

Unit 2.

Planning for Programming

ການວາງແຜນເພື່ອການຂຽນໂປຣແກຣມ

● ຈຸດປະສົງຂອງການຮຽນ (Learning objectives)

- ✓ ສາມາດເຂົ້າໃຈນິຍາມ ແລະ ຄວາມຈຳເປັນຂອງການຄິດແບບຄຳນວນ
- ✓ ສາມາດຕິດຕັ້ງໂປຣແກຣມທີ່ຈຳເປັນສຳລັບການຂຽນໂປຣແກຣມ Python
- ✓ ສ້າງສະພາບແວດລ້ອມຈຳລອງເມື່ອ python 3.x ຖືກຕິດຕັ້ງແລ້ວ ແລະ ຕິດຕັ້ງ Python Version ທີ່ຕ່າງກັນ.
- ✓ ສາມາດຈຳແນກລຳດັບຂອງ codes ໄດ້ ເມື່ອເບິ່ງຜົນໄດ້ຮັບຈາກການນຳໃຊ້ຟັງຊັນ print() .
- ✓ ສາມາດຈຳແນກໄດ້ລະຫວ່າງຂໍ້ຜິດພາດແບບ syntax ແລະ ຂໍ້ຜິດພາດແບບ run-time ໄດ້.

● Learning overview

- ✓ ຈັດດຳລັບໂປຣແກຣມ, ແລະ ຂຽນເປັນ Pseudocode ຕາມລຳດັບໃຫ້ຖືກຕ້ອງ.
- ✓ ສາມາດສະແດງວິທີແກ້ບັນຫາໃນຊີວິດປະຈຳວັນດ້ວຍຂັ້ນຕອນວິທີ (algorithm) ໂດຍນຳໃຊ້ flowchart.
- ✓ ແປຄວາມໝາຍ pseudocode ຂອງຕົນເອງ ແລະ ຂຽນຄືນໃໝ່ໃນຮູບແບບ flowchart.
- ✓ ອະທິບາຍຄວາມແຕກຕ່າງລະຫວ່າງໂປຣແກຣມ ແລະ ຂັ້ນຕອນວິທີ (algorithm).
- ✓ ສາມາດເລືອກອັນທີ່ມີປະສິດທິພາບທີ່ສຸດຈາກຂັ້ນຕອນວິທີຫຼາຍອັນ (algorithms) ດ້ວຍສັນຍາລັກ Big O

● ແນວຄວາມຄິດທີ່ຈະຕ້ອງຮູ້ຈາກ Unit ທີ່ຜ່ານມາ

- ✓ ວິທີເປີດໃຊ້ Jupyter notebook ໃນສະພາບແວດລ້ອມຈຳລອງ
- ✓ ການພິມຂໍ້ຄວາມເທິງໜ້າຈຳນຳໃຊ້ຟັງຊັນ print()
- ✓ ສາມຮູບແບບໂຄງສ້າງໃນການຄວບຄຸມໂປຣແກຣມ Python: ລຳດັບ (Sequential), ການເລືອກ (Selection), ການເຮັດຊ້ຳ (Repetition)

Keywords ຄໍາສັບສໍາຄັນ

Pseudocode

Flowchart

Algorithm

Linear search

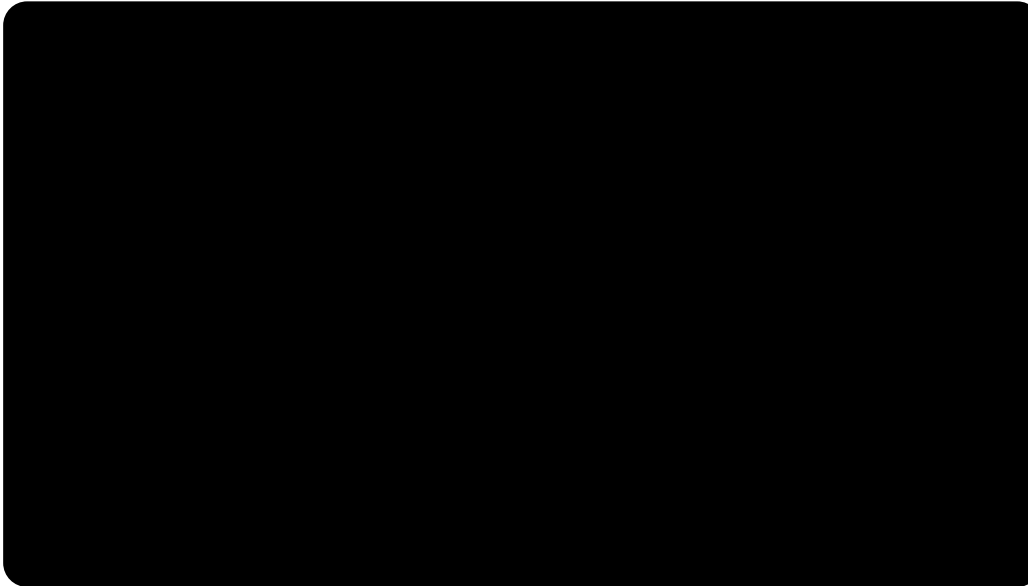
Big O notation

| Mission

1. Real world problem

1.1. ຂະບວນການຄົ້ນພົບບັນຫາ ແລະ ຊອກຫາວິທີແກ້ໄຂ

※ ເພື່ອເບິ່ງຄລິບວິດີໂອ, ເອົາ mouse ວາງເທິງ box ແລະ ປຸ່ມຫຼັ້ນປະກົດຂຶ້ນ. ຄລິກມັນເພື່ອເບິ່ງ.

