Unit 6 ເງື່ອນໄຂ ແລະ ການຕັດສິນໃຈ (Conditional Statement-1: Conditions and Decision Making)

Unit Objective

- ✓ ເພື່ອເຂົ້າໃຈເຖິງຄວາມຈຳເປັນ ແລະ ການໃຊ້ຄຳສັ່ງກວດສອບເງື່ອນໄຂ.
- ✓ ເພື່ອເຂົ້າໃຈເຖິງຄວາມແຕກຕ່າງຂອງການໃຊ້ຄຳສັ່ງຄວບຄຸມຕາມເງື່ອນໄຂ, ຄຳສັ່ງວິນຊ້ຳແລະ ຄຳສັ່ງເຮັດວຽກຕາມລຳດັບ.
- 🗸 ເພື່ອເຂົ້າໃຈເຖິງ syntax ຂຽນຄຳສັ່ງ if ແລະ ໂຄງສ້າງຄຳສັ່ງຄວບຄຸມແບບສັບຊ້ອນ.
- 🗸 ເພື່ອໃຫ້ເຂົ້າໃຈເຖິງ syntax ການຂຽນ Block ຄຳສັ່ງ ແລະ ການຫຍໍ້ໜ້າ Block ຄຳສັ່ງ.

Course Overview

- ✓ ຮຽນຮູ້ກ່ຽວກັບຄຳສັ່ງຄວບຄຸມການເຮັດວຽກຂອງໂປຣແກຣມທີ່ເຮັດວຽກຕາມລຳດັບ ແລະ ຕາມເງື່ອນໄຂ.
- ✓ ການໃຊ້ຄຳສັ່ງ if ຂຽນໂປຣແກຣມເພື່ອເຮັດວຽກທີ່ແຕກຕ່າງກັນຕາມເງື່ອນໄຂທີ່ກຳນິດ.
- √ ຮຽນຮູ້ວິທີການໃຊ້ສຳນວນເງື່ອນໄຂ ເພື່ອໃຫ້ໄດ້ຄ່າຄວາມຈິງ True/False.

Concepts You Will Need to Know From Previous Units

- ✓ ເຂົ້າໃຈກ່ຽວກັບປະເພດຂໍ້ມູນຂອງຕົວປ່ຽນ, ການປະກາດຕາມປະເພດຂໍ້ມູນທີ່ຕ້ອງການໂດຍປະສະຈາກຄວາມຜິດພາດ.
- ✓ ຮັບຄ່າຂໍ້ມູນຈາກຜູ້ໃຊ້ດ້ວຍຟັງຊັນ input() ແລະ ສະແດງຄ່າຂໍ້ມູນດ້ວຍຟັງຊັນ print().
- ✓ ປະຕິບັດການຄິດໄລ່ຕ່າງໆໂດຍການໃຊ້ຕົວດຳເນີນການທາງຄະນິດສາດ, ຕົວດຳເນີນການປຽບທຽບ ແລະ ຕັກກະສາດ.

Keywords

ຄຳສັ່ງປະມວນຜົນຕາມ ລຳດັບ (Sequential Statement) ຄຳສັ່ງເງື່ອນໄຂ (Conditional Statement)

ຄຳສັ່ງ(if)

ຂໍ້ມູນຄ່າຄວາມຈິງ (Boolean Data Type) ສຳນວນທີ່ມີເງື່ອນໄຂ (Conditional Expression)

Unit 6 ຄຳສັ່ງເງື່ອນໄຂ (Conditional Statement-1)

Mission

1. Real-world Problem

1.1. ຫວຍແມ່ນຫຍັງ?



- > ຫວຍ (Lottery) ແມ່ນເກມແບບໜຶ່ງທີ່ມີການຊື້ດ້ວຍເງິນ ແລະ ຈະໃຫ້ລາງວັນແກ່ຜູ້ຊື້ ຖ້າຫາກຖືກຕ້ອງຕາມເງື່ອນໄຂທີ່ກຳນຶດ.
- ຫວຍມີການນຳໃຊ້ໃນສັງຄືມໃນທາງບວກ, ແຕ່ກໍມີຜົນກະທົບ ຫຼາຍເຊັ່ນກັນ. ດັ່ງນັ້ນ, ປົກກະຕິຈະຄຸ້ມຄອງບໍລິຫານດ້ວຍລັດ ຖະ ບານ ຫຼື ມອບໃຫ້ສະຖາບັນເອກະຊົນ.
- ໃນບາງປະເທດ ບຸກຄົນ ຫຼື ບໍລິສັດຖືກລົງໂທດສໍາລັບການພິມ ໃບຫວຍໂດຍບໍ່ໄດ້ຮັບອະນຸຍາດ.
- ເຖິງແນວໃດກໍຕາມ, ມັນກໍມີປະໂຫຍດໃນເລື່ອງຂອງການລະດົມ ທຶນສໍາລັບບໍລິການສາທາລະນະ ແລະ ບໍລິຈາກເພື່ອການທໍາບຸນ. ດັ່ງນັ້ນ, ມີບາງຄົນເວົ້າວ່າ ຫວຍ ອາດຈະເປັນເກມທີ່ດີ ຖ້ານໍາເອົາ ລາຍໄດ້ໄປຊ່ວຍເຫຼືອສັງຄົມ.

1. Real-world Problem

1.2. ການຈຳລອງຫວຍດ້ວຍເລກຈຳນວນເຕັມ 3 ຕົວເລກ

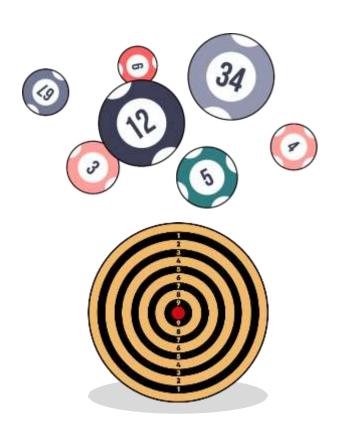


- ໃນໂຈດບັນຫານີ້ ພວກເຮົາຈະຂຽນໂປຣແກຣມແຈ້ງໃຫ້ທ່ານຊາບເມື່ອຖືກຫວຍເລກສາມຕົວ.
- ໂຈດບັນຫານີ້ ມີເລກໂຊກດີ ຄື: 2, 3 ແລະ 9 ເຊິ່ງຜູ້ໃຊ້ບໍ່ ຮູ້ຕືວເລກເຫຼົ່ານີ້, ໂດຍໃຫ້ຜູ້ໃຊ້ປ້ອນຕີວເລກເຂົ້າມາ 3 ຕີວ ເລກ.
- ໃນທີ່ນີ້, ຜູ້ໃຊ້ສາມາດປ້ອນຕີວເລກເຂົ້າມາເທື່ອລະຕິວ
 ແລະ ແຍກກັນດ້ວຍເຄື່ອງໝາຍຍະຫວ່າງ. ຖ້າຜູ້ໃຊ້ປ້ອນ ຕີວເລກຖືກຕ້ອງກັບຕີວເລກໂຊກດີ ເຊັ່ນ: 2 3 9 ຫຼື 9 3
 2 ໂປຣແກຣມກໍຈະສະແດງຂໍ້ຄວາມວ່າ: ທ່ານເປັນຜູ້ໂຊກ ດີ.
- ຈາກໂຈດບັນຫານີ້ ເຮົາຈະໄດ້ຮຽນຮູ້ຫຼັກການຂອງໂປຣແກຣມທີ່ເຮັດວຽກພາຍໃຕ້ຄຳສັ່ງເງື່ອນໄຂທີ່ສັບຊ້ອນ ython 304

Mission Unit 6

1. Real-world Problem

1.3. ຄວາມເປັນມາຂອງຫວຍ (History of lottery)



- ປະຫວັດຄວາມເປັນມາຂອງຫວຍແມ່ນເກີດຂຶ້ນມາດິນນານ
 ແລ້ວ ແລະ ມີການບັນທຶກໄວ້ວ່າ: ຕ້ອງຈ່າຍຄ່າກໍ່ສ້າງກຳແພງ
 ເມືອງຈິນ ເມື່ອ 2,200 ປີກ່ອນ ໃນສະໄໝລາດຊະວົງຊິນຂອງ
 ຈີນ.ນອກຈາກນີ້ຍັງທີການບັນທຶກການກະທຳທີ່ຄ້າຍຄືກັບການ
 ບັນທຶກຫວຍ
- ມີຫຼາຍວິທີໃນການຈັບສະຫຼາກ ແລະ ສອງວິທີທີ່ນິຍົມກໍຄື:
 ການປ່ອນໜ່ວຍບານທີ່ມີຕົວເລກໃສ່ກ່ອງ, ແລ້ວຈຶກອອກມາ ດ້ວຍເຄື່ອງຈັກ ແລະ ວິທີການຍິງທະນູໃສ່ແຜ່ນໜູນທີ່ມີຕົວ ເລກ.
- ຫວຍຈະອອກ ໃນທຸກວັນ/ທຸກອາທິດ/ທຸກໆເດືອນ ແລະ ບາງເທື່ອກໍຈະຈັດຂຶ້ນເນື່ອງ ໃນວາລະຄົບຮອບວັນສໍາຄັນຂອງຊາດ.

