ບົດທີ 3 ປະເພດຂໍ້ມູນ ແລະ ຕົວດຳເນີນການທາງດ້ານ ຄະນິດສາດພື້ນຖານ

- Learning objectives
- ຕົວດຳເນີນການທາງຄະນິດສາດພື້ນຖານເຊັ່ນ: ບວກ, ລົບ, ຄູນ, ຫານ ແລະ ຂຶ້ນກຳລັງ ໂດຍໃຊ້ພາສາ Python ແລະ ນຳໃຊ້ຟັງຊັນ ຂອງ Python ໃນການຫານເອົາສ່ວນຖ້ວນ ແລະ ການຫານເອົາສ່ວນເສດ.
- ເຂົ້າໃຈກ່ຽວກັບປະເພດຂໍ້ມູນຕ່າງໆ, ການປະມວນຜືນຂໍ້ມູນ ແລະ ສາມາດນຳໃຊ້ຕົວດຳເນີນການຄຳນວນ ຕົວເລກ ໂດຍໃຊ້ຄວາມສຳພັນລະຫວ່າງປະເພດຂໍ້ມູນ ແລະ ຕົວດຳເນີນການໄດ້.
- ຮຽນຮູ້ວິທີການຈັດການກັບຂໍ້ຄວາມ ແລະ ສາມາດພິມຄ່າຂໍ້ມູນຈາກ Code ໄດ້.
- ສາມາດຂຽນສົມຜົນໂດຍໃຊ້ລະດັບຄວາມສຳຄັນຂອງຕົວດຳເນີນການ.

- Learning overview
- ຮຽນຮູ້ຕົວດຳເນີນການທາງຄະນິດສາດພື້ນຖານເຊັ່ນ: ການບວກ, ລິບ, ຄຸນ, ຫານ, ຍົກກຳລັງ, ອື່ນໆ ໂດຍໃຊ້ປະເພດ ຂໍ້ມູນຕົວເລກ.
- ຮຽນຮູ້ປະເພດຂໍ້ມູນຂອງ Objects ແລະ ການເຮັດວຽກຂອງຕ໊ວດ □າເນີນການເຊີ່ງຂຶ້ນກັບປະເພດຂໍ້ມູນ.
- ສະແດງຜົນໂດຍໃຊ້ປະເພດຂໍ້ມູນທີ່ເປັນຂໍ້ຄວາມ.
- Concepts you will need to know from previous units
 - ການສະແດງຜືນເທິງໝ້າຈໍໂດຍໃຊ້ຄຳສັ່ງ print.
- ວິທີການເອີ້ນໃຊ້ Python ດ້ວຍ Jupyter Notebook. Python ເປັນ Python Shell ສໍາລັບການຂຽນ ໂປຣແກຣມແບບໂຕ້ຕອບ ແລະສາມາດຂຽນ Code ຕາມເງື່ອນໄຂ.

Keywords

Data Types

Numeric Data Types Arithmetic Operations

Strings

String Indexing

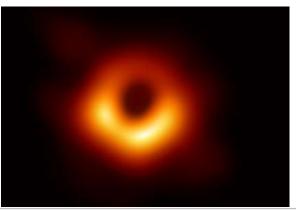
Unit 03. Basic of Numeric Data Types and Arithmetic Operation

Mission

1. Real world problem

1.1. Observing the Shadow of a Black Hole

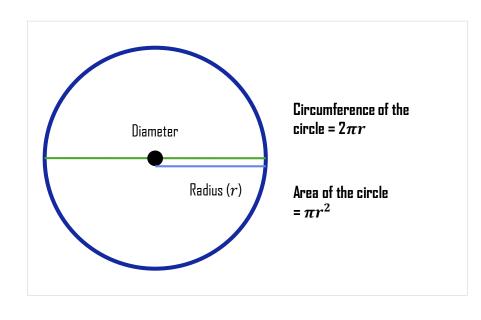




- ຮູບພາບດ້ານລຸ່ມຄື ແມ່ນຮູບຖ່າຍຂຸມດຳທຳອິດທີ່ ມະນຸດສັງເກດໄດ້
- ໂຄງການນີ້ເອີ້ນວ່າ EHT(Event Horizon Telescope) ແລະ ນັກວິທະ ຍາສາດຈາກ 32 ປະເທດທີ່ວໂລກໄດ້ເຂົ້າຮ່ວມໃນໂຄງການນີ້.
- ການເກດຂຸມດຳທີ່ຫ່າງໄກນັ້ນເປັນເລື່ອງຍາກຫຼາຍ.
- ເພື່ອແກ້ໄຂບັນຫານີ້, ສະມາຊິກຂອງໂຄງການ
 EHT ໄດ້ຕັດສິນໃຈໃຊ້ກ້ອງສ່ອງທາງໄກສັງເກດ
 ຂຸມດຳໃນຈັກະວານ, ໂດຍໃຊ້ກ້ອງສ່ອງທາງໄກ
 ໃນພື້ນທີ່ຕ່າງໆ ໃນເວລາດຽວກັນ, ແລະ ວິເຄາະ
 ຮູບພາບທີ່ສັງເກດໄດ້ເຫຼົ່ານັ້ນ.

1. Real world problem

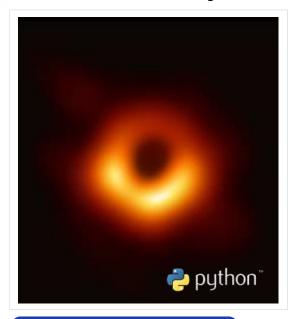
1.2. Circumference and Area Formula of the Circle



- ຈະພະຍາຍາມແກ້ໄຂບັນຫາທາງດ້ານຄະນິດສາດພື້ນຖານ, ຊຶ່ງເປັນ ໜຶ່ງໃນຫຼາຍໆວິທີທີ່ນັກວິສະວະກອນ ແລະ ນັກວິທະຍາສາດໃຊ້ໃນ ໂຄງການ EHT ເພື່ອແກ້ໄຂບັນຫາ.
- ຖ້າມີວົງວົນຕາມເສັ້ນສະແດງໃນຮູບ, ລວງຮອບ ແລະ ເນື້ອທີ່ຂອງ ວົງວົນ ຈະຖືກຄຳນວນໄດ້ ໂດຍໃຊ້ເຄື່ອງມືທີ່ເອີ້ນວ່າໂປຣແກຣມ.
- ໃນກໍລະນີນີ້, ສູດທາງດ້ານຊ້າຍສາມາດນຳມາໃຊ້ໄດ້.
- ຈາກສູດສະແດງວ່າລວງຮອບ ແລະ ເນື້ອທີ່ຂອງວົງມືນຂຶ້ນກັບລັດ ສະໜີຂອງວົງມືນ.
- ຜ່ານກິດຈະກຳນີ້, ຈະໄດ້ຮຽນຮູ້ການໃຊ້ຕົວປ່ຽນ ແລະ ວິທີການແກ້ ໄຂບັນຫາໂດຍໃຊ້ຕົວດຳເນີນການ.