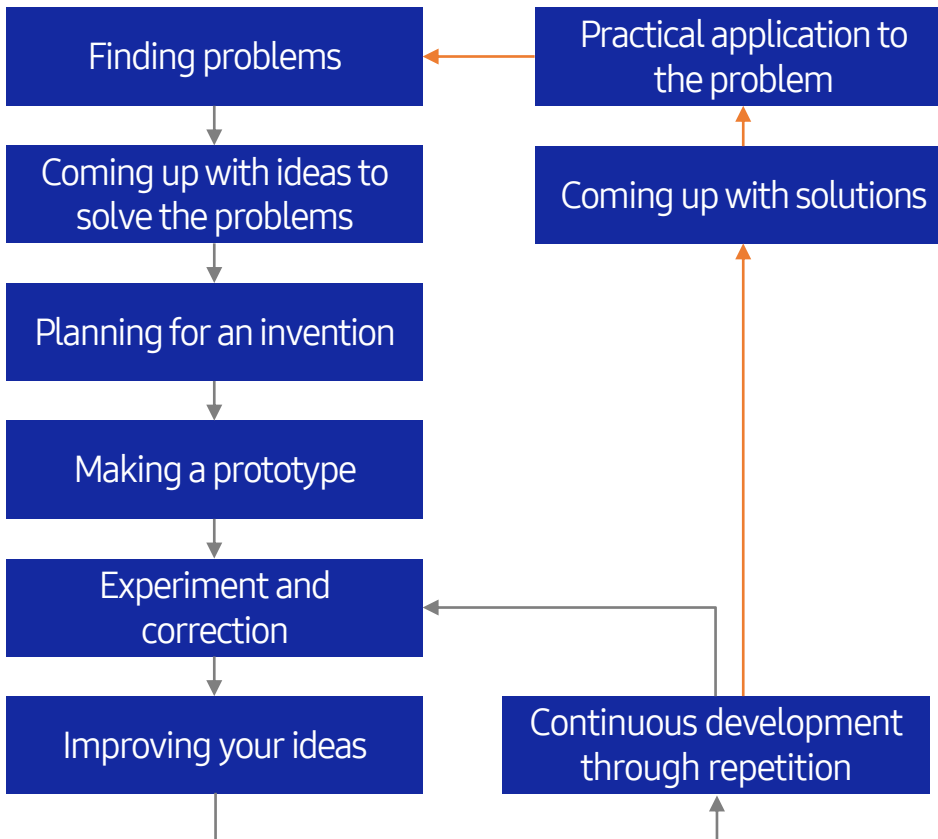


https://www.ted.com/talks/william_kamkwamba_how_i_harnessed_the_wind/transcript

- ▶ ສິ່ງປະດິດແມ່ນຜະລິດຕະພັນ, ເຕັກໂນໂລຢີ, ຫຼືຂະບວນການທີ່ສ້າງຂຶ້ນໃຫມ່ເພື່ອແກ້ໄຂບັນຫາຕ່າງໆ.
- ▶ William Kamkwamba ເກີດຢູ່ໃນມາລາວີປະເທດອາຟຣິກາບ່ອນທີ່ຄວາມອຶດຢາກຢ່າງຮ້າຍແຮງ.
- ▶ ຄອບຄົວຂອງລາວປະສົບກັບຄວາມຫຍຸ້ງຍາກໃນການປູກຝັງ ແລະ ຊີວິດການເປັນຢູ່ພື້ນຖານຍ້ອນຂາດນ້ຳ ແລະ ໄຟຟ້າ. ເພື່ອຊ່ວຍຄອບຄົວຂອງລາວ, William ອາຍຸພຽງ 14 ປີ ໄດ້ປະດິດເຄື່ອງຈັກຈາກພະລັງງານລົມຂະໜາດນ້ອຍ ທີ່ລວບລວມເສດເຫຼັກຕາມຄວາມຄິດຂອງລາວ ແລະ ສາມາດແກ້ບັນຫານ້ຳ ແລະ ໄຟຟ້າທີ່ຄອບຄົວຂອງລາວກຳລັງພົບໄດ້.
- ▶ ເປັນເຊັ່ນນີ້, ການປະດິດບໍ່ແມ່ນເລື່ອງຍາກ,ມັນເລີ່ມຕົ້ນດ້ວຍການຄົ້ນພົບບັນຫາຕ່າງໆ ໃນໂລກແຫ່ງຄວາມເປັນຈິງ ອ້ອມຕົວເຮົາ.
- ▶ ໃນຂະນະທີ່ຮຽນ Python ໃນຫຼັກສູດນີ້ ຜູ້ຮຽນຈະໄດ້ຝຶກຝົນຂະບວນການຄົ້ນຫາບັນຫາໃນຊີວິດປະຈຳວັນ ແລະ ຄົ້ນຫາວິທີແກ້ບັນຫາ.

1. Real world problem

1.2. ຂະບວນການປະດິດເຄື່ອງຈັກລົມຂອງ William

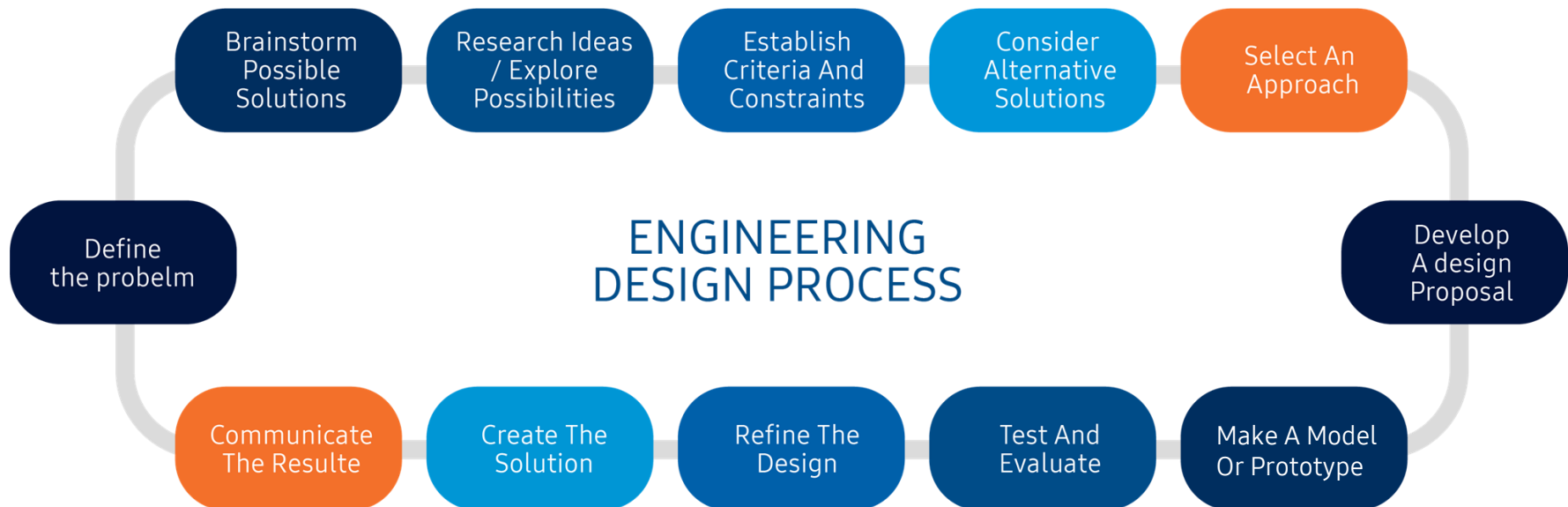


- ▶ ຮູບເບື້ອງຊ້າຍມີແມ່ນຂະບວນການຈົງທີ່ William ດຳເນີນການເພື່ອຫາແນວຄິດໃນການແກ້ບັນຫາທີ່ຄົ້ນພົບ ແລະ ປ່ຽນໃຫ້ເປັນສິ່ງປະດິດທີ່ເປັນຮູບປະທຳ.
- ▶ ຂະບວນການນີ້ບໍ່ພຽງແຕ່ William ເທົ່ານັ້ນ, ແຕ່ຍັງລວມເຖິງນັກວິທະຍາສາດ ແລະ ນັກປະດິດຈຳນວນຫຼາຍ ນຳມາໃຊ້ໃນການປະດິດບາງສິ່ງບາງຢ່າງ.
- ▶ ສິ່ງນີ້ເອີ້ນວ່າຂັ້ນຕອນການປະດິດ ຫຼື ຂະບວນການອອກແບບທາງວິສະວະກຳ (EDP: Engineering Design Process).
- ▶ ທຸກໆຂັ້ນຕອນແມ່ນມີຄວາມສຳຄັນຫຼາຍ, ແນວໃດກໍຕາມ, ສ່ວນທີ່ສຳຄັນທີ່ສຸດແມ່ນການວາງແຜນແນວຄວາມຄິດທີ່ແນ່ນອນ ແລະ ສືບຕໍ່ປັບປຸງແນວຄວາມຄິດໂດຍຜ່ານການທົດລອງ, ການແກ້ໄຂ, ແລະ ຄຳຄິດຄຳເຫັນ.
- ▶ ການຂຽນໂປຣແກຣມກໍເຊັ່ນດຽວກັນ, ຖ້າໃຜຜູ້ຫນຶ່ງຕ້ອງການເຮັດບາງສິ່ງບາງຢ່າງ, ລາວຜູ້ນັ້ນຄວນເລີ່ມວາງແຜນຄວາມຄິດໂດຍລະອຽດ, ແທນທີ່ຈະເຮັດວຽກໂດຍກົງໃນຄອມພິວເຕີ.

2. Solution

2.1. ຂັ້ນຕອນການອອກແບບທາງວິສະວະກຳ

- EDP (Engineering Design Process) ແມ່ນຊຸດຂອງຂັ້ນຕອນທີ່ວິສະວະກຳ ແລະ ນັກວິທະຍາສາດປະຕິບັດຕາມເພື່ອຊອກຫາວິທີແກ້ໄຂບັນຫາ. ແຕ່ລະຂັ້ນຕອນປະກອບດ້ວຍຂະບວນການກຳນົດເປົ້າໝາຍແລະຂໍ້ຈຳກັດ, ການສ້າງຕົວແບບ, ການທົດສອບ ແລະ ການປະເມີນຜົນຢ່າງຕໍ່ເນື່ອງ.
- ແຕ່ລະ EDP ມີ 5 ຫາ 7 ຂັ້ນຕອນ ຫຼື ຫຼາຍກວ່ານັ້ນ, ແຕ່ຂັ້ນຕອນລະອຽດທີ່ສຸດແມ່ນສະແດງຢູ່ໃນຮູບລຸ່ມນີ້.



3. Mission

3.1. ສະແດງບັນຫາລຸ່ມນີ້ເປັນແຜນຜັງ

- ຂະບວນການ EDP ຍັງສາມາດຖືກນຳໃຊ້ກັບວິສະວະກຳຄອມພິວເຕີ ແລະ ການຂຽນໂປຣແກຣມ.
- ໃນການຂຽນໂປຣແກຣມ, ຂັ້ນຕອນການວາງແຜນ (Develop A Design Proposal) ສາມາດສະແດງອອກເປັນ pseudocode ແລະ flowchart.

[MISSION]

ໃຫ້ພິຈາລະນາວ່າຕອນນີ້ເຈົ້າກຳລັງເດີນທາງ.

ເຈົ້າຕ້ອງຊອກຫາໂຮງແຮມທີ່ເຈົ້າຈອງໄວ້, ແຕ່ເຈົ້າຫຼົງທາງ.

ດັ່ງນັ້ນ, ເຈົ້າກຳລັງຖາມຄົນຢ່າງຕາມທາງ ແລະ ລາວແນະນຳເຈົ້າ ດັ່ງລຸ່ມນີ້.