Mission Unit 6

- 1. Real-world Problem
- 1.4. ລະບົບຫວຍຕ່າງໆ
 - ເຮົາສາມາດກວດເບິ່ງຜົນຫວຍ ແລະ ລາງວັນໄດ້ທຸກໆອາທິດ, ໜຶ່ງໃນລະບົບຫວຍທີ່ມີຊື່ສຽງທີ່ສຸດຂອງໂລກ
 ຈາກເວັບໄຊ lotteryusa (www.lotteryusa.com)



https://www.lotteryusa.com/

- ໃນລະບົບຫວຍນີ້ ລາງວັນຈະຖືກກຳນົດໂດຍ ໝາຍເລກທີ່ກົງກັນຈາກ 6 ໝາຍເລກທີ່ຜູ້ໃຊ້ ເລືອກ. ນອກຈາກນີ້ຍັງສາມາດກວດຜົນຫວຍໄດ້ ຈາກລະບົບ Mega Millions lottery ໄດ້ອີກ ດ້ວຍ ເຊິ່ງຈະມີລັກສະນະຄ້າຍຄືກັນ.
- ໃນລັດ ທາມີລ ນາດູ ຂອງ ປະເທດອີນເດຍ ໄດ້ ສັ່ງຫ້າມບໍ່ໃຫ້ມີການຂາຍຫວຍທຸກຮູບແບບຈືນ ຮອດປີ 2003, ແຕ່ປະຈຸບັນແມ່ນອະນຸຍາດແລ້ວ.
- ບາງປະເທດລວມທັງປະເທດ ເກົາຫຼີ ຈະຕ້ອງເສຍພາສີຫຼາຍກວ່າ 30% ຂອງຜູ້ຖືກລາງວັນ.

- 1. Real-world Problem
- 1.5. ບະບົບແນະນຳເລກຫວຍໃນຮູບແບບເວັບໄຊ
 - ມີລະບົບຫວຍຫຼາຍປະເພດທີ່ແຕກຕ່າງກັນໄປໃນແຕ່ລະປະເທດ ແລະ ມີເວັບໄຊທີ່ແນະນຳເລກຫວຍ.
 - ດັ່ງໃນຮູບລຸ່ມນີ້ ສະແດງວິທີການໃຊ້ເວັບໄຊແນະນຳໝາຍເລກຫວຍ.



The site that predicts lottery numbers.



Mission Unit 6

2. Real-world Problem

2.1. ເຄື່ອງຈັກອອກລາງວັນຫວຍເຮັດວຽກແນວໃດ?

```
1 n1, n2, n3 = 2, 3, 9
 3 | n = 0
 4 a, b, c = input('Enter a three-digit lottery number: ').split()
 5 \mid a = int(a)
 6 b = int(b)
   c = int(c)
   if a == n1 or a == n2 or a == n3: # If one of the numbers match the winning number
        n += 1
                                         #Increase n
   if b == n1 or b == n2 or b == n3:
        n += 1
13 | if c == n1 or c == n2 or c == n3:
        n += 1
15
16 | if n == 3:
                                         # If n is 3, three numbers matched
        print('You won the lottery!')
```

Enter a three-digit lottery number: You won the lottery!

2. Real-world Problem

2.2. ກິນໄກຂອງເຄື່ອງຈັກຫວຍອັດຕະໂນມັດ

```
1 | n1, n2, n3 = 2, 3, 9
 3 | n = 0
 4 a, b, c = input('Enter a three-digit lottery number: ').split()
 5 \mid a = int(a)
 6 b = int(b)
   c = int(c)
   if a == n1 or a == n2 or a == n3: # If one of the numbers match the winning number
        n += 1
                                         #Increase n
11 | if b == n1 or b == n2 or b == n3:
        n += 1
13 | if c == n1 or c == n2 or c == n3:
        n += 1
15
16 | if n == 3:
                                         # If n is 3, three numbers matched
        print('You won the lottery!')
```

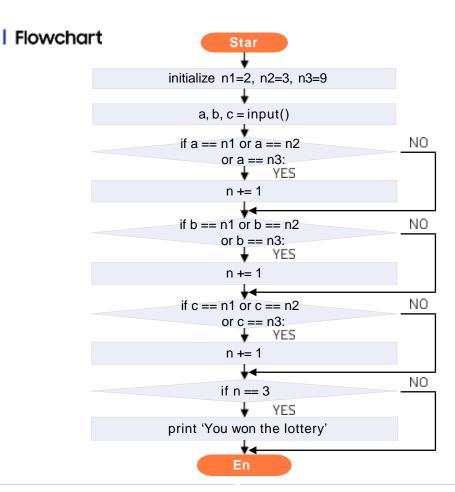
Enter a three-digit lottery number: 1, 2, 9

2. Real-world Problem

2.3. ແຜນຜັງການຂຽນໂປຣແກຣມ

Pseudocode

1 Start 2 Generate three numbers, 2, 3, 9. 3 Reset with n = 0. Receive input for three numbers a, b, c. 5 if a matches one of 2, 3, 9, then { Add 1 to n } [4] if b matches one of 2, 3, 9 then { Add 1 to n } 9 if c matches one of 2, 3, 9 then { 10 Add 1 to n } 11 if n is 3, print 'You won the lottery!' 12 End



Mission Unit 6

2. Real-world Problem

2.4. ໂປຣແກຣມເຄື່ອງຈັກຫວຍອັດຕະໂນມັດ

```
1 | n1, n2, n3 = 2, 3, 9
3 n = 0
 4 a, b, c = input('Enter a three-digit lottery number: ').split()
 5 \mid a = int(a)
 6 b = int(b)
   c = int(c)
   if a == n1 or a == n2 or a == n3: # If one of the numbers match the winning number
        n += 1
                                        #Increase n
11 if b == n1 or b == n2 or b == n3:
        n += 1
13 | if c == n1 or c == n2 or c == n3:
        n += 1
15
16 if n == 3:
                                        # If n is 3, three numbers matched
        print('You won the lottery!')
```

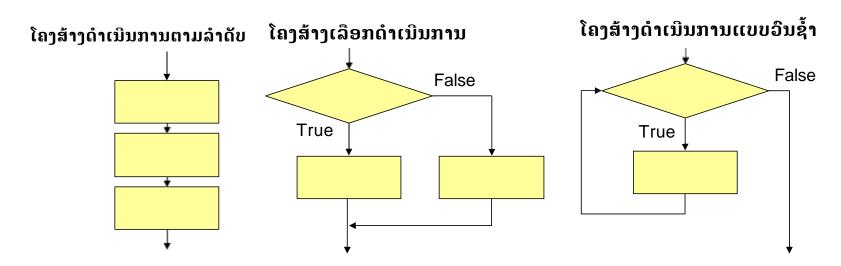
Unit 6 ຄຳສັ່ງເງື່ອນໄຂ (Conditional Statement-1)

Key concept

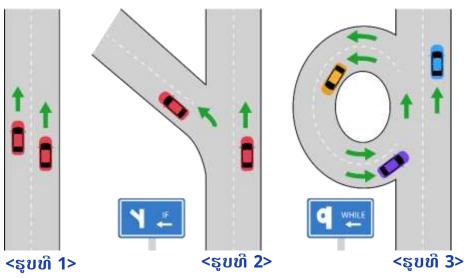
Key concept

- 1. ຄຳສັ່ງ if
- 1.1. ຄຳ້ສັ່ງເງື່ອນໄຂທີ່ມີທິດທາງການທຳງານຕ່າງກັນໂດຍຂຶ້ນກັບເງື່ອນໄຂ.
- ໂປຣແກຣມພາສາ Python ຈະມີລັກສະນະໂຄງສ້າງດຳເນີນການຕາມລຳດັບ, ໂດຍຄຳສັ່ງທີ່ຂຽນກ່ອນ
 ຈະຖືກດຳເນີນການກ່ອນ.
- ໂຄງສ້າງການດຳເນີນການຕາມລຳດັບເປັນໂຄງສ້າງທົ່ວໄປທີ່ພາສາໂປຣແກຣມສ່ວນໃຫຍ່ ແລະ ພາສາ Python ໃຊ້.
- ເຖິງແນວໃດກໍຕາມ ໃນພາສາໂປຣແກຣມ, ການເຮັດວຽກຕາມລຳດັບບໍ່ໄດ້ໃຊ້ສະເໝີໄປ.
- ຄຳສັ່ງບາງຢ່າງອາດຈະດຳເນີນການໄດ້ເມື່ອກົງກັບເງື່ອນໄຂບາງປະການເທົ່ານັ້ນ ຫຼື ການເຮັດວຽກຕ່າງໆ
 ອາດຈະຂຶ້ນກັບເງື່ອນໄຂ.
- ໃນບົດນີ້ຈະໄດ້ຮຽນຮູ້ປະໂຫຍກຄຳສັ່ງແບບມີເງື່ອນໄຂໃນການແກ້ໄຂບັນຫາທີ່ມີຄວາມສັບຊ້ອນຫຼາຍ
 ຂຶ້ນ.