2. Solution

2.1. Python contributed greatly to the shadow observation of black holes.



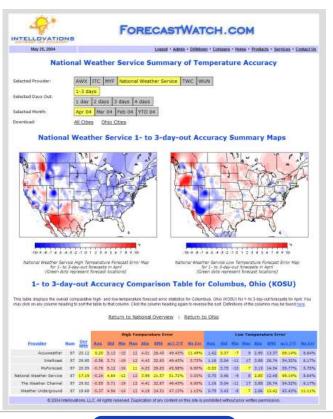
https://eventhorizontelescope.org/

- ວັນທີ 10 ເມສາ 2019, ມີການສັງເກດຂຸມດຳເທື່ອທຳອິດຂອງມະນຸດປະສິບຄວາມສຳເລັດ, ຂຸມດຳເອງບໍ່ສາມາດ ສັງເກດໄດ້ ແລະ "ເງົາ" ຖືກປະກອບເປັນຮູບພາບ.
- ນັກວິທະນາສາດໃຊ້ເຕັກໂນໂລຊີ ເພື່ອໃຫ້ໄດ້ຮູບພາບຂອງຫຼຸມດຳ, ໂດຍການລວບລວມສັນຍານຈາກກ້ອງ ທໍລະນີວິທະຍາທີ່ແຕກຕ່າງກັນທົ່ວໂລກ.
- ເວົ້າອີກຢ່າງໜຶ່ງ, ຖືວ່າໂລກເປັນກ້ອງທໍລະນີວິທະຍາຂະໜາດໃຫຍ່, ໂດຍໃຊ້ເກັບລວບລວມຂໍ້ມູນໄດ້ເຖິງ 5 Petabytes ແລະ ໃຊ້ເຄື່ອງມືຫຼາຍຢ່າງ, ໂດຍ<mark>ແພັກເກດ NumPy ຂອງ Python</mark> ມີບົດບາດສຳຄັນໃນເລື່ອງນີ້.



2. Solution

2.2. Python to solve the global climate crisis



- Python ໄດ້ກາຍເປັນເຄື່ອງມືທີ່ຈຳເປັນໃນຫຼາຍສາຂາ, ລວມ ເຖິງຊີວະວິທະຍາ, ຟີຊິກສາດ, ດາລາສາດ, ວິທະຍາສາດຄອມ ພີວເຕີ ແລະ ວິທະຍາສາດຂໍ້ມູນ.
- ເວັບໄຊ forecastwatch.com, ໃຫ້ບໍລິການກ່ຽວກັບການ ພະຍາ ກອນສະພາບອາກາດຂອງໂລກ, ໂດຍໃຊ້ Python ເພື່ອ ປະມວນຜົນຂໍ້ມູນທີ່ລວບລວມຈາກດາວທຽມ, ກວດສອບ ສະພາບອາກາດ ແລະ ພະຍາກອນສະພາບອາກາດ.

3. Mission

3.1. Print the circumference and area of the circle

- ໃນກິດຈະກຳນີ້, ຮັບຂໍ້ມູນລັດສະໝີຂອງວົງມືນເຂົ້າ.
- ສະແດງລວງຮອບ ແລະ ເນື້ອທີ່ຂອງວົງວົນທີ່ຄຳນວນເທິງໜ້າຈໍ.
- ຄຳນວນລວງຮອບ ແລະ ເນື້ອທີ່ຂອງວົງມົນໂດຍໃຊ້ຕົວດຳເນີນການທີ່ຈະຮຽນຮູ້ໃນບົດນີ້.
- ໃສ່ຕົວເລກຕ່າງໆເຊັ່ນ: ລັດສະໝີຂອງວົງມືນເທົ່າ 4, 5, 6, 10, 20, 33.

3. Mission

3.2. How printing the circumference and area of the circle works

```
PI = 3.14
2 radius = float(input('Enter the radius :'))
3 circum = PI * radius * radius
4 area = 2.0 * PI * radius

print('Radius of the circle', radius)
7 print('Area of the circle', circum)
8 print('Circumference of the circle', area)

Enter the radius : 4.0
Radius of the circle 4.0
Area of the circle 50.24
Circumference of the circle 25.12
```

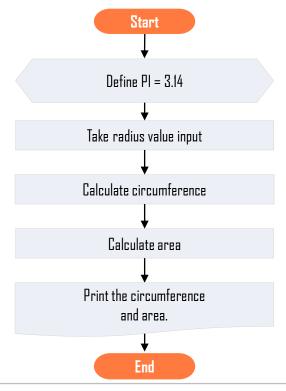
3. Mission

3.3. Programming plan

Pseudocode

- [1] ເລີ່ມຕົ້ນ
- [2] ກຳນຶດຄ່າຂອງ PI=3.14
- [3] ຮັບຂໍ້ມູນນຳເຂົ້າເຊິ່ງເປັນຄ່າລັດສະໝີຂອງວົງມືນ
- [4] ຄິດ ໄລ່ລວງຮອບ ໂດຍ ໃຊ້ຄ່າລັດສະໝີທີ່ຮັບເຂົ້າ ມາ
- [5] ຄິດ ໄລ່ເນື້ອທີ່ ໂດຍ ໃຊ້ຄ່າລັດສະໝີທີ່ຮັບເຂົ້າມາ
- [6] ສະແດງລວງຮອບ ແລະ ເນື້ອທີ່ເທິງໜ້າຈໍ
- [7] ສິ້ນສຸດ

Flow chart



3. Mission

3.4. Printing the circumference and area of the circle: final code

```
PI = 3.14
radius = float(input('Enter the radius :'))
circum = PI * radius * radius
area = 2.0 * PI * radius

print('Radius of the circle', radius)
print('Area of the circle', circum)
print('Circumference of the circle', area)
```

Unit 03. Basic of Numeric Data Types and Arithmetic Operation

Key concept

1. Numeric Data Types, Arithmetic and Division

1.1. Computation using numerical information

ການຂຽນໂປຣແກຣມ, ຕົວດຳເນີນການກ່ຽວກັບຂໍ້ມູນຕົວເລກ ແລະ ການໃຊ້ຂໍ້ມູນເຫຼົ່ານັ້ນເປັນສິ່ງທີ່ເກີດຂຶ້ນເລື້ອຍໆ ຕົວຢ່າງ ເຊັ່ນ: ຂໍ້ມູນລວງສູງ ແລະ ນ້ຳໜັກຂອງຄົນຖືກສະແດງເປັນຄ່າຕົວເລກເຊັ່ນ: 177 cm ແລະ 58 kg.

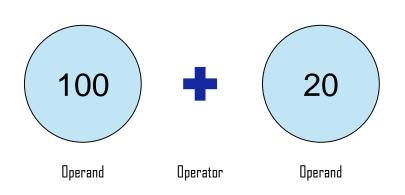


- Python ໃຊ້ຕົວເລກເພື່ອສະແດງຂໍ້ມູນປະເພດຕົວ ເລກ.
- Python ມີຕົວດຳເນີນການທາງດ້ານຄະນິດສາດຕ່າງໆ
- ເຂົ້າ ສຳລັບຂໍ້ມູນປະເພດຕົວເລີກຂອງຕົວເລກ 100 ແລະ 20 ສາມາດເຮັດໄດ້ໂດຍໃຊ້ 100+20 ໃນ Python

1. Numeric Data Types, Arithmetic and Division

1.1. Computation using numerical information

• ຮຽນຮູ້ກ່ຽວກັບຕົວດຳເນີນການທາງຄະນິດສາດ, ຂໍ້ມູນ ຫຼື ຕົວປ່ຽນທີ່ໄດ້ຮັບຈາກປະຕິບັດຕາມຕົວດຳເນີນການເອີ້ນວ່າ ຄະນິດສາດ ແລະ ຜົນທີ່ຜ່ານການຄິດໄລ່ດ້ວຍຕົວດຳເນີນການ ຕົວຢ່າງ: ການຄິດໄລ່ເຊັ່ນ 100+20





1. Numeric Data Types, Arithmetic and Division

1.2. Operators

I The table below explains the operator of Python and the operation of the operator.