“ໄປຊື່ທາງທິດຕາເວັນຕົກຕາມສີ່ແຍກນີ້. ເຈົ້າຈະເຫັນ 4 ຕຶກ.

ເມື່ອເຈົ້າເຫັນຫ້ອງການໄປສະນີ, ລຽວຂວາຂ້າມຫ້ອງການໄປສະນີ.

ໄປຊື່ ແລະ ເມື່ອເຈົ້າເຫັນຮ້ານອາຫານອິຕາລີ, ຕຶກທີ່ສາມແມ່ນໂຮງແຮມທີ່ເຈົ້າກຳລັງຊອກຫາ.”

ເຈົ້າຍັງໃໝ່ກັບທຸກຢ່າງໃນພື້ນທີ່ນີ້.

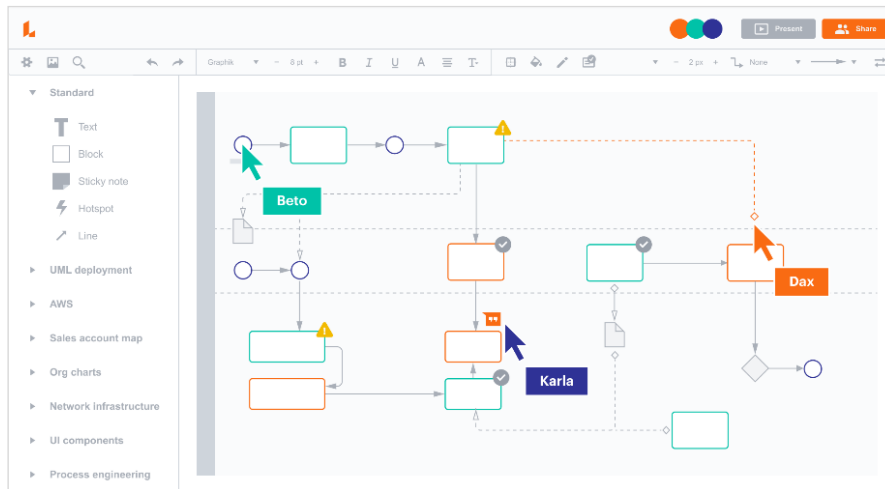
ຫາກຕ້ອງການຄື້ນຫາໂຮງແຮມໃຫ້ສຳເລັດ ເຈົ້າຈະຕ້ອງຄື້ນຫາໂຮງແຮມຫຼັງຈາກສ້າງ flowchart ຂອງຄຳແນະນຳທີ່ເຈົ້າໄດ້ຮັບ.

3. ໂຈດບັນຫາ (Mission)

3.2. ແຫຼ່ງອ້າງອີງສໍາລັບການແກ້ໄຂບັນຫາ (References for solving missions)

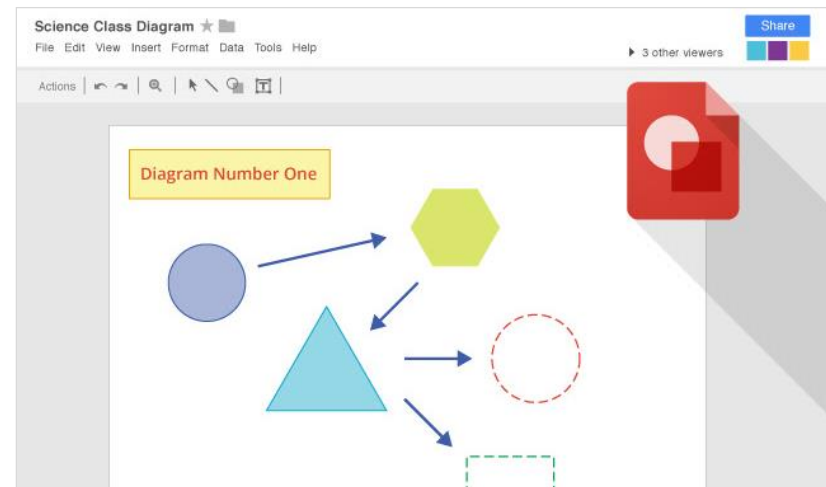
- Program ລຸ່ມນີ້ແມ່ນ digital tools ທີ່ຊ່ວຍທ່ານແຕ້ມ flowchart ໄດ້ຢ່າງງ່າຍດາຍ.
- ມີເຄື່ອງຈໍານວນຫຼາຍ, ແຕ່ໂຕທີ່ແນະນຳລຸ່ມນີ້ແມ່ນ free ແລະ ງ່າຍໃນການນຳໃຊ້

Lucid chart



<https://www.lucidchart.com/pages/>

Google drawings



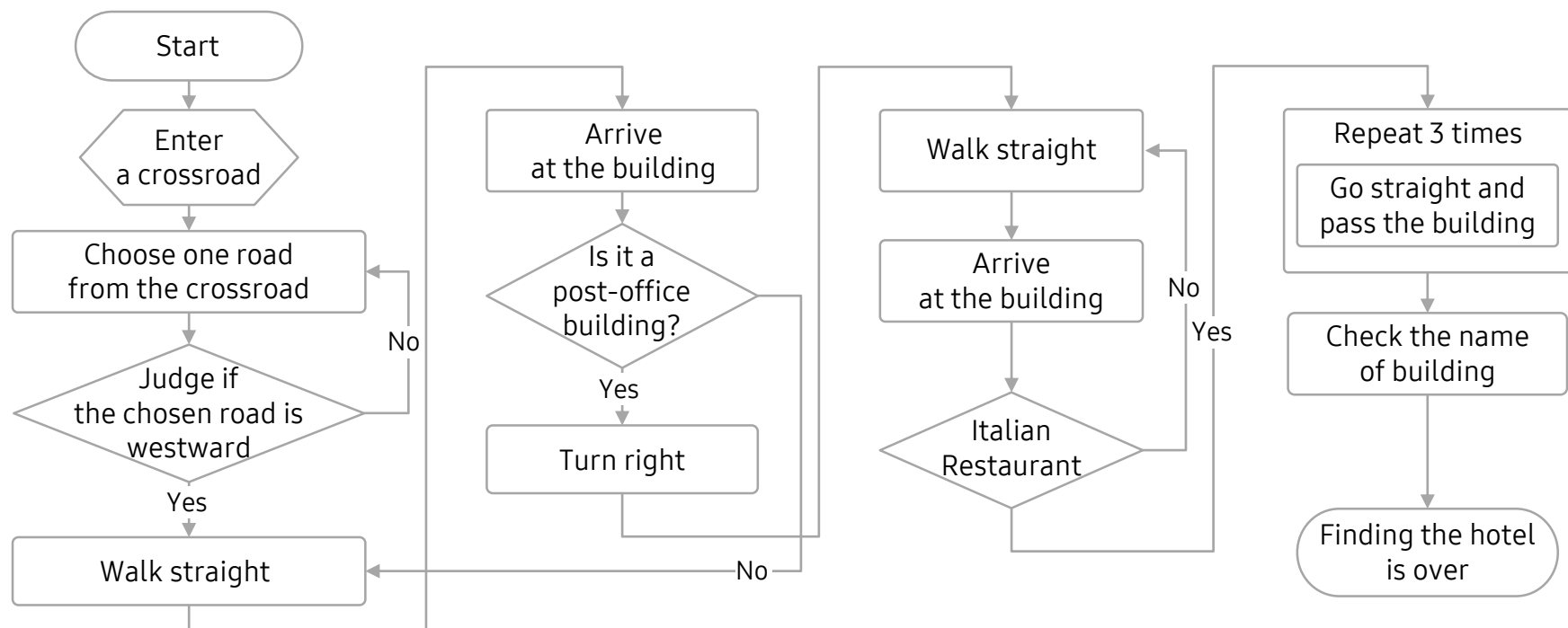
Search at <https://chrome.google.com>

3. Mission

3.3. flowchart ແລະ Final Code ທົດທາງຂອງ flowchart

I ຂ້າງລຸ່ມນີ້ແມ່ນສະແດງ flowchart .