- 1. ຄຳສັ່ງ if
- 1.2. ໂຄງສ້າງພື້ນຖານຂອງຄຳສັ່ງຄວບຄຸມການເຮັດວຽກໃນໂປຣແກຣມ
- ປະກອບມີ 3 ໂຄງສ້າງຫຼັກຄຶ:
 - > ໂຄງສ້າງດຳເນີນການຕາມລຳດັບ: ເປັນໂຄງສ້າງທີ່ຄຳສັ່ງຖືກດຳເນີນການຕາມລຳດັບ.
 - > ໂຄງສ້າງເລືອກດຳເນີນການ: ເປັນໂຄງສ້າງທີ່ເລືອກ ແລະ ດຳເນີນການຄຳສັ່ງຂໍ້ໃດຂໍ້ໜຶ່ງຈາກຫຼາຍຄຳສັ່ງ.
 - > ໂຄງສ້າງດຳເນີນການແບບວົນຊ້ຳ: ເປັນໂຄງສ້າງທີ່ຄຳສັ່ງດຽວກັນຖືກດຳເນີນການວົນຮອບກະທຳຊ້ຳ.
- ແຜນຜັງຂອງສາມໂຄງສ້າງຂ້າງເທິງ ສະແດງດັ່ງຮູບລຸ່ມນີ້:



- 1. ຄຳສັ່ງ if
- 1.2. ໂຄງສ້າງພື້ນຖານຂອງຄຳສັ່ງຄວບຄຸມການເຮັດວຽກໃນໂປຣແກຣມ
- ໂຄງສ້າງຄຳສັ່ງຄວບຄຸມການເຮັດວຽກຂອງໂປຣແກຣມມີລັກສະນະຄ້າຍຄືກັບໂຄງສ້າງເລໂກ້. ງານເລໂກ້ເກືອບທັງ ໜິດຖືກສ້າງຂຶ້ນໂດຍໃຊ້ works ພຽງບໍ່ເທົ່າໃດເທົ່ານັ້ນ.
- ໂປຣແກຣມກໍເຊັ່ນດຽວກັນ, ເຖິງວ່າໂປຣແກຣມຈະມີຄວາມສັບຊ້ອນຫຼາຍປານໃດກໍຕາມ ກໍສາມາດສ້າງໄດ້ດ້ວຍ
 works ພື້ນຖານພຽງສາມ works ເທົ່ານັ້ນ.
- > ຕາມລຳດັບ ສາມາດພິຈາລະນາເສັ້ນທາງທີ່ລົດຈະໄປພຽງທາງຊື່ ເຊິ່ງສະແດງດັ່ງຮູບທີ 1.
- ຮຸບທີ 2 ເປັນທາງແຍກທີ່ລົດຈະເລືອກໄປທາງໃດທາງໜຶ່ງໃນສອງທາງ
- ການວົນຊ້ຳ ສະແດງດັ່ງຮູບທີ 3, ອາດຈະເວົ້າ ໄດ້ວ່າ ເປັນວົງວຽນທີ່ລົດເຄື່ອນທີ່ວົນໄປ. ໂຄງ ສ້າງແບບນີ້ສາມາດວົນກະທຳຊ້ຳຄຳສັ່ງເກົ່າໄດ້ ຫຼາຍຄັ້ງ.



- 1. ຄຳສັ່ງ if
- 1.3. ໂຄ້ງສ້າງຄຳສັ່ງດຳເນີນການຕາມລຳດັບ
- ໂຄງສ້າງຄຳສັ່ງດຳເນີນການຕາມລຳດັບ: ແມ່ນໂຄງສ້າງທີ່ຄຳສັ່ງໃດຂຽນກ່ອນຈະຖືກປະມວນຜືນກ່ອນ. ໃນແຕ່ລະຄັ້ງ num =
 num + 100 ຖືກດຳເນີນການ, ຕົວປ່ຽນ num ຈະເພີ່ມຂຶ້ນເທື່ອລະ 100.

```
      1
      num = 100

      2
      print('num = ', num)
      # 100 ຈະຖືກພິມສະແດງ

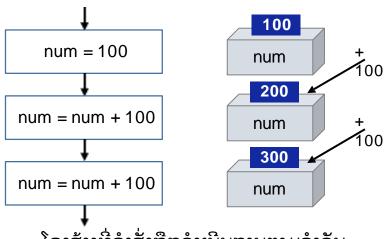
      3
      num = num + 100

      4
      print('num = ', num)
      # 100 ຖືກເພີ່ມເຂົ້າໃນຕົວປ່ຽນ num ແລະ ສະແດງຄ່າເປັນ 200

      5
      num = num + 100

      6
      print('num = ', num)
      # # 100 ຖືກເພີ່ມເຂົ້າໃນຕົວປ່ຽນ num ແລະ ສະແດງຄ່າເປັນ 300
```

```
num = 100
num = 200
num = 300
```



ໂຄງສ້າງທີ່ຄຳສັ່ງຖືກດຳເນີນການຕາມລຳດັບ

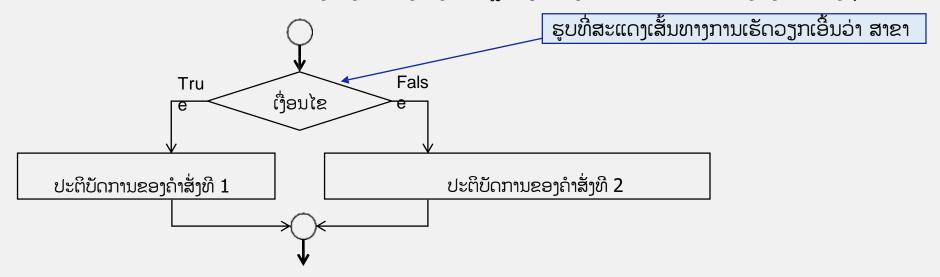
- 1. ຄຳສັ່ງ if
- 1.4. Sequence statement ແລະ Flow statement
- Flow statement ບໍ່ແມ່ນ Sequence statement
- ຄຳສັ່ງຄວບຄຸມ: ເປັນຄຳສັ່ງທີ່ຄວບຄຸມທິດທາງການເຮັດວຽກຕາມດຳລັບຂອງໂປຣແກຣມ.
- ຄຳສັ່ງຄວບຄຸມການດ□າເນີນການປະກອບມີ 3 ປະເພດຄື:
 - ຄຳສັ່ງແບບມີເງື່ອນໄຂ: ມີຄຳສັ່ງ if, if-else, if-elif-else
 - ຄຳສັ່ງວິນຊ in: ມີຄຳສັ່ງ for, while
 - ຄຳສັ່ງທີ່ປ່ຽນທິດທາງການວົນຊ in line in line

- 1. ຄຳສັ່ງ if
- 1.5. ເປັນຫຍັງຈິ່ງຕ້ອງມີການເລືອກໂຄງສ້າງ
- ເປັນຫຍັງຈິ່ງຕ້ອງມີການເລືອກໂຄງສ້າງ
 - ໃນບາງຄັ້ງຂັ້ນຕອນຂອງໂປຣແກຣມມີເສັ້ນທາງທີ່ຕ້ອງການດຳເນີນການຫຼາຍກວ່າໜຶ່ງເສັ້ນທາງ, ເຊິ່ງເຮົາ ຕ້ອງເລືອກໜຶ່ງເສັ້ນທາງ.

ກຳບໍ່ມີການເລືອກໂຄງສ້າງ ໂປຣແກຣມກຳຈະວົນຊ້ຳກະທຳອັນເກົ່າຕະຫຼອດ ແລະ ຈະເຮັດໃຫ້ໂປຣແກຣມບັນລຸຂໍ້ສະຫຼຸບທີ່ຕາຍຕົວຢູ່ສະເໝີ.

- 1. ຄຳສັ່ງ if
- 1.6. ທິດທາງການເຮັດວຽກຂອງຄຳສັ່ງເງື່ອນໄຂ

ເຮົາຈະສາມາດເລືອກປະມວນຜົນຄຳສັ່ງຢ່າງໃດຢ່າງໜຶ່ງຈາກຫຼາຍໆຄຳສັ່ງໄດ້ແນວໃດ? ສະແດງດັ່ງແຜນຜັງລຸ່ມນີ້:

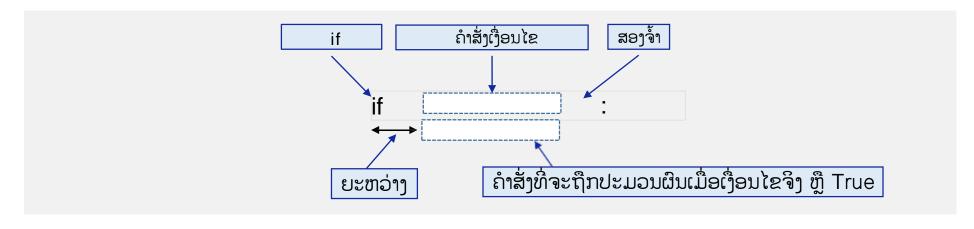


- ປະຕິບັດການຂອງຄຳສັ່ງທີ 1 ຫຼື ປະຕິບັດການຂອງຄຳສັ່ງທີ 2 ຈະດຳເນີນຕາມເງື່ອນໄຂສະເພາະ.
- ເງື່ອນໄຂ ຈະຖືກກຳນົດໄວ້ໃນ ຄຳສັ່ງເງື່ອນໄຂ, ເຊິ່ງຜົນຂອງການກວດສອບເງື່ອນໄຂຕ້ອງໃຫ້ຄ່າເປັນ True ຫຼື False.