ເຄື່ອງໝາຍ	ຄວາມໝາຍ	ອະທິບາຍ
+	ບວກ	ບວກຄ່າທາງຊ້າຍກັບຄ່າເບື້ອງຂວາເຂົ້າກັນ
-	ລິບ	ລິບຄ່າເບື້ອງຊ [້] າຍດ້ວຍຄ່າເບື້ອງຂວາ
*	ຄູນ	ຄຸນຄ່າເບື່ອງຊ [້] າຍດ້ວຍຄ່າເບື່ອງຂວາ
/	ຫານ	ການຫານໃນໄພທອນອອກຄ່າເປັນຈຳນວນຈິງຕະຫລອດ
//	ຫານເອົາຄ່າຖ້ວນ	ເປັນການຫານທີ່ເອົາຜົນການຫານເປັນຈຳນວນຖ້ວນ
%	ຫານເອົາຕິວເສດ	ເປັນການຫານທີ່ເອົາຈຳນວນເສດ
**	ຍົກກຳລັງ	ເປັນການຂື້ນກຳລັງຂອງຕົວເລກ. ໃນກໍລະນີຂອງ **0.5, ທຳການຖອນຮາກຂັ້ນສອງ.

1. Numeric Data Types, Arithmetic and Division

1.3. Example code using operator

Perform an operation using numerical data types. The operation may be performed using numerical data and operator +, -, *, /, //, %.

```
1 100 + 20 # Find addition of numerical value of 100 and 20
120
 1 100 * 20 # Find multiplication of numerical value of 100 and 20
2000
 1 100 - 20 # Find subtraction of numerical value of 100 and 20
80
 1 100 / 20 # Divide numerical value of 100 with 20
5.0
 1 100 // 20 # Divide numerical value of 100 with 20 and find quotient
5
 1 100 % 20 # Divide numerical value of 100 with 20 and find quotient
```

2. Object Data Types, Operator Priorities, and Data Type Conversion

- 2.1. Classification of data types ການແຍກປະເພດຂອງຂໍ້ມູນໃນ Python ເປັນສິ່ງທີ່ສຳຄັນຫຼາຍ.
- ເຖິງແມ່ນວ່າຕົວອັກສອນ ແລະ ຕົວເລກມີຮູບແບບທີ່ຄ້າຍຄືກັນ, ແຕ່ຄຸນສືມບັດຂອງພວກມັນແຕກຕ່າງກັນ.
- ຕົວເລກ 10 ເປັນຂໍ້ມູນປະເພດຕົວເລກ ແລະ ສາມາດນຳມາຄຳນວນໄດ້. ໃນທາງຕັກກະສາດ, "10" ເປັນຕົວອັກສອນເຖິງແມ່ນວ່າຮູບ
 - ແບບຈະຄ້າຍກັບເລກ[ີ] 10 ແຕ່ມີຄວາມແຕກຕ່າງທາງດ້ານຄວາມໝາຍ. ເຄື່ອງໝາຍທີ່ສາມາດໃຊ້ໄດ້ຂຶ້ນກັບປະເພດຂໍ້ມູນທີ່ໃຊ້. ໃນກໍລະນີນີ້ "10" ຕົວອັກສອນເມື່ອທຳການຫານ, Python ຈະແຈ້ງຂໍ້ຜິດພາດເນື່ອງຈາກ ຕົວອັກສອນບໍ່ສາມາດທຳການຫານໄດ້.



2. Object Data Types, Operator Priorities, and Data Type Conversion

2.1. Classification of data types

- ໃນ Python ຕົວດຳເນີນການມີການໃຊ້ທີ່ແຕກຕ່າງກັນຂຶ້ນຢູ່ກັບປະເພດຂອງຂໍ້ມູນ.
- ການຄູນ (*) ຂອງຕົວເລກ 100 ແລະ 2 ໃນ Python ຈະໄດ້ຜົນຮັບເປັນ 200.
- ການຄູນຂອງຂໍ້ຄວາມຈະສະແດງຂໍ້ຄວາມຊໍ້າຕາມຈຳນວນ. ນັ້ນໝາຍຄວາມວ່າການຄູນຂໍ້ຄວາມ '100', ດ້ວຍ 2 ຈະສະແດງຜົນຮັບເປັນ 100100, ແລະ ຄູນຂໍ້ຄວາມ 'Hello' ດ້ວຍ 2 ຈະສະແດງຜົນເປັນ 'HelloHello'.

```
1 100 * 2 # Find multiplication of numerical value 100 and 2
200

1 'Hello' * 2 # Find multiplication of string 'Hello' and 2
'HelloHello'

1 '100' * 2 # Find multiplication of string '100' and 2
'100100'
```

2. Object Data Types, Operator Priorities, and Data Type Conversion

2.2. Basic data types in Python

ປະເພດຂໍ້ມູນພື້ນຖາ້ນ ແລະ ຕົວຢ່າງການໃຊ້ວຽກໃນ Python ທີ່ນິຍົມມີດັ່ງນີ້.

Data Types	Example
Integers(int)	, -3, -2, -1, 0, 1, 2, 3,
Real Number(float)	3.14, 4.28, 0.01, 123.432
String(str)	'Hello World', '123', "Hi"
Bool(bool)	True, False

2. Object Data Types, Operator Priorities, and Data Type Conversion

2.2. Basic data types in Python

• ນີ້ຄືໂຄງສ້າງຂໍ້ມູນທີ່ສຳຄັນ ແລະ ຕົວປ່ຽນທີ່ສາມາດໃຊ້ໃນການຂຽນພາສາ Python

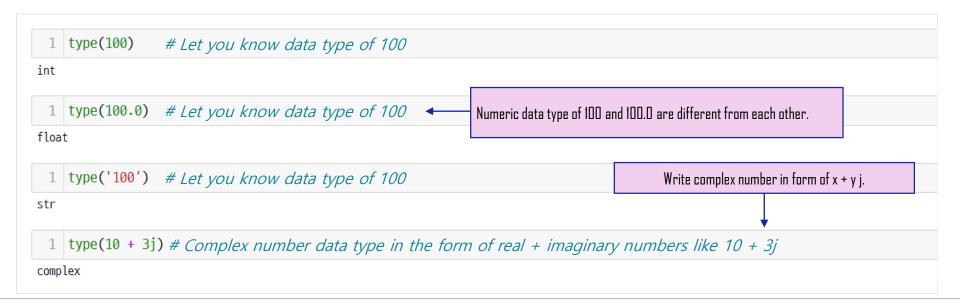
Number Boolean String List Tuple Dictionary ex: 'hello' EX: int EX: EX: [10, 20, 30]{ 'name' : 'David', '100' (10, 20, 30)True, False ex : 10, 200 'age': 23, 'height' : 56.5 } float ex : 3.14

complex ex : 4 + 3j

2. Object Data Types, Operator Priorities, and Data Type Conversion

2.3. How to distinguish data types

- ຄ່າທີ່ນຳມາໃຊ້ກັບຄ່າຕິວເລກ, ຟັງຊັນ ແລະ Class ເຊິ່ງເປັນສ່ວນປະກອບຂອງໂປຣແກຮມ Python ຄື Object
- ບົດບາດຂອງເຄື່ອງໝາຍແຕກຕ່າງກັນ ຂຶ້ນຢູ່ກັບປະເພດຂໍ້ມູນຂອງ Object ວ່າເປັນຈຳນວນຖ້ວນ ຫຼື ຂໍ້ຄວາມ
- ຟັງຊັນ type() ນຳມາໃຊ້ເພື່ອກວດສອບປະເພດ Object ຕໍ່ໄປນີ້ຄືຕົວຢ່າງທີ່ໃຊ້ຟັງຊັນ type() ເພື່ອກວດສອບປະເພດຂໍ້ມູນຂອງ Object()



2. Object Data Types, Operator Priorities, and Data Type Conversion

2.4. Precedence of operators

• ຖ້າເຄື່ອງໝາຍບວກ ແລະ ຄຸນ ປະກິດໃນສິມຜິນດຽວກັນ, ເຄື່ອງໝາຍຄູນຈະຖືກດຳເນີນການກ່ອນ ຕາມລຳດັບຄວາມສຳຄັນຂອງຕົວດຳ ເນີນການ.

```
1 | 10 + 20 * 30  # Multiplication is calculated first.

610

1 | (10 + 20) * 30  # Operator in parentheses is calculated first.

900
```

2. Object Data Types, Operator Priorities, and Data Type

2.5. Operator precedence chart

Let's look at Python operators and precedence. These are the contents to be learned in this lecture.