ເຮົາສາມາດ ລຳດັບຂະບວນການໜ້າວຽກໃນການແກ້ໄຂບັນຫາ.



|

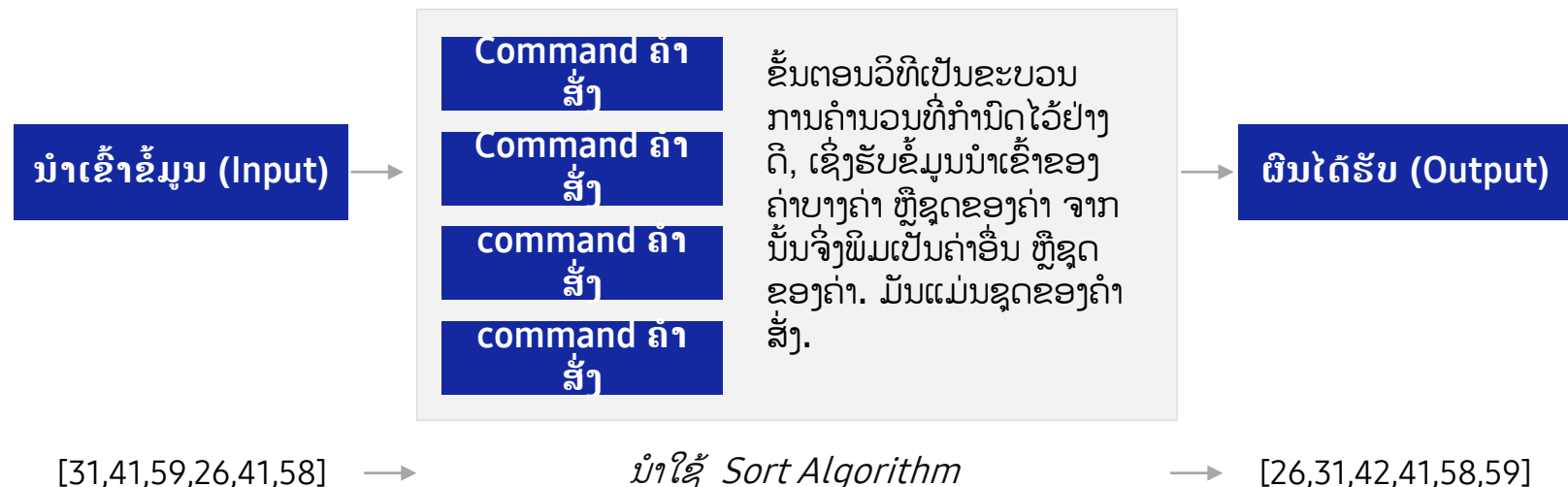
Key concept

1. Algorithm

1.1. ນິຍາມຂອງຂັ້ນຕອນວິທີ

- ຂັ້ນຕອນວິທີແມ່ນກຸ່ມຂອງຄໍາສັ່ງສໍາລັບເຮັດວຽກບາງຢ່າງ
- ຂັ້ນຕອນວິທີຄວນມີລໍາດັບການຄໍານວນຢ່າງມີເຫດຜົນ ແລະ ໜ້າວຽກທີ່ເໝາະສົມໃນການແກ້ບັນຫາ, ບໍ່ແມ່ນພຽງແຕ່ເປັນຊຸດຄໍາສັ່ງ.
- ພວກເຮົາສາມາດວາງແຜນສໍາລັບການສ້າງ algorithm, ເຊິ່ງເປັນຂັ້ນຕອນທີ່ປະກອບດ້ວຍຄໍາສັ່ງສໍາລັບການແກ້ໄຂບັນຫາ. ຂະບວນການວາງແຜນສາມາດສ້າງໄດ້ດ້ວຍ pseudocode ແລະ flowchart.

ຂັ້ນຕອນວິທີ (Algorithm)



2. ການນຳສະເໜີ ແລະ ການວາງແຜນຂັ້ນຕອນວິທີ

2.1. Pseudocode

- | Sequence ແມ່ນຄອມພິວເຕີປະຕິບັດຄຳສັ່ງຈາກ code ຕາມລຳດັບ.
- | Pseudocode ເປັນຕົວແທນຂອງສູດການຄິດໄລ່ໂດຍໃຊ້ພາສາໃນຊີວິດປະຈຳວັນ. Pseudocode ສາມາດເປັນການເລີ່ມຕົ້ນທີ່ດີສຳລັບການພັດທະນາ algorithm ໃນເວລາທີ່ທ່ານບໍ່ມີຄວາມຮູ້ພື້ນຖານພຽງພໍກ່ຽວກັບການຂຽນໂປຣແກຣມ.
- | Pseudocode ມີບົດບາດເປັນຂົວຕໍ່ສຳລັບການເຮັດ flowchart ໃຫ້ສຳເລັດ ແລະ code ຕົວຈິງ. ມັນມັກຈະຖືກນຳໃຊ້ໂດຍນັກຂຽນໂປຣແກຣມເພື່ອເປັນຄຳແນະນຳພື້ນຖານໃນເວລາທີ່ພວກເຂົາຂຽນ code ຕົວຈິງ.
- | ມັນເປັນສິ່ງຈຳເປັນທີ່ສຸດທີ່ທ່ານຄວນອະທິບາຍຢ່າງຊັດເຈນ ແລະ ວາງແຜນສິ່ງທີ່ແຕ່ລະແຖວຂອງ code ຄວນປະຕິບັດ.

ໂປຣແກຣມນີ້ຈະຊ່ວຍໃຫ້ຜູ້ໃຊ້ສາມາດກວດສອບຕົວເລກວ່າ ຈະເປັນເລກຄູ່ ຫຼື ເລກຄີກ.

If “4”

print result
“This number is even”

If “3”

print result
“This number is odd”

- ▶ ຂຽນປະໂຫຍກທີ່ອະທິບາຍຈຸດປະສົງຂອງ algorithm ໃນຕອນເລີ່ມຕົ້ນ.
- ▶ ຂຽນສິ່ງ ກະທັດຮັດ.
- ▶ ສ້າງ ແລະ ນຳໃຊ້ກົດລະບຽບຂອງຊື່ທີ່ງ່າຍ ແລະ ແຕກຕ່າງກັນເພື່ອນຳໃຊ້.
- ▶ ການຫຍໍ້ໜ້າ ແລະ ຍະຫວ່າງແມ່ນຈຸດສຳຄັນ
- ▶ ສະແດງໂຄງສ້າງລຳດັບ, ໂຄງສ້າງການຄັດເລືອກ, ແລະ ໂຄງສ້າງການວິນຊ້າໂດຍໃຊ້ if, for, while.

2. ການນຳສະເໜີ ແລະ ການວາງແຜນຂັ້ນຕອນວິທີ

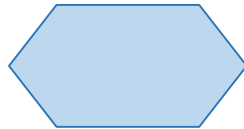
2.2. Flowchart

- Flowchart ສະແດງຄຳສັ່ງດ້ວຍຮູບຊົງມາດຕະຖານຫຼາຍຮູບແບບ ແລະ ສະແດງຂັ້ນຕອນການເຮັດວຽກໂດຍໃຊ້ລູກສອນ.
- ສຳລັບຮູບຮ່າງ ແລະ ສັນຍາລັກ ຈະໃຊ້ຮູບຮ່າງ ແລະ ສັນຍາລັກທີ່ກຳນົດໂດຍອົງການລະຫວ່າງປະເທດເພື່ອການເປັນມາດຕະຖານ (ISO)
- ສິ່ງທີ່ສຳຄັນແມ່ນ, ຕ້ອງແຕ້ມແຕ່ເທິງລົງລຸ່ມ ແລະ ຈາກຊ້າຍໄປຫາຂວາໂດຍບໍ່ທັບຊ້ອນກັນ.