Key concept

- 1. ຄຳສັ່ງ if
- 1.7. ຄຳສັ່ງ if ແລະ ເງື່ອນໄຂທີ່ກ່ຽວຂ້ອງທີ່ຈະຖືກປະມວນຜົນໃນທາງໃດທາງໜຶ່ງ
- ສຶກສາຄຳສັ່ງ if ແລະ ເງື່ອນໄຂທີ່ກ່ຽວຂ້ອງທີ່ຈະຖືກປະມວນຜົນໃນທາງໃດທາງໜຶ່ງ.
- ຂຽນໂປຣແກຣມໃຫ້ເຮັດວຽກຕາມສະຖານະການທີ່ສະແດງດັ່ງຕໍ່ໄປນີ້:
 - ສະຖານະການ 1: ສະແດງ 'Youth discount' ຖ້າລູກຄ້າ ອາຍຸຕໍ່າກວ່າ 20.
 - ສະຖານະການ $\mathbf{2}$: ສະແດງ 'Goal achieved' ຖ້າບາດກ້າວຍ່າງເກີນ $\mathbf{1,000}$ ບາດກ້າວ.
 - > ການປະຕິບັດໂປຣແກຣມນີ້ຈຳເປັນຈະຕ້ອງມີສຳນວນທີ່ກຳນຶດວ່າ ເງື່ອນໄຂນັ້ນເປັນໄປຕາມ ຫຼື ບໍ່.
 - ເຮົາເອີ້ນສຳນວນນີ້ວ່າ: ສຳນວນເງື່ອນໄຂ ແລະ ສຳນວນນີ້ຕ້ອງໃຫ້ຜົນຮັບໃນຮູບແບບຂໍ້ມູນຊະນິດຄ່າຄວາມຈິງ
 True ຫຼື False.
 - > ຕົວດຳເນີນການທີ່ກ່ຽວຂ້ອງຈາກບົດທີ່ຜ່ານມາຖືກນຳໃຊ້ເພື່ອປຽບທຽບສອງຕົວຖືກດ□າເນີນການ ແລະ ໃຫ້ຄ່າ ຜົນຮັບເປັນຄ່າ True ຫຼື False. ດັ່ງນັ້ນ, ມັນຈິ່ງສາມາດນຳໃຊ້ໄດ້ໃນສຳນວນເງື່ອນໄຂ.

- 1. ຄຳສັ່ງ if
- 1.8. syntax ຂອງຄຳສັ່ງ IF ສຶກສາ syntax ຂອງຄຳສັ່ງ IF



ສະຖານະການ 1: ສະແດງ 'Youth discount' ຖ້າອາຍຸຕ່ຳກວ່າ 20.

1

if age <20 :
 print('youth discount')</pre>

ສະຖານະການ 2: ສະແດງ 'Goal achieved' ຖ້າບາດກ້າວເກີນ 1.000.



if walk_count >= 1000 : print('goal achieved')

- (ສະຖານະການ1) ຢູ່ໃນປະໂຫຍກເງື່ອນໄຂທີ່ປະກິດກ່ອນເຄື່ອງ ໝາຍຈໍ້າສອງເມັດ (:), ຖ້າວ່າ ອາຍຸຕໍ່າກວ່າ 20 ປີ ຈະພິມຄໍາວ່າ 'youth discount'
- (ສະຖານະການ 2) ຢູ່ໃນປະໂຫຍກເງື່ອນໄຂທີ່ປະກິດກ່ອນ ເຄື່ອງໝາຍຈ້ຳສອງເມັດ (:), ໃນເວລາບາດກ້າວຫຼາຍກວ່າ ຫຼື ເທົ່າກັບ 1000 ຈະສະແດງຄຳວ່າ 'goal achieved' ອອກ ມາ.

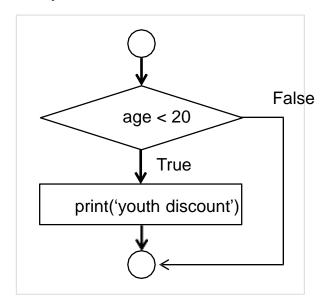
- 1. ຄຳສັ່ງ if
- 1.9. ຕົວຢ່າງການຂຽນໂປຣແກຣມຄຳສັ່ງ IF
 - ຂຽນໂປຣແກຣມທີ່ມີການນ□າໃຊ້ຄົ□າສັ່ງ IF

```
age = 18  # if age is 18

if age < 20:  # result of age < 20 is True

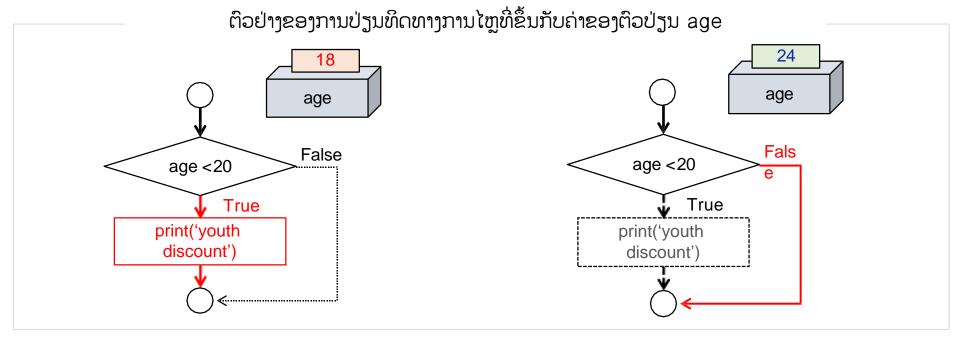
print ('youth discount')
```

youth discount



- ເມື່ອອາຍຸເທົ່າກັບ 18 ປີ, ເງື່ອນໄຂຈະໃຫ້ຄ່າເປັນ True ສະນັ້ນ, ໂປຣ ແກຣມຈະສະແດງຄຳວ່າ: 'youth discount' ອອກທາງໜ້າຈໍ.
- ເມື່ອອາຍຸຫຼາຍກວ່າ 20 ກໍຈະບໍ່ສະແດງຄ່າໃດໆອອກທາງໜ້າຈໍ.

- 1. ຄຳສັ່ງ if
- 1.9. ຕົວຢ່າງການຂຽນໂປຣແກຣມຄຳສັ່ງ IF



```
age = 18 # if age is 18

if age < 20: # result of age < 20 is True
print ('youth discount')

youth discount
```

```
age = 24 # if age is 24

if age < 20: # result of age < 20 is False print('youth discount')
```

- 1. ຄຳສັ່ງ if
- 1.10. Block ຄໍາສັ່ງ IF

🦚 ຄຳສັ່ງເງື່ອນໄຂ ແລະ Block ຄຳສັ່ງ

- ຮຽນຮູ້ຄວາມຈຳເປັນຂອງການຫຍໍ້ໜ້າ ແລະ Block ຄຳສັ່ງ
- ຕົວຢ່າງການຂຽນໂປຣແກຣມທີ່ມີຂໍ້ຜິດພາດ ສະແດງດັ່ງລຸ່ມນີ້:

```
1 age = 18
2
3 if age < 20:
4 print('youth discount')</pre>
```

File "<ipython-input-11-897b539e10ef>", line 4 print('youth discount')

IndentationError: expected an indented block

- ▶ ໃນສ່ວນໃດຂອງໂປຣແກຣມທີ່ສາມາດດຳເນີນການໄດ້ໃນເມື່ອເງື່ອນໄຂ True ເອີ້ນວ່າ Block.
- ຄຳສັ່ງໃດທີ່ນອນຢູ່ໃນເງື່ອນໄຂ ຈະຕ້ອງຫຍໍ້ໜ້າເຂົ້າ ຖ້າບໍ່ດັ່ງນັ້ນຈະເກີດຂໍ້ຜິດພາດດັ່ງຕົວຢ່າງຂ້າງເທິງນີ້.

