากับ range (0, 2)ในรูปแบบ แร่ คือ[0, 1]
 คังนั้น loop2 ใช้วันสูปล่านค่าใน range (B.shape[1]) ที่ละตัว เก็บไว้ในตัวแปร c_b

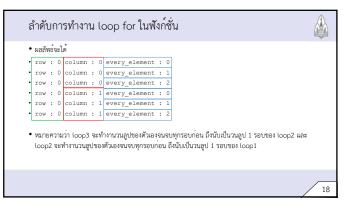
15 16



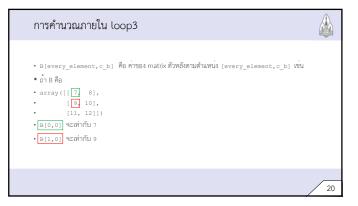
61คับการทำงาน loop for ในฟังก์ชั่น

• Loop1 คือ for r_a in range(A.shape[0]):
• Loop2 คือ for c_b in range(B.shape[1]):
• Loop3 คือ for every_element in range(A.shape[1]):
• ก
• range(A.shape[0]) = [0, 1]
• range(B.shape[1]) = [0, 1]
• range(A.shape[1]) = [0, 1, 2]
• Loop1
• Loop2
• Loop3
• print(f'row : {r_a} column : {c_b} every_element : {every_element}')

17 18







การคำนวณการคูณ matrix

• ดังนั้น

• ([r_a,c_b] = C[r_a,c_b] + (A[r_a,every_element] * B[every_element,c_b])

• ในการทำงานรอบแรกของ loop1

• ([r_a,c_b] จะเป็นด้วนปรที่เข้ามารับค่าผลลังธ์จากการคำนวณเพื่อนำค่าไปบวกใน loop รอบถัดไป เช่น

• ([0,0] = ([0,0] + (A[0,0] * B[0,0]))

• ([0,0] = 0.0 + (1 * 7))

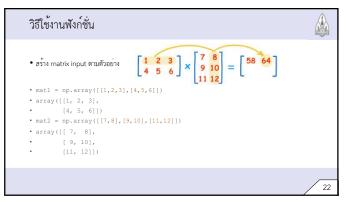
• ([0,0] = 7.0 และเอาไปคำนวณใน loop รอบถัดไป

• ([0,0] = ([0,0] + (A[0,1] * B[1,0]))

• ([0,0] = 7.0 + (2 * 9))

• ([0,0] = 25.0 และเอาไปคำนวณใน loop รอบถัดไปจนจบ ([0,0] และไปเริ่ม ([0,1])

21 22



ผลลัพธ์ของพังก์ชั่น
 • mat_mul(mat1,mat2)ชื่อฟังก์ชั่นตามด้วย input(A,B) ที่ตั้งไว้
 • ผลลัพธ์จะได้
 • array([[58., 64.],
 [139., 154.]])
 • ซึ่งถ้าเปรียบเพียบตัวแปรในฟังก์ชั่นคือตัวแปรที่เข้ามารับผลลัพธ์การคำนวณในแต่ละรอบของ loop1
 • [[c[0,0], c[0,1]],
 • [c[1,0], c[1,1]]]

23 24