The higher the table, the higher the priority.

Operator	Explanation
()	ເຄື່ອງໝາຍວິງເລັບ
**	ເຄື່ອງໝາຍຂຶ້ນກຳລັງ
~,+,-	monadic operator
*,/,%,//	ເຄື່ອງໝາຍຄູນ, ຫານ, ຫານເອົາເສດ ແລະ ຫານເອົາຈຳນວນຖ້ວນ
+,-	ເຄື່ອງໝາຍບວກ, ລິບ
>> , <<	ເຄື່ອງໝາຍເລື່ອນທາງ ຂວາ, ຊ້າຍ
&	ເຄື່ອງໝາຍ ແລະ &
^,	ເຄື່ອງໝາຍ XOR , OR
<=,<,>,>=	ເຄື່ອງໝາຍປຽບທຽບ
== , !=	ເຄື່ອງໝາຍເທົ່າກັບ, ບໍ່ເທົ່າກັບ
=, %=, /=, //=, -=, +=, *=, **=	assignment operator, complex assignment operator
is, is not	identity operator ຕົວດຳເນີນການແບບສະແດງເອກະລັກ
in, not in	membership test operator ຕົວດຳເນີນການແບບເປັນສະມາຊິກ
not, or, and	logical operator ຕົວດຳເນີນການທາງຕັກະສາດ

2. Object Data Types, Operator Priorities, and Data Type Conversion

2.6. Data type conversion

- ຂໍ້ມູນໃນ python ສາມາດຄຳນວນໄດ້ໂດຍການປ່ຽນແປງຊະນິດຂໍ້ມູນເຊັ່ນ sting '10' ສາມາດແປງເປັນ ຊະນິດ int ໄດ້ໂດຍໃຊ້ int('10') ດັ່ງນັ້ນ int('10')/2 ຈະຖືກປະມວນຜົນໂດຍບໍ່ມີຂໍ້ຜິດພາດ.
- ເພື່ອປ່ຽນຕົວເລກ 10 ຈາກຂໍ້ມູນໃນ python ເປັນ string '10' ໃຊ້ str(10) ປ່ຽນເປັນ string ແລ້ວສາມາດຂຽນໄດ້ດ້ວຍການ ເພີ່ມ string ອີກ. ຕົວຢ່າງດັ່ງລຸ່ມນີ້.

```
1 int('10') / 2 # Convert string '10' into integer and divide by 2.
5.0

1 'I like number ' + str(10) # Convert number 10 into string and use + operator
'I like number 10'

1 a = None
2 type(a)
NoneType
```

One More Step

- None ສາມາດໃຊ້ໄດ້ເມື່ອຊະນິດຂອງ Object ໃນ Python ບໍ່ໄດ້ກຳນິດໄວ້.
- ຊະນິດຂອງຕົວປ່ຽນ a ທີ່ມີຄ່າເປັນ None ຄື NoneType.
- None ເບິ່ງຄືບໍ່ຈຳເປັນເນື່ອງຈາກບໍ່ມີຄ່າຂໍ້ມູນ, ເຖິງໃດກໍຕາມມັກຈະເປັນເລື່ອງຈຳເປັນເພາະເປັນວິທີດຽວໃນການລະບຸ ຄວາມສູນເສຍຂອງຄ່າ.

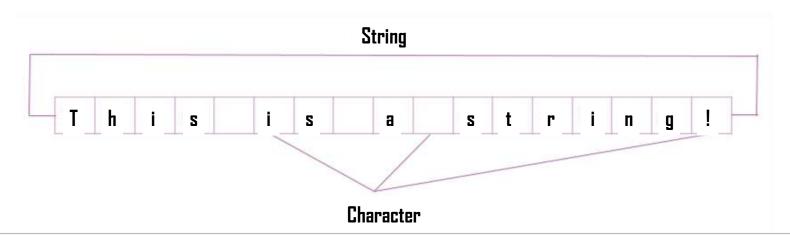
```
1 a = None
2 type(a)
```

NoneType

3. String Application

3.1. Take a look at string

- ໃນ Python, string ຫຼື srt ເໝາະສົມສໍາລັບການສະແດງຂໍ້ມູນໃນຊັອບແວ.
- ຟາຍ str ທັງໝົດສາມາດແປງເປັນຂໍ້ຄວາມໄດ້, ເຖິງໃດກໍຕາມເນື່ອງຈາກເປັນຂໍ້ຄວາມ, ຈຶ່ງບໍ່ສາມາດນຳໄປໃຊ້ຄຳນວນແບບເລກຄະນິດສາດໄດ້.
- Ex ຊື່ຜູ້ໃຊ້, ລະຫັດປະຈຳຕົວ (ID), ລະຫັດຜ່ານອື່ນໆ ອາດຈະຖືກສະແດງໃນຮູບແບບຂອງ string
- String ໃນ Python ເປັນຮູບແບບທີ່ນຳເອົາຕິວອັກສອນແຕ່ລະຕິວມາເຊື່ອມຕໍ່ກັນ.
- ການໃຊ້ຕົວດຳເນີນການບວກ ແລະ ຄຸນເປັນໄປໄດ້ສຳລັບ String ໃນ Python



3. String Application

3.2. String operator +, *
• ມາເບິ່ງການ + ແລະ * ທີ່ໃຊ້ໃນ String. ການ + ຂອງ String ຄືການເຊື່ອມຕໍ່ສອງ string ເຂົ້າດ້ວຍກັນ,ຕົວດຳເນີນການ * ຂອງ string ຕ້ອງເປັນຈຳນວນຖ້ວນ n ຄືຈະສະແດງ string ຊ້ຳ ເທົ່າກັບຈຳນວນ n.

```
1 'I' + ' Love' + ' Python!' # String can be added to strings.

1 'Python ' * 10 # Generates string of 'Python' repeated 10 times

1 '*' * 50 #Generates string of '*' repeated 50 times

1 str(100) * 10 #String '100' repeats 10 times
```