Key concept

- 1. ຄຳສັ່ງ if
- 1.11. ການຍະຫວ່າງ

🎯 Focus) ພາສາ Python ເປັນພາສາໂປຣແກຣມທີ່ວ່າ ການຍະຫວ່າງເປັນສິ່ງທີ່ສຳຄັນ, ທັງນີ້ແມ່ນຂຶ້ນຢູ່ກັບການ ຍະວ່າງໃນໂປຣແກຣມດຽວກັນ, ແຕ່ໃຫ້ຜົນຮັບຕ່າງກັນ ສັງເກດຕົວຢ່າງຕໍ່ໄປນີ້:

 \succ ການຍະຫວ່າງແບບທີ f 1 : ເງື່ອນໄຂ IF $\ \ \, {
m age} < f 20$ ເປັນໄປຕາມເງື່ອນໄຂ

```
1 age = 18
2 if age < 20:
3    print('youth discount')
4 print('Welcome')</pre>
```

youth discount Welcome

>ການຍະຫວ່າງແບບທີ 2 : ເງື່ອນໄຂ IF age < 20 ບໍ່ເປັນໄປຕາມເງື່ອນໄຂ

```
1 age = 24
2 if age < 20:
3     print('youth discount')
4 print('Welcome')</pre>
```

Welcome

- 1. ຄຳສັ່ງ if
- 1.11. ການຍະຫວ່າງ
 - ການຍະຫວ່າງແບບທີ 3 : ເງື່ອນໄຂ IF age < 20 ເປັນໄປຕາມເງື່ອນໄຂ</p>

```
1 age = 18
2 if age < 20:
3     print('age', age) ('youth
    printwelcome')
5     print('youth discount')</pre>
```

Age 18 youth welcome youh discount

ຼີ ແຖວທີ 3-5

• ຍ້ອນວ່າ age = 18 ສະນັ້ນ, ເງື່ອນໄຂໃຫ້ຜົນຮັບເປັນ True ແລ້ວຄຳສັ່ງໃນແຖວທີ 3-5 ຈິ່ງຖືກປະຕິບັດ.

Key concept

- 1. ຄຳສັ່ງ if
- 1.11. ການຍະຫວ່າງ
 - ການຍະຫວ່າງແບບທີ 4 : ເງື່ອນໄຂ IF age < 20 ບໍ່ຖືກຕ້ອງຕາມເງື່ອນໄຂ</p>

```
1 age = 24
2 if age < 20:
3    print('age', age)
4    print('youth welcome')
5    print('youth discount')</pre>
```

No Output

```
🕮 แทอ 3-5
```

• ເນື່ອງຈາກອາຍຸ 24 ສະນັ້ນ, ເງື່ອນໄຂໃຫ້ຜົນຮັບເປັນ False ແລ້ວຄຳສັ່ງໃນແຖວທີ 3-5 ຈິ່ງບໍ່ຖືກປະຕິບັດ.

<u>-Ö</u>-

One More Step

- Block ຖືກເອີ້ນວ່າ: Block ລະຫັດຄຳສັ່ງ
- ໝາຍເຖິງ ຄຳສັ່ງບ່າງສ່ວນທີ່ສາມາດເຊື່ອມ ໂຍງກັບແຫຼ່ງທີ່ມາຂອງ Code.
- ພາສາ Python ຕ້ອງມີບ Code Block ທີ່ມີການຫຍໍ້ໜ້າເຂົ້າ ແລະ ຢູ່ຫຼັງຄຳສັ່ງ IF. ຖ້າຄຳສັ່ງ IF ຕາມດ້ວຍ ເຄື່ອງໝາຍຈໍ້າສອງເມັດ (:). else, elif, for, while, def, class, ອື່ນໆກໍ່ຕ້ອງມີການຍະຫວ່າງດ້ວຍ.
- ຖ້າຈ້ຳນວນຂອງການຍະຫົວ່າງບໍ່ຢູ່ພາຍໃນບ Block ຄຳສັ່ງດຽວກັນ ດັ່ງຄຳສັ່ງໃນແຖວທີ 5 ຈະເກີດຂໍ້ຜິດພາດ.
- ຮູບແບບຂອງຄຳສັ່ງພາສາ Python, ຊ່ອງຫວ່າງ 4 ຊ່ອງແມ່ນຂະໜາດຂອງການຍະຫວ່າງ.

🐞 ການຍະຫວ່າງທີ່ຜິດພາດ

```
1 age = 18
2 if age < 20:
3     print('age', age)
4     print ('youth welcome')
5     print ('youth discount')

File "<tokenize>", line 4
     print('youth welcome')

IndentationError: unindent does not match any outer indentation level
```

<u>-Ö</u>.

One More Step

- ຫຼັກການຂອງການຍະຫວ່າງ
- ແນວທາງ PEP-8 ຂອງພາສາ Python ແນະນຳໃຫ້ຂຽນໃນແຖວທາງຫຼັງເຄື່ອງໝາຍ :
- ສໍາລັບ IDE ແມ່ນຈະຍະຫວ່າງໃຫ້ແບບອັດຕະ ໂນມັດ
- ນອກຈາກນີ້, ໃຫ້ໃຊ້ປຸ່ມຍະຫວ່າງແທນການໃຊ້ປຸ່ມແທັບ. Block ສາມາດຂຽນໄດ້ຫຼາຍໆແຖວ.
- ສຳລັບຂຽນຄຳສັ່ງແບບຫຼາຍແຖວ ການຍະຫວ່າງຕ້ອງເທົ່າກັນທົງໝົດ.

```
1 age = 18
2 if age < 20: print('youth discount') # Bad coding</pre>
```

youth discount

```
1 age = 18
2 if age < 20:
3 print('youth discount') # Good coding</pre>
```

youth discount

- 2. ການໃຊ້ເງື່ອນໄຂຕົວເລກ ແລະ ຂໍ້ຄວາມກັບຄຳສັ່ງກວດສອບເງື່ອນໄຂ.
- 2.1. ໂປຣແກຣມເພື່ອກວດເບິ່ງວ່າແມ່ນຕິວເລກທີ່ໄດ້ຈາກຜືນຄູນຂອງເລກ 3 ຫຼື ບໍ່
- ຂຽນໂປຣແກຣມທີ່ມີການຮັບຄ່າຈາກຜູ້ໃຊ້ ແລະ ທ□າການກວດເບິ່ງວ່າ ຕົວເລກທີ່ຮັບມາແມ່ນຜືນຄຸນຂອງເລກ 3 ຫຼື ບໍ່

```
number = int(input('Enter an integer: ')) # ປ່ຽນຂໍ້ມູນທີ່ຮັບເຂົ້າມາໃຫ້ເປັນເລກຈຳນວນ
if (number % 3) == 0: ຖ້ວນ
print(number, 'is a multiple of 3.')
```

Enter an integer: 6 6 is a multiple of 3.

🖫 ແຖວທີ 2

- ຖ້າຕົວເລກທີ່ຮັບເຂົ້າມາຫານຂາດໃຫ້ 3 ແລະ ເສດ 0 ຄ□າສັ່ງໃນແຖວທີ 3 ຈະຖືກປະມວນຜົນ.
- ດ້ວຍວິທີນີ້, ຈິ່ງເຮັດໃຫ້ລະບຸໄດ້ວ່າ ຕົວເລກດັ່ງກ່າວແມ່ນຜົນຄູນຂອງເລກ 3 ຫຼື ບໍ່.

Key concept

- 2. ການໃຊ້ເງື່ອນໄຂຕົວເລກ ແລະ ຂໍ້ຄວາມກັບຄຳສັ່ງກວດສອບເງື່ອນໄຂ.
- 2.2. ໂປຣແກຣມເພື່ອກວດເບິ່ງວ່າແມ່ນຕິວເລກທີ່ໄດ້ຈາກຜົນຄູນຂອງເລກ 3 ແລະ 5 ຫຼື ບໍ່
- ຂຽນໂປຣແກຣມທີ່ມີການຮັບຄ່າຈາກຜູ້ໃຊ້ ແລະ ທ□າການກວດເບິ່ງວ່າ ຕົວເລກທີ່ຮັບມາແມ່ນຜົນຄູນຂອງເລກ 3 ແລະ 5 ຫຼື ບໍ່

```
number = int(input('Enter an integer: '))
if (number % 3) == 0 and (number % 5) == 0:
    print(number, 'is a multiple of 3 and 5.')
```

Enter an integer: 15 15 is a multiple of 3 and 5.

🏬 ແຖວທີ 2, 5

- ຖ້າວ່າຜົນຂອງການຫານໃຫ້ 3 ເສດ 0 ແລະ ຜົນຂອງການຫານໃຫ້ 5 ເສດ 0, ຄຳສັ່ງໃນແຖວທີ 3 ຈະຖືກ ປະມວນຜົນ.
- ດ້ວຍວິທີນີ້, ຈິ່ງເຮັດໃຫ້ລະບຸໄດ້ວ່າ ຕົວເລກດັ່ງກ່າວແມ່ນຜົນຄູນຂອງເລກ 3 ແລະ 5 ຫຼື ບໍ່.