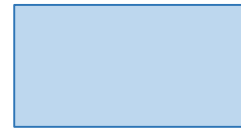
Start, End

ສະແດງເຖິງ ຈຸດເລີ່ມຕົ້ນ
ແລະ ຈຸດສິ້ນສຸດຂອງ
flowchart



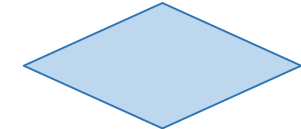
Preparation

ເລີ່ມຕົ້ນຂໍ້ມູນ



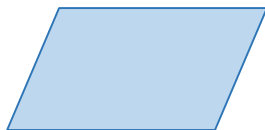
Action or Process

ດຳເນີນການທາງຄະນິດສາດ/ການປະ
ມວນຜົນຂັ້ນຕອນດຽວ



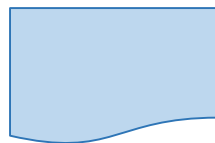
Decision

ໃຊ້ເພື່ອແຍກໄປສ່ວນຕ່າງໆຂອງ
flowchart ໂດຍພິຈາລະນາຈາກເງື່ອນໄຂ



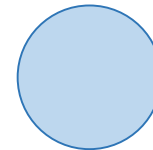
Input, Output

ໃຊ້ສຳລັບຂໍ້ມູນເຂົ້າ
ແລະ ຂໍ້ມູນອອກ



Document

ສະແດງເອກະສານ ຫຼື
ລາຍງານທີ່ພິມອອກ



On-page Connector

ຈຸດເຊື່ອມຕໍ່ກັບ flowchart



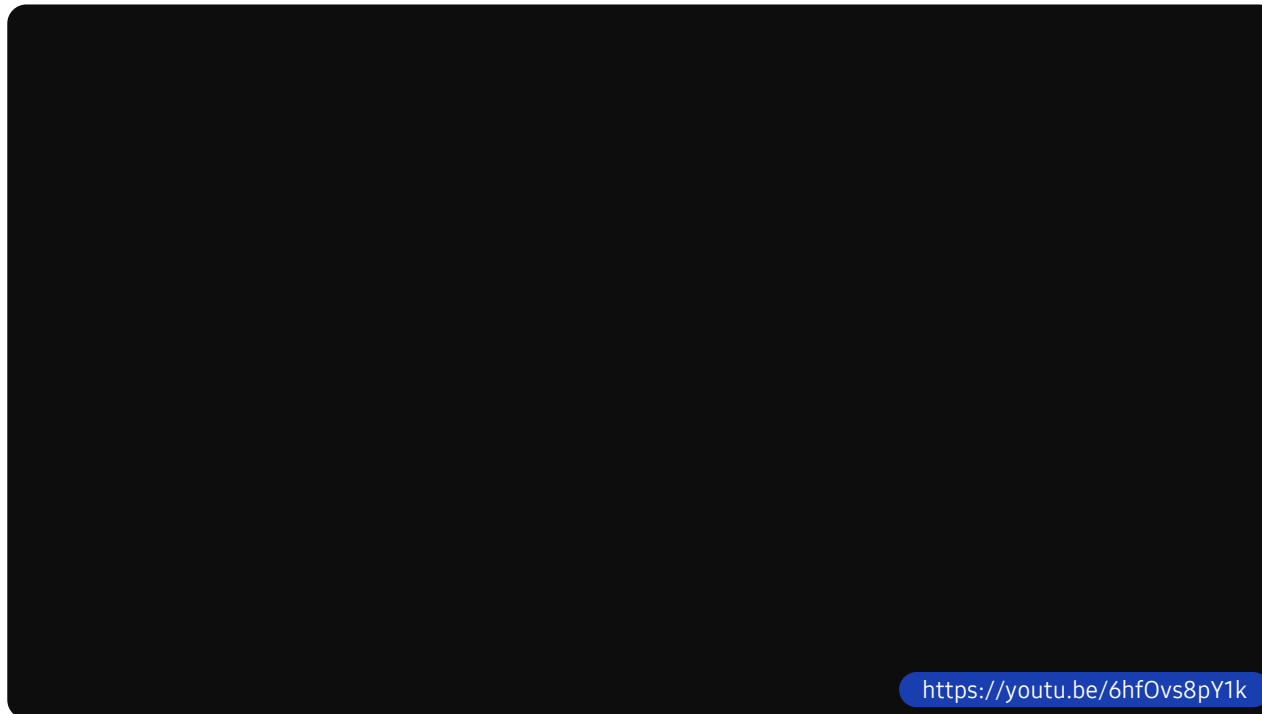
Flow Arrow

ສະແດງເຖິງເສັ້ນທາງການໄຫຼ
ຂອງການປະມວນຜົນ

3. ປະສິດທິພາບຂອງ Algorithm

3.1. ເຮົາຄວນເລືອກ ແລະ ໃຊ້ algorithm ໃດ?

※ ເພື່ອເບິ່ງຄລິບວິດີໂອ, ເອົາ mouse ວາງເທິງ box ແລະ ປຸ່ມຫຼັ້ນປະກົດຂຶ້ນ. ຄລິກມັນເພື່ອເບິ່ງ.



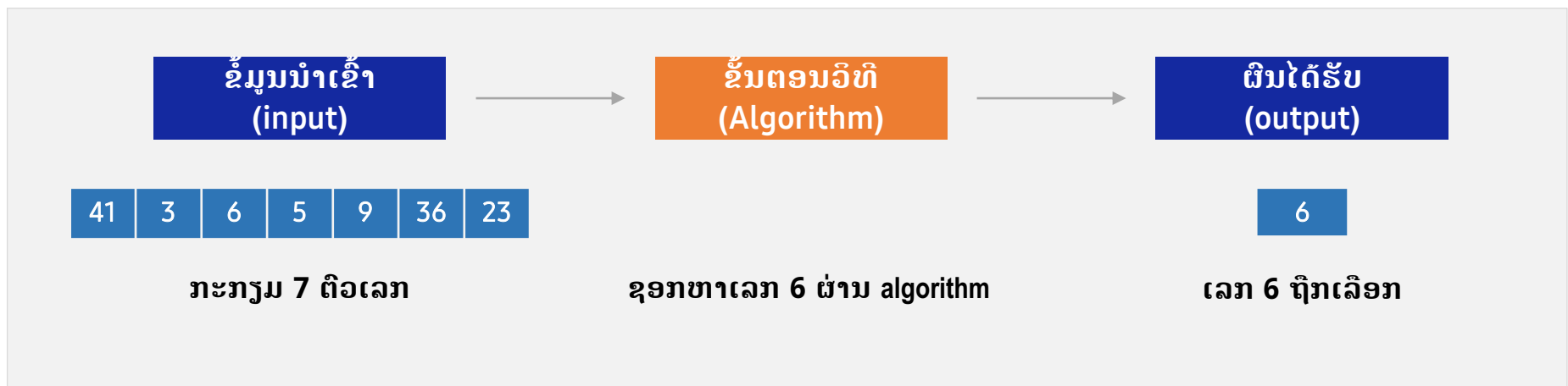
What's an algorithm - David Malan

3. ປະສິດທິພາບຂອງ Algorithm

3.2. ວິທີໃດມີປະສິດທິພາບຫຼາຍກວ່າ?

- I ກະກຽມຂໍ້ມູນ 7 ຕົວເລກ [array] ໂດຍບໍ່ຈັດລຳດັບແຕ່ຈັດລຽງ.
- I ໃຫ້ພວກເຮົາຄິດກ່ຽວກັບປະສິດທິພາບຂອງ algorithm ໂດຍເບິ່ງສອງປະເພດຂອງ algorithms ທີ່ຊອກຫາຄ່າສະເພາະໃນຂໍ້ມູນທີ່ກະກຽມ.

ເມື່ອທ່ານໄດ້ຮັບຜົນໄດ້ຮັບທີ່ຕ້ອງການໂດຍຜ່ານຂັ້ນຕອນວິທີ,
ສາມາດເວົ້າໄດ້ບໍ່ວ່າ algorithm ທີ່ໃຊ້ແມ່ນເປັນອັນທີ່ມີປະສິດທິພາບ?



3. ປະສິດທິພາບຂອງ Algorithm

3.3. ຂັ້ນຕອນວິທີ A ເພື່ອຊອກຫາຄ່າ 6

- ຂັ້ນຕອນວິທີນີ້ (algorithm) ງ່າຍຫຼາຍ. ມັນຈະກວດເບິ່ງຂໍ້ມູນເທື່ອລະລາຍການຕັ້ງແຕ່ເລີ່ມຕົ້ນຈົນກ່ວາມັນຄົ້ນພົບຂໍ້ມູນທີ່ຕ້ອງການ
- ແບບນີ້ເອີ້ນວ່າ ເປັນ algorithm ຄົ້ນຫາແບບເສັ້ນຊື່ຍ້ອນວ່າມັນຊອກຫາຂໍ້ມູນຕາມລຳດັບ.

Step 1

41	3	6	5	9	36	23
----	---	---	---	---	----	----

- ປຽບທຽບ 41, ຂໍ້ມູນທຳອິດກັບ 6, ຂໍ້ມູນທີ່ຈະພົບເຫັນ,.
- ຍ້າຍໄປໜຶ່ງຊ່ອງໄປທາງຂວາເພາະວ່າມັນບໍ່ກົງກັບຂໍ້ມູນທີ່ຕ້ອງຄົ້ນຫາ.

Step 2

41	3	6	5	9	36	23
----	---	---	---	---	----	----

- ປຽບທຽບ 3, ຂໍ້ມູນທີສອງກັບ 6, ຂໍ້ມູນທີ່ຈະພົບເຫັນ,.
- ຍ້າຍໄປໜຶ່ງຊ່ອງໄປທາງຂວາເພາະວ່າມັນບໍ່ກົງກັບຂໍ້ມູນທີ່ຕ້ອງຄົ້ນຫາ.

Step 3

41	3	6	5	9	36	23
----	---	---	---	---	----	----

- ປຽບທຽບ 6, ຂໍ້ມູນທີສອງກັບ 6, ຂໍ້ມູນທີ່ຈະພົບເຫັນ,.
- ຢຸດການເຄື່ອນໄຫວເນື່ອງຈາກມັນກົງກັບຂໍ້ມູນທີ່ຄົ້ນຫາ.