🎼 ການ * ຂອງ string ກັບຈຳນວນ ຈະຕ້ອງເປັນຈຳນວນຖ້ວນເທົ່ານັ້ນ ບໍ່ດັ່ງນັ້ນຈະເກີດຂໍ້ຜິດພາດ

```
TypeError

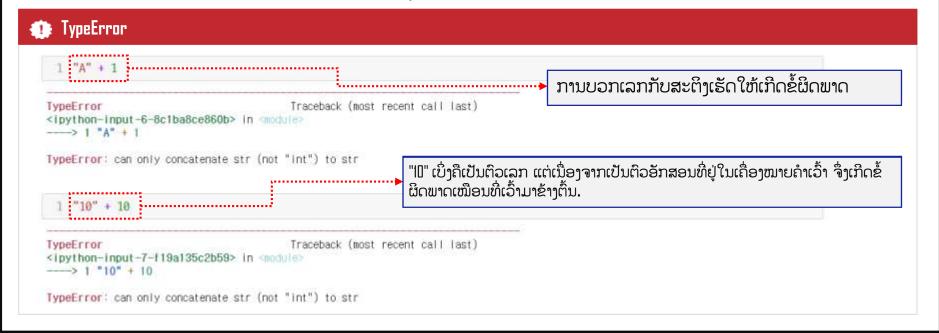
Traceback (most recent call last)

<ipython-input-5-5c37ed9b20aa> in <module>
---> 1 '*' * 3.1

TypeError: can't multiply sequence by non-int of type 'float'
```

One More Step

- ໃນ Python ໄດ້ແບ່ງຊະນິດຂໍ້ມູນອອກເປັນ: ຕົວອັກສອນ (letters), ລຳດັບ (number) ແລະ ອອບເຈັກ (objects)
- ໃນນັ້ນ String ແລະ ຕົວເລກສາມາດໃຊ້ຕົວດຳເນີນການເພີ່ມເຕີມໄດ້ ແຕ່ບັນຫາຈະເກີດຂຶ້ນ ເມື່ອມີການບວກຄ່າທີ່ມີ ຊະນິດຂໍ້ມູນຕ່າງກັນ.
- ດັ່ງນັ້ນຈຶ່ງຈຳເປັນຕ້ອງມີການແປງໃຫ້ເປັນຊະນິດຂໍ້ມູນດຽວກັນ ແລະ ສະແດງດັ່ງຕໍ່ໄປ



Unit 03. Basic of Numeric Data Types and Arithmetic Operation

Paper coding

- ພະຍາຍາມເຂົ້າໃຈແນວຄິດພື້ນຖານຢ່າງລະອຽດກ່ອນທີ່ຈະເລີ່ມຕົ້ນຂັ້ນຕອນຖັດໄປ.
- ການບໍ່ມີຄວາມເຂົ້າໃຈໃນແນວຄິດພື້ນຖານ ຈະເພີ່ມຄວາມຫຍຸ້ງຍາກຕໍ່ການຮຽນຮູ້ໃນລາຍວິຊານີ້, ເຊິ່ງອາດເຮັດໃຫ້ຫຼົ້ມເຫຼວໃນ ການຮຽນວິຊານີ້ໄດ້.
- ອາດຈະເປັນເລື່ອງຍາກໃນຂະນະນີ້, ແຕ່ສຳລັບການຮຽນລາຍວິຊານີ້ ເພື່ອໃຫ້ສຳເລັດຢ່າງສົມບູນ ຂໍແນະນຳໃຫ້ເຂົ້າໃຈແນວຄິດ ຢ່າງລະອຽດ ແລະ ກ້າວໄປສູ່ຂັ້ນຕອນຖັດໄປ.

Q1. ໃຊ້ຟັງຊັນ int ເພື່ອສະແດງຜົນເປັນຕິວເລກ 100 ໂດຍການເພີ່ມຕິວອັກສອນ "50" ແລະ ຕິວເລກ 50 ເຂົ້າດ້ວຍກັນ ແລະ ໃຊ້ຟັງຊັນ str ເພື່ອສະແດງຜົນຂອງການບວກອອກເປັນ "5050"

ເງື່ອນໄຂສຳ ລັບການດຳ	ແຖວທຳອິດສະແດງຜົນຮັບເປັນຕົວເລກ 100 ແລະ ແຖວທີ 2 ສະແດງຜົນຮັບເປັນຕົວອັກສອນ "5050"
ເນີນການ ເວລາ	5 มาที

Paper coding UNIT 03

Q2. ໃຫ້ໃຊ້ຕົວອັກສອນ '1' ແລະ '0' ສາມຕົວ ເພື່ອສ້າງເລກ 1000 ໂດຍໃຊ້ພຽງການບວກລະຫວ່າງ str ເທົ່ານັ້ນ ແລະ ໃຊ້ຟັງຊັນ int() ສາມາດໃຊ້ໄດ້ພຽງເທື່ອດຽວເທົ່ານັ້ນ

ເງື່ອນໄຂສຳ ລັບການດຳ	ສະແດງຜົນຮັບເປັນຕິວເລກ 1000
ເນີນ ເວລາ	5 ນາທີ

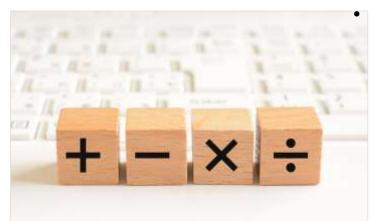
Unit 03. Basic of Numeric Data Types and Arithmetic Operation

Let's code

Let's code UNIT 03

1. Numeric Data Types, Arithmetic and Division

1.1. Arithmetic operation



ນຳສະເໜີຕົວດຳເນີນການທາງຄະນິດສາດໃນພາສາ Python ເຊິ່ງລວມເຖິງການບວກ, ການລົບ, ການຄຸນ, ການຫານ ແລະ ການຍົກກຳລັງສອງ. Let's code **UNIT** 03

1. Numeric Data Types, Arithmetic and Division

1.1. Arithmetic operation

Find out arithmetic calculation through coding.

```
1 1 + 5
 2 1 - 8
-7
 3 4 * 5
20
 4 5 // 3
```

Fig Line 1

• ການຄຳນວນການບວກ

Line 2

• ການຄຳນວນການລົບ

1. Numeric Data Types, Arithmetic and Division

1.1. Arithmetic operation

Line 3

• ການຄຳນວນການຄູນ

Line 4

• ການຄຳນວນການຫານເພື່ອເອົາສ່ວນຖ້ວນຂອງມັນ ຄືນຄ່າເປັນ 1 ຊຶ່ງເປັນສ່ວນຖ້ວນຂອງການຫານ 5 ດ້ວຍ 3

1. Numeric Data Types, Arithmetic and Division

1.2. Power, remainder

```
      1
      2 ** 10

      1024
      2
      9 % 5

      4
      4
```

Line 1 ~ 2

1: ຍົກກຳລັງ, 2^{10} ການຄຳນວນເພື່ອຄືນຄ່າ 1024.

2: ເອີ້ນວ່າ ໂມດູເລດ ຄືນຄ່າເສດຂອງການຫານ 9 ດ້ວຍ 5 ເຊິ່ງເທົ່າກັບ 4.

1. Numeric Data Types, Arithmetic and Division

1.3. Mixed operation

```
1 1 + 2 * 5
11
2 (1 + 2) * 5
15
```

ine 1 ~ 2

- 1: ຕົວດຳເນີນການທາງຕົວເລກຈະມີລຳດັບຄວາມສຳຄັນແຕກຕ່າງກັນຄື: ການຄຸນ ແລະ ການຫານຖືກດຳເນີນການກ່ອນ ການບວກ
- 2: ຖ້າມີວົງເລັບຈະຄຳນວນຄ່າໃນວົງເລັບກ່ອນ.