- 2. ການໃຊ້ເງື່ອນໄຂຕິວເລກ ແລະ ຂໍ້ຄວາມກັບຄຳສັ່ງກວດສອບເງື່ອນໄຂ.
- 2.3. ຕົວດຳເນີນການ ==
 - ຂຽນໂປຣແກຣມເພື່ອກວດເບິ່ງວ່າ ຂໍ້ຄວາມທີ່ປ້ອນເຂົ້າມາມີຄ່າເທົ່າກັບຄ່າທີ່ເກັບໃນຕົວປ່ຽນ my_id ຫຼື ບໍ່, ເຊິ່ງ
 ເຮົາຈະໃຊ້ຕົວດຳເນີນການ ==

```
1 my_id = 'david'
2 
3 s = input('Enter your id: ')
4 if s == my_id:
5 print('ID matches.')
```

Enter your id: david ID matches.

ຼື ແຖວທີ 2, 5

• ຖ້າວ່າ ຂໍ້ຄວາມທີ່ຮັບເຂົ້າມີຄ່າເປັນ david, ຕົວປ່ຽນທັງສອງ s ແລະ my_id ມີຄ່າເທົ່າກັນ ແລ້ວຄຳສັ່ງໃນ ແຖວທີ 5 ກໍຈະຖືກປະມວນຜົນ. ຖ້າຮັບຂໍ້ຄວາມອື່ນເຂົ້າມາຄຳສັ່ງໃນແຖວທີ 5 ຈະບໍ່ຖືກປະມວນຜົນ.

- 3. ສະຖານະການຕ່າງໆຂອງເງື່ອນໄຂ
- 3.1. ການຈຳແນກເລກຄູ່
- ຕົວຢ່າງລຸ່ມນີ້ເປັນການຂຽນໂປຣແກຣມຈ□າແນກເລກວ່າແມ່ນເລກຄູ່ ຫຼື ບໍ່.

```
1  n = int(input("Enter an integer:"))
2  print("n =", n)
3  if n % 2 == 0:
    print(n, "is an even number.")
```

Enter an integer : 50 n = 50 50 is an even number.

🔢 ແຖວທີ 3

• ກວດເບິ່ງວ່າ ຖ້ານຳເອົາຕິວເລກທີ່ຮັບເຂົ້າມາໄປຫານໃຫ້ 2 ແລ້ວເສດ 0, ສືມມຸດ 50 ທີ່ຮັບເຂົ້າມາ ເຮັດໃຫ້ ເງື່ອນໄຂເປັນ True ດັ່ງນັ້ນ, ຄຳສັ່ງໃນແຖວທີ 4 ຈິ່ງຖືກປະມວນຜົນ.

Key concept

- ສະຖານະການຕ່າງໆຂອງເງື່ອນໄຂ
 ການກວດເບິ່ງສອງຂໍ້ຄວາມວ່າ ກົງກັນ ຫຼື ບໍ່
- ຕົວຢ່າງລຸ່ມນີ້ເປັນການຂຽນໂປຣແກຣມກວດເບິ່ງຂໍ້ຄວາມໃນສອງຕົວປ່ຽນວ່າກົງກັນ ຫຼື ບໍ່.

```
str1 = 'aaa'
2 | str2 = 'bbb'
  if str1 == str2:
       print('Two strings match.')
  if str1 != str2:
       print('Two strings are different.')
```

Two strings are different.

🔢 ແຖວທີ 3

• ຄຳສັ່ງໃນແຖວທີ 4 ຈະຖືກປະມວນຜົນ ຖ້າວ່າສອງຂໍ້ຄວາມຫາກກຶງກັນ, ເຊິ່ງຈະສະແດງຄຳວ່າ 'Two strings match'. ໃນກໍລະນີນີ້ str1 ແລະ str2 ມີຄ່າຕ່າງກັນ ໂປຣແກຣມຈະປະມວນຜົນໃນແຖວທີ 6 ເຊິ່ງຈະສະແດງຄຳວ່າ 'Two strings are different'.

Key concept

- **4.** ຄຳສັ່ງ pass
- **4.1.** ໜ້າທີ່ຂອງຄຳສັ່ງ pass
- ຄ□າສັ່ງ pass ຂອງພາສາ Python ໃຊ້ເພື່ອຍົກເວັ້ນ ຫຼື ຂ້າມການປະມວນຜົນຄຳສັ່ງພາຍໃນ Block.
- ສ່ວນໃຫຍ່ຈະຖືກເອິ້ນໃຊ້ໃນພາຍຫຼັງເມື່ອເວລາຕ້ອງການຂຽນ Code ເພາະຍັງບໍ່ໄດ້ກ□ານິດການດ□າເນີນການທີ່ ແນ່ນອນຂອງຄຳສັ່ງແບບເງື່ອນໄຂເທື່ອ.

```
1   num = 2
2   if num % 2 == 0:
        print('Even number')
4   if num % 3 == 0:
        pass
```

Even number

🌃 ແຖວທີ 5

• ໂດຍຈະປະໃຫ້ Block ຫວ່າງໄວ້ ເພື່ອໃຫ້ຟັງຊັນການເຮັດວຽກສຳເລັດສົມບຸນໃນພາຍຫຼັງ.

- 4. ຄຳສັ່ງ pass
- **4.1.** ໜ້າທີ່ຂອງຄຳສັ່ງ pass
- ຄຳສັ່ງ pass ສາມາດໃຊ້ໃນການຕັ້ງຊື່ ແລະ ໃຊ້ຟັງຊັນຕ່າງໆໃນພາຍຫຼັງ, ເຊິ່ງພວກເຮົາຈະຮຽນຮູ້ກ່ຽວກັບຟັງຊັນນີ້.
- ຟັງຊັນ func ດັ່ງລຸ່ມນີ້ ເປັນພຽງການກຳນົດຊື່ ແລະ ຈະນຳໃຊ້ໃນພາຍຫຼັງ.

```
1 def func():
2 pass
3
4 func()
```

🔢 ແຖວທີ 4

• ຟັງຊັນ func ຍັງບໍ່ມີຄຳສັ່ງໃຫ້ດຳເນີນການໃດໆ ເພາະວ່າບໍ່ມີຄຳສັ່ງຢູ່ພາຍໃນ func().

- 4. ຄຳສັ່ງ pass
- **4.1.** ໜ້າທີ່ຂອງຄຳສັ່ງ pass

🦚 ถำสั่ງ pass มิขิดขาดสำถับ

```
def func():
    func()
```

File "<ipython-input-3-7c72fccf7a42>", line 4
func()

IndentationError: expected an indented block

- ▶ ເຖິງວ່າຈະບໍ່ມີຟັງຊັນດຳເນີນງານ ແຕ່ຄຳສັ່ງ pass ກໍເຮັດໜ້າທີ່ໃນການຈອງຕຳແໜ່ງ.
- ດັ່ງນັ້ນ, ຖ້າບໍ່ໃສ່ຄຳສັ່ງ pass ກໍຈະເກີດຂໍ້ຜິດພາດຂຶ້ນ ດັ່ງໃນຕົວຢ່າງຂ້າງເທິງນີ້.

<u>-Ö</u>.

One More Step

- ຄຳສັ່ງ pass ແລະ ຟັງຊັນ
- ການພົມ, ການຮັບຂໍ້ມູນ, ອື່ນໆ ທີ່ເຮົາໃຊ້ແມ່ນຊຸດຄຳສັ່ງທີ່ດຳເນີນການຕາມທີ່ກຳນຶດ.
- ຟັງຊັນນີ້ ຈະເຮັດວຽກສະເພາະເມື່ອໃດທີ່ຖືກເອີ້ນໃຊ້ງານ.
- ບາງຄັ້ງ ພວກເຮົາຕ້ອງສ້າງຟັງຊັນຂຶ້ນມາດ້ວຍຕົວເອງ. (ຈະໄດ້ຮຽນອີກໃນພາຍຫຼັງ)
- ເຮົາອາດຈະສ້າງ ແລະ ທົດສອບຟັງຊັນ ໂດຍບໍ່ຕ້ອງຂຽນຄຳສັ່ງພາຍໃນຟັງຊັນ ໃຫ້ໃຊ້ຄຳສັ່ງ pass ເພື່ອໃຫ້ຮູ້ວ່າຟັງຊັນຍັງ ບໍ່ໄດ້ມີຄຳສັ່ງ.

```
def func_name(x1, x2, ...):
  code1 code2
  ...
return n1[, n2,...]
```

```
ຮູບແບບທົ່ວໄປຂອງຟັງຊັນ
```

```
def func_name(x1, x2, ...):
    pass
```

ຟັງຊັນທີ່ຍັງບໍ່ມີຄຳສັ່ງໃນໂຄງສ້າງ

Unit 6 ຄຳສັ່ງເງື່ອນໄຂ (Conditional Statement-1)

Paper Conding

- ພະຍາຍາມທຳຄວາມເຂົ້າໃຈພື້ນຖານແນວຄວາມຄິດກ່ອນຈະກ້າວໄປຍັງຂັ້ນຕອນຖັດໄປ.
- ຖ້າບໍ່ເຂົ້າໃຈພື້ນຖານແນວຄວາມຄິດກໍ່ຈະພາໃຫ້ມີບັນຫາໃນການຮຽນສໍາລັບຫຼັກສຸດນີ້ ອາດຈະເຮັດໃຫ້
 ເຮົາບໍ່ຜ່ານຫຼັກສູດນີ້.
- ມັນອາດຈະຂ້ອນຂ້າງຍາກໃນຕອນນີ້, ແຕ່ຜົນສຳເລັດຂອງຫຼັກສູດນີ້ພວກເຮົາຂໍແນະນ□າໃຫ້ທ່ານ
 ເຂົ້າໃຈຢ່າງລະອຽດກ່ຽວກັບແນວຄວາມຄິດ ແລ້ວຈິງກ້າວໄປຂັ້ນຕອນຕໍ່ໄປ.