**ໃນຕົວຢ່າງນີ້,
ດຳເນີນການຄຳສັ່ງ 3 ຄັ້ງເພື່ອປຽບທຽບ
ຕົວເລກທີ່ກຳນົດເພື່ອຄົ້ນຫາເລກ 6 ທີ່
ຕ້ອງການ**

**ຖ້າຈຳນວນຂໍ້ມູນທັງຫມົດມີຈຳນວນ
ຫຼາຍ ຫຼືບໍ່ມີຂໍ້ມູນທີ່ຈະຊອກຫາ, ຈະ
ເກີດຫຍັງຂຶ້ນ?**

**ມັນຕ້ອງປຽບທຽບ ແລະ ສືບຕໍ່ໄປຈັກ
ເທື່ອ?**

3. ປະສິດທິພາບຂອງ Algorithm

3.4. ຂັ້ນຕອນວິທີ B ເພື່ອຊອກຫາຄ່າ 6

I ກ່ອນທີ່ຈະເລີ່ມ algorithm B, ຈັດລຽນຂໍ້ມູນຕາມລຳດັບຕົວເລກກ່ອນ.

3	5	6	9	23	36	41
---	---	---	---	----	----	----

I ປຽບທຽບຂໍ້ມູນຢູ່ເຄິ່ງກາງຂອງຊຸດຂໍ້ມູນດ້ວຍ 6 ແລະ ຫຼັງຈາກນັ້ນຫຼຸດພື້ນທີ່ຄົ້ນຫາເປັນເຄິ່ງຫນຶ່ງ.

(ແບບນີ້ເອີ້ນວ່າ ການຄົ້ນຫາແບບ binary)

Step 1

3	5	6	9	23	36	41
---	---	---	---	----	----	----

- ຊອກຫາຄ່າກາງຂອງຂໍ້ມູນ (median value)
- ປຽບທຽບກັບ 6, ຂໍ້ມູນທີ່ຈະພົບເຫັນ,
- ເນື່ອງຈາກ 9 ແມ່ນໃຫຍ່ກວ່າ 6, ມັນ infers ໄດ້ວ່າ 6 ແມ່ນຢູ່ໃນທິດທາງຊ້າຍຈາກສູນກາງ (center)

Step 2

3	5	6
---	---	---

- ແຍກຄ່າໃນທິດທາງທີ່ບໍ່ຈຳເປັນອອກຈາກຂອບເຂດຂອງການຄົ້ນຫາ
- ຊອກຫາຄ່າກາງຂອງຂໍ້ມູນ (median value) ອີກຄັ້ງໃນຊຸດຂໍ້ມູນນີ້.
- ເນື່ອງຈາກ 5 ແມ່ນນ້ອຍກວ່າ 6, ມັນ infers ໄດ້ວ່າ 6 ແມ່ນຢູ່ໃນທິດທາງຂວາຈາກສູນກາງ (center).

Step 3

6

- ແຍກຄ່າໃນທິດທາງທີ່ບໍ່ຈຳເປັນອອກຈາກຂອບເຂດຂອງການຄົ້ນຫາ
- ພົບຄ່າທີ່ຕ້ອງການ

ເພື່ອຊອກຫາຕົວເລກທີ່ຕ້ອງການ, ໃນຂະນະທີ່ປຽບທຽບຈາກສູນກາງ, ໄລຍະການຄົ້ນຫາຂອງຂໍ້ມູນຖືກຫຼຸດລົງເຄິ່ງຫນຶ່ງ. ເມື່ອພົບມູນຄ່າທີ່ຕ້ອງການ, ຂະບວນການສິ້ນສຸດລົງ.

ຖ້າຈຳນວນຂໍ້ມູນທັງຫມົດມີຈຳນວນຫຼາຍ ຫຼືບໍ່ມີຂໍ້ມູນທີ່ຈະຊອກຫາ, ຈະເກີດຫຍັງຂຶ້ນ?

ລອງຄິດເບິ່ງເມື່ອປຽບທຽບກັບ Algorithm A.

3. ປະສິດທິພາບຂອງ Algorithm

3.5. ປຽບທຽບ ຂັ້ນຕອນວິທີ A ແລະ B

- ໄດ້ຮັບການພິສູດແລ້ວວ່າເຖິງແມ່ນວ່າ algorithm ໃຫ້ຜົນໄດ້ຮັບຕາມທີ່ເຮົາຕ້ອງການ, ແຕ່ກໍບໍ່ສາມາດເປັນ algorithm ທີ່ມີປະສິດທິພາບໄດ້.
- ການຮຽນຮູ້ Python, ສິ່ງສໍາຄັນແມ່ນຮັບຜົນໄດ້ຮັບ. ແຕ່ມັນກໍເປັນສິ່ງຈໍາເປັນທີ່ຕ້ອງຊອກຫາສູດການຄິດໄລ່ທີ່ມີປະສິດທິພາບຫຼາຍຂຶ້ນ.

Algorithm A Linear search

1	2	3	...	99	100
---	---	---	-----	----	-----

- ມີຂໍ້ມູນໜຶ່ງຮ້ອຍ
- ສໍາລັບການຊອກຫາຄ່າທີ່ຕ້ອງການ, ຕ້ອງປຽບທຽບຂໍ້ມູນ 100 ເທື່ອໃນກໍລະນີທີ່ຮ້າຍແຮງທີ່ສຸດ.

ຖ້າຈໍານວນຂໍ້ມູນແມ່ນ n ,
ໜຶ່ງຮ້ອຍ (100) ຂໍ້ມູນ

n ຂັ້ນຕອນ
100 ຂັ້ນຕອນ

Algorithm B binary search

1	2	3	...	99	100
---	---	---	-----	----	-----

- ມີຂໍ້ມູນໜຶ່ງຮ້ອຍ
- ເນື່ອງຈາກຊ່ວງການຊອກຫາແມ່ນຫຼຸດລົງເຄິ່ງໜຶ່ງ, ຈໍານວນຂໍ້ມູນທີ່ຈະປຽບທຽບແມ່ນຫຼຸດລົງ $100 \rightarrow 50 \rightarrow 25 \rightarrow 13 \rightarrow 7 \rightarrow 4 \rightarrow 2 \rightarrow$ the result.
- ຕົວເລກໃດກໍຕາມທີ່ທ່ານຕ້ອງການ, ທ່ານສາມາດຊອກຫາຜົນໄດ້ຮັບໄດ້ໃນ 7 ຂັ້ນຕອນ.

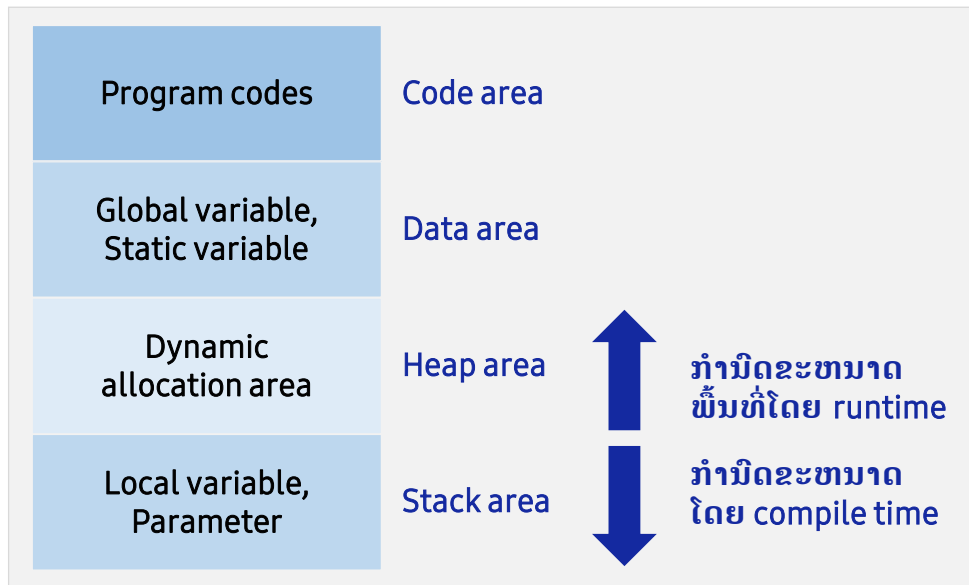
ຖ້າຈໍານວນຂໍ້ມູນແມ່ນ n ,
ໜຶ່ງຮ້ອຍ (100) ຂໍ້ມູນ

$\log_2 n$ ຂັ້ນຕອນ
 $\log_2 100 = 7$ ຂັ້ນຕອນ

4. Algorithm performance

4.1. ໂຄງສ້າງໜ່ວຍຄວາມຈຳ

- ມີສອງວິທີຕົ້ນຕໍໃນການວັດແທກການປະຕິບັດຂອງ algorithm: **space complexity**, ເຊິ່ງວັດແທກວ່າຄອມພິວເຕີໃຊ້ໜ່ວຍຄວາມຈຳຫຼາຍປານໃດ ແລະ **time complexity** ແມ່ນເວລາໃນການແກ້ບັນຫາເທົ່າໃດ.
- Space complexity** ແມ່ນການວັດແທກວ່າຄອມພິວເຕີໃຊ້ໜ່ວຍຄວາມຈຳຫຼາຍປານໃດ.
- ໜ່ວຍຄວາມຈຳທີ່ໃຊ້ໃນຄອມພິວເຕີຂອງພວກເຮົາເປັນແນວໃດ, ແລະ ເປັນຫຍັງພວກເຮົາຕ້ອງວັດແທກການນຳໃຊ້ໜ່ວຍຄວາມຈຳ?