1. Numeric Data Types, Arithmetic and Division

1.4. Complex number operation

```
1 (3 + 4j) + (5 + 2j) # Addition of complex numbers

(8+6j)

2 (3 + 4j) - (5 + 2j) # Subtraction of complex numbers

(-2+2j)

3 (3 + 4j) * (5 + 2j) # Multiplication of complex numbers

(7+26j)

4 (3 + 4j) / (5 + 2j) # Division of complex numbers

(0.793103448275862+0.48275862068965514j)
```

Line 1 ~ 4

• ການບວກ, ການລົບ, ການຄູນ ແລະ ການຫານຂອງເລກຈຳນວນສິນ

2. Object Data Types, Operator Priorities, and Data Type Conversion

2.1. Identifying and converting data types

• ການກຳນົດ ແລະ ປ່ຽນຊະນິດຂອງຂໍ້ມູນເປັນເລື່ອງສຳຄັນຫຼາຍ.

int("10") # Convert string "10" to integer 10

(ຜູ້...) ຊະນິດຂໍ້ມູນມີຄວາມສຳຄັນເນື່ອງຈາກຕົວດຳເນີນການທີ່ຈະໃຊ້ມີຄວາມແຕກຕ່າງກັນ, ຂຶ້ນກັບຊະນິດຂອງຂໍ້ມູນ.

2. Object Data Types, Operator Priorities, and Data Type Conversion

2.1. Identifying and converting data types

• ຊອກຫາຊະນິດຂໍ້ມູນໂດຍໃຊ້ຟັງຊັນ type

```
1 type(10)
int

2 type("10")
str

3 type(10.5)
float
```

Line 1

• ເມື່ອມີຕົວເລກ (integer) ເປັນຂໍ້ມູນນຳເຂົ້າ, ຈະຖືກສະແດງຜົນເປັນ int.

E Line 2

• ເມື່ອສະຕິງເປັນຂໍ້ມູນນຳເຂົ້າ, ຈະຖືກສະແດງຜົນເປັນ str (string).

2. Object Data Types, Operator Priorities, and Data Type Conversion

2.1. Identifying and converting data types

• ຊອກຫາຊະນິດຂໍ້ມູນໂດຍໃຊ້ຟັງຊັນ type

```
1 type(10)
int

2 type("10")
str

3 type(10.5)
float
```

Line 3

• ເມື່ອຕົວເລກ (ຈຳນວນຈິງ) ທີ່ມີທົດສະນິຍົມເປັນຂໍ້ມູນນຳເຂົ້າ ຈະຖືກສະແດງຜົນເປັນ float (ທົດສະນິຍົມ).

One More Step

ໃນເຊວ ພິມ (100) ຈະສະແດງຜົນເປັນ int, ແຕ່ print(type(100)) ຈະສະແດງເປັນ <class 'int'> Class ເປັນຄຳ ສຳຄັນທີ່ກຳນົດວັດຖຸ, ເຊິ່ງຖືກນຳສະເໜີຢ່າງລະອຽດໃນ Unit 21.

```
1 type(10) # Cell output is int
int

1 print(type(10)) # Cell output is <class 'int'>
<class 'int'>
```

2. Object Data Types, Operator Priorities, and Data Type Conversion

2.2. How to convert a data type

```
1 int("50")
50

2 int("50") + 20
70

3 "10" + str(10)
'1010'

4 float("10.5")
10.5
```

line 1 ~ 2

- 1: ໃຊ້ຟັງຊັນ int ເພື່ອແທນຕົວອັກສອນດ້ວຍຕົວເລກ
- 2: ເນື່ອງຈາກສະຕິງ "50" ໄດ້ຖືກປ່ຽນຈຳນວນຖ້ວນໂດຍໃຊ້ int("50") ດັ່ງນັ້ນເຄື່ອງໝາຍ + 20 ຈະຖືກດຳເນີນການໂດຍສົມບູນ.

2. Object Data Types, Operator Priorities, and Data Type Conversion

2.2. How to convert a data type

```
1 int("50")

50

2 int("50") + 20

70

3 "10" + str(10)

'1010'

4 float("10.5")

10.5
```

```
Elline 3 ~ 4
```

- 3: ໃຊ້ຟັງຊັນ str ເພື່ອປ່ຽນຕົວເລກໃຫ້ເປັນຕົວອັກສອນ
- 4: ໃຊ້ຟັງຊັນ float ເພື່ອປ່ຽນຕົວອັກສອນໃຫ້ເປັນຈຳນວນຈິງ.

3. Strings and Operators

3.1. Addition and Multiplication Operators of String

• string ອາດໃຊ້ເຄື່ອງໝາຍ ການວກ ແລະ ການຄຸນ, ບໍ່ສາມາດໃຊ້ຕົວດຳເນີນການລົບ ແລະ ການຫານໄດ້.

```
1 "Hello" + "World"

'HelloWorld'

2 "text" + "repeat" * 10

'textrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatre

Line 1

• string ບໍ່ສາມາດລົບກັນໄດ້, ແຕ່ສາມາດບວກກັນໄດ້, ເມື່ອເອົາ 2 string ມາບວກກັນ, ຜົນທີ່ໄດ້ຮັບຄື ຈະນຳເອົາ 2 string ມາຕໍ່ໃສ່ ກັນ
```

3. Strings and Operators

3.1. Addition and Multiplication Operators of String

Find out the addition and multiplication operators of strings.

```
1 "Hello" + "World"

'HelloWorld'

2 "text" + "repeat" * 10

'textrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatrepeatr
```

```
Time 2
```

- ເມື່ອ string ຖືກຄູນດ້ວຍຈຳນວນຖ້ວນ, string ຈະສະແດງຊ້ຳຕາມຈຳນວນທີ່ກຳນົດເຊັ່ນ: 'repeat ຖືກເຮັດຊ້ຳ 10 ເທື່ອ'.
- ເຖິງຢ່າງໃດກໍຕາມ ບໍ່ສາມາດທຳການຫານ ແລະ ການລົບ ໄດ້.

3. Strings and Operators

3.2. Single and Double Quotation Marks

• ເມື່ອພິມ string ສະແດງຜົນຄືກັນບໍ່ວ່າຈະໃຊ້ເຄື່ອງໝາຍວົງຢືມດ່ຽວເຊັ່ນ: 'Hello Python!!' ຫຼື ເຄື່ອງໝາຍວົງຢືມຄູ່ເຊັ່ນ: "Helo Python!!".

```
1 print("Hello World!")
 2 print('Hello World!')
Hello World!
Hello World!
```

```
1 ~ 2 ≥ 1 ~ 2
    • ຂໍ້ຄວາມ "~~" ແລະ '~~' ສະແດງຜົນຄືກັນ
```

3. Strings and Operators

3.3. Using comma

• ເມື່ອຕ້ອງການເຊື່ອມຕໍ່ string ແລະ ຄ່າຕົວເລກຈະໃຊ້ເຄື່ອງໝາຍຈຸດ (,) ຂັ້ນກາງ.

```
print('I am', 23, 'years old!')
print('I', 'really ' * 3, 'love', 'Python!')

I am 23 years old!
I really really love Python!
```

```
ोहि Line 1 ~ 2
```

- string ແລະ ຄ່າຕິວເລກຖືກພິມໂດຍໃຊ້ເຄື່ອງໝາຍຈຸດ.
- ເຄື່ອງໝາຍຈຸດຈະເພີ່ມຊ່ອງຫວ່າງໂດຍອັດຕະ ໂນມັດແມື່ອສະແດງຜົນ

3. Strings and Operators

3.4. Escape sequence

• ລຳດັບການຫຼີກລ້ຽງ: \" ຫຼື \" ອາດໃຊ້ເພື່ອສະແດງຜົນເຄື່ອງໝາຍວົງຢືມໃຫຍ່ ຫຼື ນ້ອຍ

```
1 print("\"double quotation marks\"")
"double quotation marks"

1 print("\'single quotation marks\'")
'single quotation marks'
```