Q1. ຖ້າວ່າ ຜູ້ໃຊ້ເກມ game_score ມີຄະແນນເກີນ 1000 ຄະແນນ, ໃຫ້ສະແດງຂໍ້ຄວາມວ່າ 'Your

ຜືນໄດ້ຮັບໂປຣແກຣມ	Enter game score : 1500 game_score = 1500 You are a master.
	or Enter game score : 100 game_score = 100
ใຊ้เอลา	5 ນາທີ

ຂຽນ Code ທັງໝົດ ແລະ ຜົນໄດ້ຮັບໄວ້ໃນປຶ້ມບັນທຶກ

ຈຶ່ງຂຽນໂປຣແກຣມຮັບຄ່າຕົວເລກຈຳນວນຖ້ວນຂອງ x ໃນຊ່ວງ -100 ຫາ 100, 1) ໃຫ້ສະແດງຄ່າ x ອອກທາງ \mathbb{Q}_2 . ໜ້າຈໍ 2) ໃຫ້ສະແດງຄຳວ່າ '.... Is natural number', ຖ້າ x ແມ່ນເລກຈຳນວນຖ້ວນໃຫຍ່ກວ່າ 0. ຖ້າບໍ່ດັ່ງນັ້ນ, ໃຫ້ສະແດງ x = -10.

ຜືນໄດ້ຮັບໂປຣແກຣມ	Enter integer: x = 50 50 is a natural number.
	or
	Enter integer: x = -10
ใຊ้เอลา	5 ນາທີ

🙋 ຂຽນCodeທັງໝົດ ແລະ ຜົນໄດ້ຮັບໄວ້ໃນປຶ້ມບັນທຶກ

Unit 6 ຄຳສັ່ງເງື່ອນໄຂ (Conditional Statement-1)

Let's Code

Let's Code
Unit 6

- 1. ຄຳສັ່ງ IF
- 1.1. ຄຳສັ້ງ if ຈະຖືກດຳເນີນການເມື່ອກຶງກັບເງື່ອນໄຂບາງປະການ

```
1 x = 100
2 if x > 1:
3 print("x is greater than 1.")
```

x is greater than 1.

- \succ ທຶດສອບການເຮັດວຽກຂອງຄຳສັ່ງ IF ດັ່ງຕົວຢ່າງຂ້າງເທິງ ດ້ວຍການກຳນຶດຄ່າ 100 ເກັບໄວ້ໃນຕົວປ່ຽນ x ແລະ ຂຽນຄຳສັ່ງ if, ໃນກໍລະນີທີ່ສຳນວນເງື່ອນໄຂ x > 1 ຈະໃຫ້ຄ່າ True
- >ຈາກນັ້ນ, ຈະສະແດງຄຳວ່າ 'x is greater than 1' ອອກທາງໜ້າຈໍ. ໂດຍຄຳສັ່ງນີ້ຈະມີການຂຽນຫຼັງ ເຄື່ອງໝາຍຈໍ້າສອງເມັດ : ແລະ ມີການຍະຫວ່າງພາຍໃນ Block.

- 1. ຄຳສັ່ງ IF
- **1.2.** ຕົວດຳເນີນການຄວາມສຳພັນ = , = =

🌌 🔁 ຕົວດຳເນີນການກຳນົດຄ່າ = ແລະ ຕົວດຳເນີນການຄວາມສຳພັນ == ແມ່ນມີຄວາມແຕກຕ່າງກັນ

ໃນສ່ວນນີ້ ແມ່ນໄດ້ເວົ້າຢູ່ຕະຫຼອດ ຍ້ອນວ່າມື ໃໝ່ຫັດຂຽນໂປຣແກຣມມັກຜິດພາດເລື້ອຍໆ. ຕົວດຳເນີນການ =
 ແມ່ນໃຊ້ສຳລັບກຳນິດຄ່າທີ່ຢູ່ເບື້ອງຂວາມື ໃຫ້ເກັບໄວ້ຢູ່ໃນຕົວປ່ຽນທີ່ຢູ່ເບື້ອງຊ້າຍ. ສ່ວນຕົວດຳເນີນການ ==
 ແມ່ນໃຊ້ສຳລັບປຽບທຽບຄ່າຂອງຕົວປ່ຽນທີ່ຢູ່ທັງເບື້ອງຊ້າຍ ແລະ ເບື້ອງຂວາວ່າ ມີຄ່າເທົ່າກັນ ຫຼື ບໍ່, ໂດຍຈະໃຫ້ຄ່າເປັນ True ຫຼື False.

Let's Code
Unit 6

1. ຄຳສັ່ງ IF

- 1.3. ການຍະຫວ່າງໃນພາສາ Python
- ການຍະຫວ່າງຖືເປັນອົງປະກອບທີ່ສຳຄັນຂອງການຂຽນໂປຣແກຣມ
- ພວກເຮົາຈະຂຽນໂປຣແກຮມສະແດງແນວໃດ ຖ້າວ່າ: ໄດ້ຮັບການຍອມຮັບ ແລະ ມີຄຸນສົມບັດທີ່ຈະໄດ້ຮັບທຶນ. ຖ້າມີ
 ຄະແນນເກີນ 90 ຂຶ້ນໄປ. ໃນກໍລະນີນີ້ຈະຂຽນໂປຣແກຣມຍະຫວ່າງເປັນກຸ່ມດັ່ງລຸ່ມນີ້:

ຖ້າຄະແນນເທົ່າ 90 ຫຼື ສູງກວ່າ, ສອງແຖວຄຳສັ່ງ print() ຈະຖືກເອີ້ນໃຊ້ ຫຼື ຖືກປະມວນຜົນ ຍ້ອນວ່າທັງສອງ
 ປະໂຫຍກມີການຍະຫວ່າງໃຫ້ຢູ່ໃນກຸ່ມດຽວກັນ.

```
if score >= :

90

print("You are accepted.")
print("You are qualified for a scholarship.")
```

1. ຄຳສັ່ງ IF

- 1.3. ການຍະຫວ່າງໃນພາສາ Python
- ການຍະຫວ່າງເປັນປັດໄຈທີ່ສ□າຄັນຫຼາຍໃນພາສາ Python.
- ປະໂຫຍກຄ□າສັ່ງທັງໝົດທີ່ຢູ່ໃນ Block ດຽວກັນຈະຖືກປະມວນຜົນພ້ອມກັນ.
- ປະໂຫຍກຄ : ສັ່ງ 4 ແຖວທີ່ຢູ່ໃນ Block ຈະມີການຍະຫວ່າງດ້ານໜ້າເມື່ອປຽບທຽບກັບປະໂຫຍກດ້ານເທິງ, ສັງເກດຕົວຢ່າງລຸ່ມ

```
score = int(input("Enter the score: "))

if score >= 90: # if the condition is true, block indent is all executed

----print("Congratulations.")

----print("You are accepted.")

----print("You are qualified for a scholarship.")
```

Enter the score: 95 Congratulations.

You are accepted.

You are qualified for a scholarship.

🌃 ແຖວທີ 2-5

• ແຖວທີ 3, 4 ແລະ 5 ຈະຖືກປະມວນຜົນ ຍ້ອນວ່າຄະແນນທີ່ຮັບມາແມ່ນ 95.

- 1. ຄຳສັ່ງ IF
- 1.3. ການຍະຫວ່າງໃນພາສາ Python
- ໃນພາສາ Python ຖ້າມີຈຳນວນການຍະຫວ່າງທາງດ້ານໜ້າຫາກຄືກັນ ຫຼື ເທົ່າກັນ ກຸ່ມປະໂຫຍກຄຳສັ່ງເຫຼົ່ານັ້ນກໍ ຈະຢູ່ໃນ Block ດຽວກັນ. ແຕ່ຖ້າຈຳນວນການຍະຫວ່າງຢູ່ທາງດ້ານໜ້າຂອງແຕ່ລະແຖວບໍ່ຄືກັນ ກໍຈະເກີດຂໍ້ ຜິດພາດເກີດຂຶ້ນ.

📱 ແຖວທີ 3-5

• ເກີດຂໍ້ຜິດພາດຂຶ້ນ ຍ້ອນວ່າຈ□ານວນການຍະຫວ່າງດ້ານໜ້າຂອງຄ□າສັ່ງພາຍໃນ Block ບໍ່ເທົ່າກັນ.