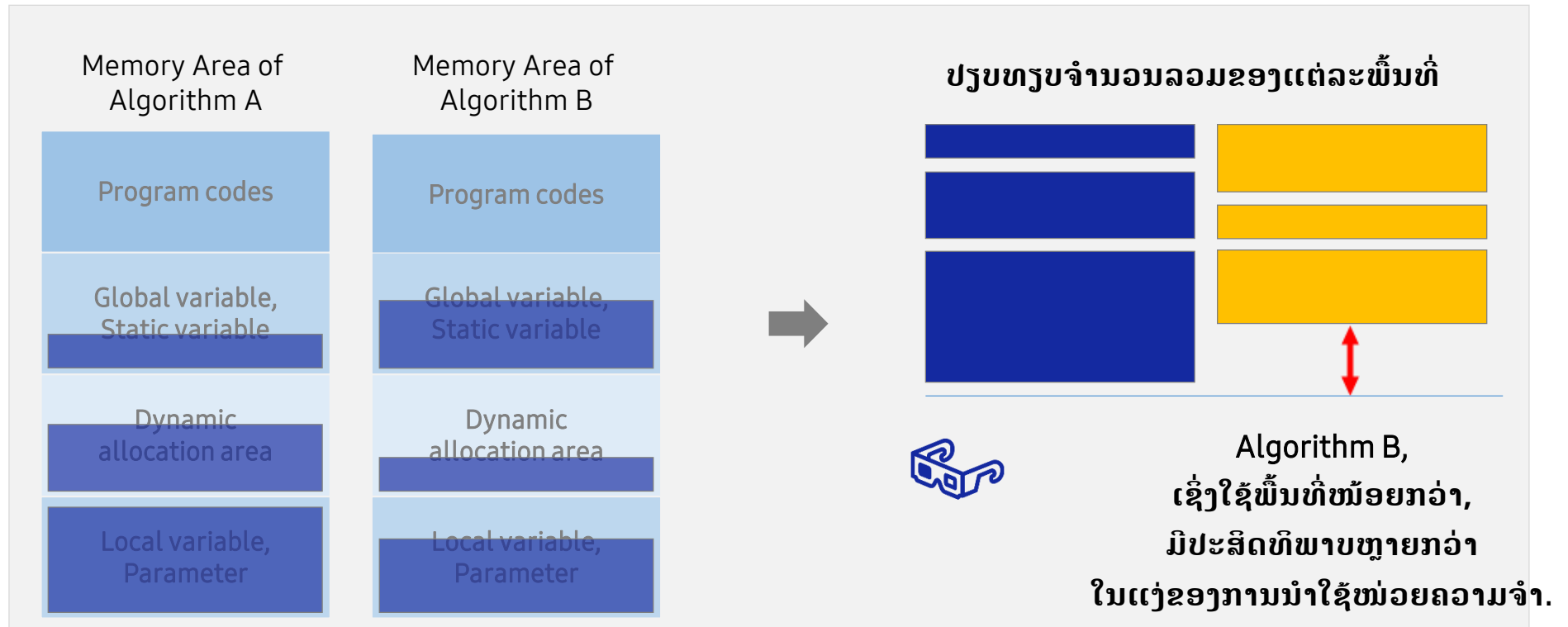


- **Code area:** ແມ່ນພື້ນທີ່ເກັບ code ໂປຣແກຣມທີ່ຈະເອີ້ນໃຊ້. CPU ຮັບຄຳສັ່ງເທື່ອລະຄຳສັ່ງ ແລະ ປະມວນຜົນຄຳສັ່ງເຫຼົ່ານັ້ນ.
- **Data area:** ພື້ນທີ່ການຈັດສັນຕົວປ່ຽນແບບ global ແລະ ຕົວປ່ຽນແບບຄົງທີ່ (static). ມັນຖືກຈັດສັນເມື່ອໂປຣແກຣມເລີ່ມຕົ້ນ ແລະ ຖືກທຳລາຍເມື່ອໂປຣແກຣມສິ້ນສຸດ.
- **Heap area:** ແມ່ນພື້ນທີ່ຈັດການໂດຍຜູ້ຂຽນໂປຣແກຣມ (Programmer)
- **Stack area:** ແມ່ນພື້ນທີ່ທີ່ເກັບຕົວປ່ຽນແບບ local ແລະ parameters ທີ່ກ່ຽວຂ້ອງກັບການເອີ້ນໃຊ້ຟັງຊັນ ຈະຖືກທຳລາຍເມື່ອການເອີ້ນໃຊ້ຟັງຊັນສຳເລັດ.

4. Algorithm performance

4.2. Space Complexity

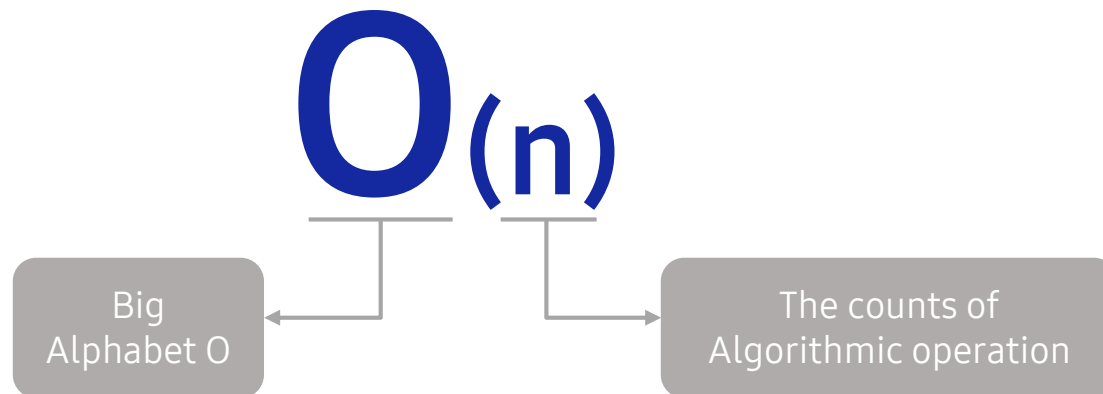
- ໜ່ວຍຄວາມຈຳຂອງຄອມພິວເຕີຂອງເຮົາມີສັບພະຍາກອນຈຳກັດ. ດັ່ງນັ້ນ, algorithm ທີ່ໃຊ້ພື້ນທີ່ໜ້ອຍກວ່າຈິ່ງມີປະສິດທິພາບຫຼາຍກວ່າໃນແງ່ຂອງການນຳໃຊ້ໜ່ວຍຄວາມຈຳ.



4. Algorithm performance

4.3. Big O (Time Complexity in Big O)

- | Big-O notation ແມ່ນການສະແດງທາງຄະນິດສາດຂອງປະສິດທິພາບຂອງຂັ້ນຕອນວິທີ (algorithm).
- | ຄວາມໄວຂອງ algorithm ອາດຈະໄດ້ຮັບຜົນກະທົບຈາກປະສິດທິພາບຂອງຄອມພິວເຕີ. ດັ່ງນັ້ນ, ເຮົາຈຶ່ງຈຳເປັນຕ້ອງປຽບທຽບປະສິດທິພາບຂອງຄອມພິວເຕີ ດ້ວຍການສະແດງຂັ້ນຕອນການເຮັດວຽກດ້ວຍສັນຍາລັກທາງຄະນິດສາດ ແທນທີ່ຈະປະເມີນດ້ວຍເວລາຈິງທີ່ໃຊ້ໄປ.
- | ເມື່ອປຽບທຽບກັບຄວາມຊັບຊ້ອນທາງພື້ນທີ່ເຊິ່ງວັດແທກໄດ້ຍາກ, ຄວາມຊັບຊ້ອນຂອງເວລາມີປະໂຫຍດໃນການປະເມີນປະສິດທິພາບຂອງ algorithm.