Line $1 \sim 2$

- string ຄວນຖືກລ້ອມຮອບດ້ວຍເຄື່ອງໝາຍວົງຢືມໃຫຍ່ ຫຼື ວົງຢືມນ້ອຍ
- ດັ່ງນັ້ນ ເປັນເລື່ອງຍາກທີ່ຈະບົ່ງບອກເຄື່ອງໝາຍວົງຢືມໃນກໍລະນີນີ້ ສາມາດໃຊ້ \" or \"
- ລຳດັບການຫຼີກລ້ຽງໝາຍເຖິງອັນສອນພິເສດ ທີ່ໃຊ້ຄວບຄຸມຄອມພີວເຕີ ແລະ ເນື່ອງຈາກມີການກຳນົດ string ຈຶ່ງຈຳອອກສະແດງຜົນ ໂດຍໃຊ້ຕົວອັກສອນຄວບຄຸມ \' ຫຼື \' ເພື່ອສະແດງຜົນ

3. Strings and Operators

3.4. Escape sequence

• ລຳດັບການຫຼີກລ້ຽງ: \n ໃຊ້ເພື່ອລົງແຖວໃໝ່

```
1 print("line\nbreak")
line
break
```



• <Inverse Slash+Character> ເພື່ອສະແດງອັກສອນພິເສດ ເຊັ່ນ: ການລົງແຖວ \n ເປັນຕົວອັກສອນລົງແຖວ ແລະ \t ເປັນຕົວ ອັກສອນແທັບ.

3. Strings and Operators

3.4. Escape sequence

• ລຳດັບການຫຼີກລ້ຽງ: \t ໃຊ້ເມື່ອຕ້ອງການປ້ອນຕົວອັກສອນເທັບ

```
1 print("Enter a \t tab in the middle of the sentence.")
Enter a tab in the middle of the sentence.
```



• ເມື່ອຕ້ອງການປ້ອນຕົວອັກສອນແທັບໃນປະໂຫຍກໃຫ້ປ່ອນ \t ທີ່ຕຳແໜ່ງທີ່ສວດຄ່ອງກັນ.

3. Strings and Operators

3.4. Escape sequence

• ລຳດັບການຫຼີກລ້ຽງ: \t ໃຊ້ເມື່ອຕ້ອງການປ້ອນຕົວອັກສອນເທັບ

```
print("\tEnter a tab at first!")
Enter a tab at first!
```

Line 1

• ເມື່ອຕ້ອງການປ້ອນຕົວອັກສອນແທັບທີ່ຕັ້ນປະໂຫນດ ໃຫ້ປ້ອນ \t ເປັນຕົວອັກສອນທຳອິດທີ່ສະແດງ.

3. Strings and Operators

3.5. String indexing

• ຕຳແໜ່ງຂໍ້ຄວາມ: ຕຳແໜ່ງອາດໃຊ້ເພື່ອຮັບສະເພາະບາງສ່ວນຂອງຂໍ້ຄວາມທີ່ຍາວ ໂດຍໃຊ້ວົງຂໍ [] ແລະ ຕົວເລກ

```
1 "hello"[0]
'h'
2 "hello"[2]
'l'
```

Line 1

- 1 ຕຳແໜ່ງຂໍ້ຄວາມເລີ່ມຕົ້ນຈາກ 0.
- 2 ນັ້ນຄື ເມື່ອດັດສະນີເປັນສູນ ຕົວອັກສອນ h ເຊິ່ງເປັນຕົວອັກສອນທຳອິດ "hello" ແມ່ນ ຕຳແໜ່ງ.

3. Strings and Operators

3.6. Negative string indexing

• ຕຳແໜ່ງຂໍ້ຄວາມ: ຕຳແໜ່ງອາດໃຊ້ເພື່ອຮັບສະເພາະບາງສ່ວນຂອງຂໍ້ຄວາມທີ່ຍາວ ໂດຍໃຊ້ວົງຂໍ [] ແລະ ຕົວເລກ

```
1 "hello"[-1]
'o'
```



• Python ຮອງຮັບຕຳແໜ່ງແບບລົບ ແລະ ເມື່ອຕຳແໜ່ງເປັນ -1ຈະເປັນຕຳແໜ່ງຕົວອັກສອນສຸດທ້າຍ.

3. Strings and Operators

3.6. Negative string indexing

• ຕຳແໜ່ງຂໍ້ຄວາມ: ຕຳແໜ່ງອາດໃຊ້ເພື່ອຮັບສະເພາະບາງສ່ວນຂອງຂໍ້ຄວາມທີ່ຍາວ ໂດຍຈະຮັບການອະທິບາຍຢ່າງລະອຽດໃນພາຍຫຼັງ

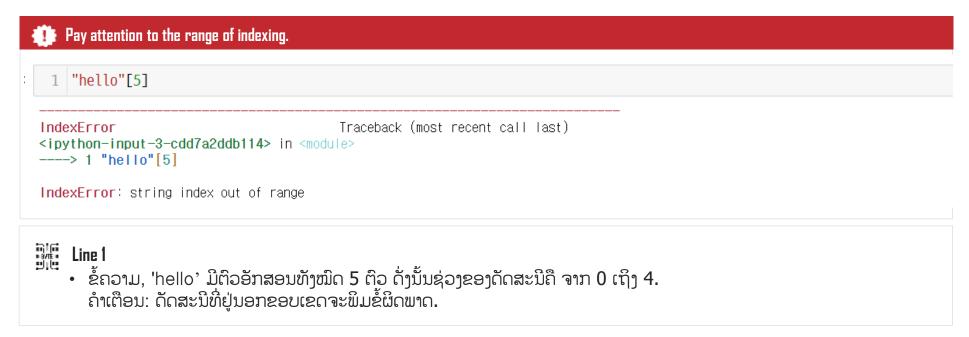
```
1 "hello"[-5]
'h'
```

Fig. Line 1

• Python ຮອງຮັບຕຳແໜ່ງແບບລົບ ແລະ ເມື່ອຕຳແໜ່ງເປັນ -5 ຈະປັນຕຳແໜ່ງຕົວອັກສອນທຳອິດ.

3. Strings and Operators

3.6. Negative string indexing



Unit 03. Basic of Numeric Data Types and Arithmetic Operation

Pair programming



ແບບເຝິກຫຼັບນາກຮົໄກ ເດອແນອກ



ແນວທາງ, ກົນໄກ ແລະ ແຜນສຸກເສີນ

ການກະກຽມໂປຣແກຣມຈັບຄູ່ກ່ຽວຂ້ອງກັບການກຳນົດແນວທາງ ແລະ ກົນໄກ ເພື່ອຊ່ວຍໃຫ້ນັກຮຽນຈັບຄູ່ໄດ້ຢ່າງຖືກຕ້ອງ ແລະ ເພື່ອໃຫ້ພວກເຂົາຈັບຄູ່ກັນ. ຕົວຢ່າງເຊັ່ນ ນັກຮຽນຄວນ ຜັດປ່ຽນກັນ "ບັງຄັບ mouse" ການກະກຽມທີ່ມີປະສິດຕິພາບ ຈຳເປັນຕ້ອງໃຫ້ມີແຜນສຸກເສີນ ໃນກໍລະນີທີ່ຄູ່ຮ່ວມງານຄົນໃດຄົນໜຶ່ງບໍ່ຢູ່ ຫຼື ຕັດສິນໃຈທີ່ຈະບໍ່ເຂົ້າຮ່ວມດ້ວຍເຫດຜົນໃດ ໜຶ່ງ ຫຼື ເຫດຜົນອື່ນ. ໃນກໍລະນີເຫຼົ່ານີ້, ສິ່ງສຳຄັນຄືຕ້ອງໄດ້ລະບຸໃຫ້ຊັດເຈນວ່າ ນັກຮຽນທີ່ມີການປະກອບສ່ວນຈະບໍ່ຖືກລົງໂທດຍ້ອນວ່າການຈັບຄູ່ບໍ່ໄດ້ຜົນດີ.