Let's Code
Unit 6

- 2. ການນໍາໃຊ້ String ແລະ ຟັງຊັນ Split()
- 2.1. ຟັງຊັນ split()
 - ຟັງຊັ້ນ split() ເພື່ອແຍກ object ທີ່ເປັນຂໍ້ຄວາມ, ລາຍການຂໍ້ຄວາມຈະຖືກສ້າງຂຶ້ນ ເຊິ່ງສະແດງດັ່ງຜົນຮັບຂອງໂປຣແກຣມລຸ່ມນີ້:

```
words = 'Python is beautiful'
word_list = words.split()
print(word_list)

['Python', 'is', 'beautiful']
```

- ການແຍກຈະໃຊ້ຍະຫວ່າງເປັນຄ່າເລີ່ມຕົ້ນ. ດັ່ງນັ້ນ, ແຕ່ລະຄ່າຈະໃຊ້ເຄື່ອງໝາຍຍະຫວ່າງເປັນຕົວຂັ້ນ.
- ໃນວິທີນີ້, ຕົວອັກຂະສອນທີ່ແບ່ງຂໍ້ຄວາມອອກເປັນໜຶ່ງ ຫຼື ຫຼາຍຂໍ້ຄວາມເອີ້ນວ່າ: ຕົວຂັ້ນ.

- 2. ການນໍາໃຊ້ String ແລະ ຟັງຊັນ Split()
- 2.2. ຟັງຊັນ input() ແລະ split()
- ຟັງຊັນ input() ໃຊ້ຮັບຂໍ້ມູນປະເພດຂໍ້ຄວາມເທົ່ານັ້ນ, ສ່ວນຟັງຊັນ split() ຈະຕັດຂໍ້ຄວາມທີ່ຮັບເຂົ້າມາດ້ວຍຍະຫວ່າງ. ເພື່ອ
 ແບ່ງດ້ວຍເຄື່ອງໝາຍ , ກໍຈະໃຊ້ຟັງຊັນ split(',').
- ເພື່ອຢາກດຳເນີນການສິ່ງນີ້ກັບຕົວເລກ, ຈຳເປັນຕ້ອງໄດ້ແປງຂໍ້ມູນໃຫ້ເປັນພະເພດຈຳນວນຖ້ວນ ຫຼື ຈຳນວນຈິງ ດ້ວຍການໃຊ້ຟັງ
 ຊັນ int ຫຼື float.

```
1 str1 = input('Enter three integers divided by space: ')
 2 print('Input: ', str1)
  3 print('Input separated by split() method: ', str1.split())
                                                              ແຍກດ້ວຍເຄື່ອງໝາຍຍະ
                                               10 20 30
Enter three integers separated by space:
                                                              ຫວ່າາ
Input: 10 20 30
Input separated by split() method: ['10', '20', '30']
  1 str1 = input('Enter three integers separated by .: ')
 2 print('Input: ', str1)
 3 print('Input separated by split() method: ', str1.split(','))
                                                        ແຍກດ້ວຍເຄື່ອງໝາຍຈຸດ
                                         10. 20, 30
Enter three integers separated by ,:
Input: 10,20,30
Input separated by split() method: ['10', '20', '30']
```

- ສ້າງລາຍຊື່ຂອງ String ໂດຍໃຊ້ String ທີ່ໄດ້ຮັບໂດຍຟັງຊັນ split().
- > ຫຼັງຈາກນັ້ນ, ອົງປະກອບໃນລາຍການສາມາດປ່ຽນເປັນປະເພດຕົວເລກຈຳນວນຖ້ວນ.

- 2. ການນໍາໃຊ້ String ແລະ ຟັງຊັນ Split()
- 2.2. ວິທີປະມວນຜົນຄ່າທີ່ຮັບເຂົ້າມາດ້ວຍຟັງຊັນ split() ແລະ int()
- ຈຳນວນຕີວເລກໃນລາຍການຂໍ້ຄວາມຈະຖືກແຍກດໍ່ວຍຟັ່ງຊັ້ນ split(), ຈຳກນັ້ນການດຳເນີນການທາງຕົວເລກຈະ ເປັນໄປໄດ້ໃນເມື່ອປ່ຽນໃຫ້ເປັນຂໍ້ມູນຕີວເລກຈຳນວນຖ້ວນເທົ່ານັ້ນດ້ວຍຟັງຊັນ int()

```
num1, num2, num3 = input('Enter three integers separated by ,: ').split(',')
num1, num2, num3 = int(num1), int(num2), int(num3)

Enter three integers separated by ,:

Enter three integers separated by ,:

1    num1, num2, num3 = input('Enter three integers separated by ,: ').split(',')
2    num1, num2, num3 = int(num1), int(num2), int(num3)
3    print(num1, num2, num3)
```

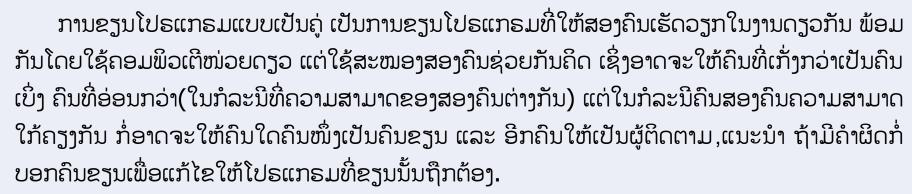
Enter three integers separated by ,: 5,6,7 5 6 7

Unit 6 ຄຳສັ່ງເງື່ອນໄຂ (Conditional Statement-1)

Pair Programming



ຝຶກການຂຽນໂປຣແກຣມແບບເປັນຄູ່



- ແນວທາງການຂຽນໂປຣແກຣມແບບນີ້ໃຊ້ໄດ້ຫຼາຍກໍລະນີ ເຊັ່ນ:
 - ✓ ຖ້າຕ້ອງຂຽນໂປຣແກຣມຍາກໆທີ່ຄິດຄົນດຽວບໍ່ອອກ ກໍ່ລອງຫາຄູ່ຂຽນໂປຣແກຣມທີ່ມີຄວາມສາມາດໃນ ລະດັບດຽວກັນ ຫຼື ໃຫ້ຄຽງກັນ ເພື່ອພັດປ່ຽນການຂຽນໂປຣແກຣມ ແລະ ທຳງານ.
 - ✓ ຕ້ອງການພັດທະນານັກຂຽນໂປຣແກຣມມືໃໝ່ ເພື່ອຮັບເຂົ້າທີມພັດທະນາ

<u>=</u> </>

ຝຶກການຂຽນໂປຣແກຣມແບບເປັນຄູ່

- 🗖 ຂໍ້ດີຂອງການຂຽນໂປຣແກຣມແບບເປັນຄູ່
- ເຮົາຈະໄດ້ພັດທະນາທັກສະການເຮັດວຽກເປັນທີມ, ຝຶກການເວົ້າ, ອະທິບາຍໃຫ້ໝູ່ເຂົ້າໃຈວ່າກຳລັງຄົ້ນຄິດ
 ເລື່ອງຫຍັງ ແລະ ກຳລັງເຮັດຫຍັງຢູ່.
- ຊ່ວຍລຸດຄວາມຜິດພາດໃນການຂຽນ Code ໂປຣແກຣມ ເພາະວ່າມີຄົນໜຶ່ງເປັນຜູ້ເບິ່ງ,ຕິດຕາມຄວາມ
 ຖືກຕ້ອງຂອງCodeທີ່ຂຽນ.
- ໄດ້ຮຽນຮູ້ເທັກນິກການຂຽນໂປຣແກຣມ ແລະ ການແກ້ໄຂປັນຫາຈາກຄູ່ ຫຼື ທີ່ມ ເຊິ່ງຈະຊ່ວຍໃຫ້ເຮົາຮຽນ
 ຮູ້ໄດ້ໄວຂຶ້ນ

Q1

ຈຶ່ງຂຽນໂປຣແກຣມທີ່ມີການຮັບຄ່າອາຍຸຈາກຜູ້ໃຊ້ ແລະ ໃຫ້ພິມຄຳວ່າ 'Adult' ຖ້າ ອາຍຸໃຫຍ່ກວ່າ ຫຼື ເທົ່າກັບ 20, ແຕ່ຖ້າອາຍຸຕ່ຳ ກວ່າ 20 ໃຫ້ພິມຄຳວ່າ ແລະ ໃຫຍ່ກວ່າ ຫຼື ເທົ່າກັບ 10 ແມ່ນໃຫ້ພິມຄຳວ່າ 'Youth'. ຖ້າອາຍຸຕ່ຳກວ່າ 10 ໃຫ້ສະແດງຄຳວ່າ 'Kid'.

ຜົນໄດ້ຮັບ

Enter age: 16
Youth

Enter age: 33
Adult
Enter age: 5
Kid

Q2. ຈຶ່ງຂຽນໂປຣແກຣມສະແດງຄຳວ່າ 'You can enter' ຖ້າຄ່າອາຍຸ ຫຼາຍກວ່າ ຫຼື ເທົ່າກັບ 20 ແລະ ມີລວງສູງເທົ່າ ກັບ ຫຼື ສູງກວ່າ 180 ຊັງຕີແມັດ.

ຜິນໄດ້ຮັບ

Enter age: 20

Enter height in cm: 180

You can enter.