ການຈັບຄູ່ຄວາມສາມາດທີ່ຄ້າຍຄືກັນ, ບໍ່ຈຳເປັນຕ້ອງມີຄວາມສາມາດເທົ່າທຽມກັນ ໃນຖານະເປັນຄູ່ຮ່ວມງານກັນ

ການຂຽນໂປຣແກຣມຈັບຄູ່ ທີ່ມີປະສິດທິພາບເມື່ອນັກຮຽນທີ່ມີຄວາມສາມາດຄ້າຍຄືກັນ, ເຖິງວ່າບໍ່ຈຳເປັນຕ້ອງມີຄວາມເທົ່າທຽມກັນຖືກຈັບຄູ່ເປັນຄູ່ຮ່ວມງານກັນ. ການຈັບຄຸ່ນັກຮຽນທີ່ບໍ່ກົງ ກັນມັກຈະເຮັດໃຫ້ການມີສ່ວນຮ່ວມທີ່ບໍ່ສົມດຸນກັນ. ຄຸສອນຕ້ອງເນັ້ນໜັກວ່າການຂຽນໂປຣແກຣມຈັບຄຸ່ບໍ່ແມ່ນກິນລະຍຸດ "ແບ່ງປັນ ແລະ ເອົາຊະນະ", ແຕ່ເປັນຄວາມພະຍາຍາມຮ່ວມມືກັນ ທີ່ແທ້ຈິງ ໃນທຸກໆຄວາມພະຍາຍາມສຳລັບໂຄງການທັງໝົດ. ຄູຄວນຫຼີກເວັ້ນການຈັບຄູ່ນັກຮຽນທີ່ອ່ອນຫຼາຍກັບນັກຮຽນທີ່ເກັ່ງຫຼາຍ..

ສ້າງແຮງຈູງໃຈໃຫ້ກັບນັກຮຽນໂດຍການສະເໜີສິ່ງຈູງໃຈເພີ່ມເຕີມ

ການສະເໜີແຮງຈຸງໃຈເພີ່ມເຕີມສາມາດຊ່ວຍກະຕຸ້ນນັກຮຽນໃຫ້ຈັບຄູ່, ໂດຍສະເພາະກັບນັກຮຽນທີ່ຮຽນເກັ່ງ. ຄຸບາງຄົນພົບວ່າການກຳນົດໃຫ້ນັກຮຽນຈັບຄູ່ພຽງແຕ່ໜຶ່ງ ຫຼື ສອງວຽກນັ້ນຈະມີ ຜົນດີ.



ແບບເຝິກຫັດການຂຸງນໂປຣແກຣມ



ແນວທາງ, ກືນໄກ ແລະ ແຜນສຸກເສີນ

ການກະກຽມໂປຣແກຣມຈັບຄູ່ກ່ຽວຂ້ອງກັບການກຳນົດແນວທາງ ແລະ ກົນໄກ ເພື່ອຊ່ວຍໃຫ້ນັກຮຽນຈັບຄູ່ໄດ້ຢ່າງຖືກຕ້ອງ ແລະ ເພື່ອໃຫ້ພວກເຂົາຈັບຄູ່ກັນ. ຕົວຢ່າງເຊັ່ນ ນັກຮຽນຄວນ ຜັດປ່ຽນກັນ "ບັງຄັບ mouse" ການກະກຽມທີ່ມີປະສິດຕິພາບ ຈຳເປັນຕ້ອງໃຫ້ມີແຜນສຸກເສີນ ໃນກໍລະນີທີ່ຄູ່ຮ່ວມງານຄົນໃດຄົນໜຶ່ງບໍ່ຢູ່ ຫຼື ຕັດສິນໃຈທີ່ຈະບໍ່ເຂົ້າຮ່ວມດ້ວຍເຫດຜົນໃດ ໜຶ່ງ ຫຼື ເຫດຜົນອື່ນ. ໃນກໍລະນີເຫຼົ່ານີ້, ສິ່ງສຳຄັນຄືຕ້ອງໄດ້ລະບຸໃຫ້ຊັດເຈນວ່າ ນັກຮຽນທີ່ມີການປະກອບສ່ວນຈະບໍ່ຖືກລົງໂທດຍ້ອນວ່າການຈັບຄູ່ບໍ່ໄດ້ຜົນດີ.

ການຈັບຄູ່ຄວາມສາມາດທີ່ຄ້າຍຄືກັນ, ບໍ່ຈຳເປັນຕ້ອງມີຄວາມສາມາດເທົ່າທຽມກັນ ໃນຖານະເປັນຄູ່ຮ່ວມງານກັນ

ການຂຽນໂປຣແກຣມຈັບຄູ່ ທີ່ມີປະສິດທິພາບເມື່ອນັກຮຽນທີ່ມີຄວາມສາມາດຄ້າຍຄືກັນ, ເຖິງວ່າບໍ່ຈຳເປັນຕ້ອງມີຄວາມເທົ່າທຽມກັນຖືກຈັບຄູ່ເປັນຄູ່ຮ່ວມງານກັນ. ການຈັບຄຸ່ນັກຮຽນທີ່ບໍ່ກົງ ກັນມັກຈະເຮັດໃຫ້ການມີສ່ວນຮ່ວມທີ່ບໍ່ສົມດຸນກັນ. ຄຸສອນຕ້ອງເນັ້ນໜັກວ່າການຂຽນໂປຣແກຣມຈັບຄູ່ບໍ່ແມ່ນກົນລະຍຸດ "ແບ່ງປັນ ແລະ ເອົາຊະນະ", ແຕ່ເປັນຄວາມພະຍາຍາມຮ່ວມມືກັນ ທີ່ແທ້ຈິງ ໃນທຸກໆຄວາມພະຍາຍາມສຳລັບໂຄງການທັງໝົດ. ຄຸຄວນຫຼີກເວັ້ນການຈັບຄູ່ນັກຮຽນທີ່ອ່ອນຫຼາຍກັບນັກຮຽນທີ່ເກັ່ງຫຼາຍ..

ສ້າງແຮງຈູງໃຈໃຫ້ກັບນັກຮຽນໂດຍການສະເໜີສິ່ງຈູງໃຈເພີ່ມເຕີມ

ການສະເໜີແຮງຈຸງໃຈເພີ່ມເຕີມສາມາດຊ່ວຍກະຕຸ້ນນັກຮຽນໃຫ້ຈັບຄູ່, ໂດຍສະເພາະກັບນັກຮຽນທີ່ຮຽນເກັ່ງ. ຄຸບາງຄົນພົບວ່າການກຳນົດໃຫ້ນັກຮຽນຈັບຄູ່ພຽງແຕ່ໜຶ່ງ ຫຼື ສອງວຽກນັ້ນຈະມີ ຜົນດີ.

ຈາກສຸດ n! = n * (n-1) * (n-2) *...*2*1. ຈົ່ງຂຽນໂປຣແກຣມ 5! ແລະ 10! ໂດຍໃຊ້ຈຳນວນຖ້ວນ ແລະ ຕົວດຳເນີນການ * ແລ້ວພິມຄຳສັ່ງດັ່ງ ເພື່ອໃຫ້ສະແດງຜົນໄດ້ຮັບດັ່ງລຸ່ມນີ

Output Example

Calculate the factorial.

5! = 120

Calculate the factorial.

10! = 3628800