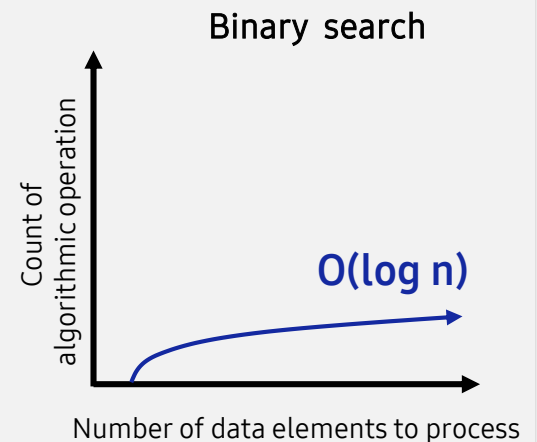
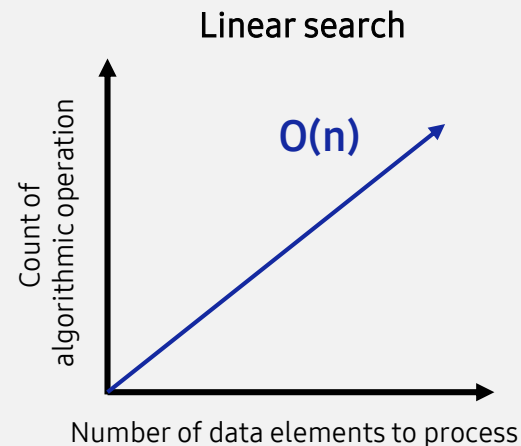
4. Algorithm performance

4.4. ຂຽນ ແລະ ປຽບທຽບ linear search ແລະ binary search ໃນ Big O.

- | ດັ່ງທີ່ຮູ້ແລ້ວວ່າການຄົ້ນຫາແບບ binary search ມີຂັ້ນຕອນການເຮັດວຽກໜ້ອຍກວ່າ ການຄົ້ນຫາແບບ linear search
- | ລອງປຽບທຽບການຄົ້ນຫາແບບ binary ແລະ ການຄົ້ນຫາແບບ linear ອີກເທື່ອໜຶ່ງດ້ວຍ Big O
- | algorithms ທີ່ຈະຮຽນ, ອັນທີ່ດີທີ່ສຸດແມ່ນ binary search ແລະ linear search, ສາມາດສະແດງຂັ້ນຕອນການຄຳນວນດ້ວຍ Big O ເພື່ອປຽບທຽບ ແລະ ວິເຄາະ algorithm ທີ່ມີປະສິດທິພາບ.
- | ຫາກທ່ານບໍ່ເຂົ້າໃຈວິທີສະແດງຈຳນວນການດຳເນີນງານໃນການຄົ້ນຫາແບບ binary ໃນ log, ສຶກສາດ້ວຍຕົນເອງນຳໃຊ້ link ລຸ່ມນີ້.

ພິຈາລະນາວ່າຈະໃຊ້ເວລາ 1 ມິນລິວິນາທີໃນການ
ກວດສອບຂໍ້ມູນໜຶ່ງລາຍການ

ຈຳນວນຂໍ້ມູນ	Linear search	Binary search
100	100 milliseconds	7 milliseconds
10.000	10 seconds	14 milliseconds
10^9	11 day	32 milliseconds

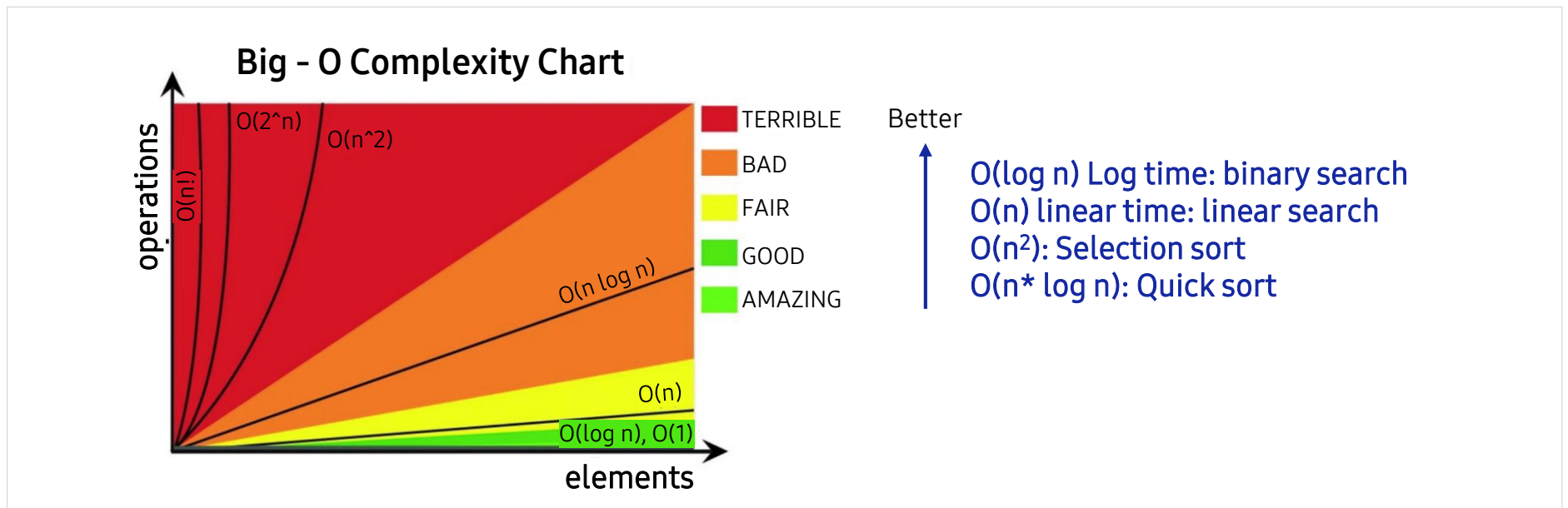


<https://www.khanacademy.org/math/algebra2/x2ec2f6f830c9fb89:logs/x2ec2f6f830c9fb89:log-intro/v/logarithms>

4. Algorithm performance

4.5. ປຽບທຽບ algorithms ອື່ນ ດ້ວຍ Big O.

- ໃນຫຼັກສູດນີ້, ຈະໄດ້ຮຽນຮູ້ປະເພດຕ່າງໆຂອງ algorithms. ຄ້າຍຄືກັນກັບ binary search ແລະ linear search, ຈຳນວນຄັ້ງຂອງການດຳເນີນການຂອງ algorithms ສາມາດຖືກຫມາຍດ້ວຍ Big O ຂຶ້ນກັບຈຳນວນຂໍ້ມູນ.
- ສາມາດລະບຸ algorithms ທີ່ມີປະສິດທິພາບສູງໄດ້ ດັ່ງສະແດງໃນຮູບລຸ່ມນີ້ດ້ວຍ Big O



Source : <https://medium.com/swlh/programming-pieces-big-o-notation-b42cfdc88132>



One more step

ລະຫັດເລກຖານສອງ (Binary code)

※ ເພື່ອເບິ່ງຄລິບວິດີໂອ, ເອົາ mouse ວາງເທິງ box ແລະ ປຸ່ມຫຼັ້ນປະກົດຂຶ້ນ. ຄລິກມັນເພື່ອເບິ່ງ.



<https://youtu.be/wgbV6DLVezo>

| Paper coding

ພະຍາຍາມເຂົ້າໃຈແນວຄວາມຄິດພື້ນຖານຢ່າງຄົບຖ້ວນກ່ອນທີ່ຈະກ້າວໄປສູ່ຂັ້ນຕອນຕໍ່ໄປ.

ການຂາດຄວາມເຂົ້າໃຈແນວຄວາມຄິດພື້ນຖານຈະເພີ່ມພາລະຂອງທ່ານໃນການຮຽນຮູ້ຫຼັກສູດນີ້, ເຊິ່ງອາດຈະເຮັດໃຫ້ທ່ານລົ້ມເຫລວໃນຫຼັກສູດ.

ຕອນນີ້ອາດຈະເປັນເລື່ອງຍາກ, ແຕ່ເພື່ອໃຫ້ສໍາເລັດຫຼັກສູດນີ້ ພວກເຮົາແນະນຳໃຫ້ທ່ານທຳຄວາມເຂົ້າໃຈແນວຄິດນີ້ຢ່າງທ່ອງແທ້ ແລະ ກ້າວໄປສູ່ຂັ້ນຕອນຕໍ່ໄປ



Pair Programming Practice

Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

I Prevent collaboration cheating

The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

I Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1. ສະແດງວິທີການຊອກຫາໂຮງແຮມໂດຍໃຊ້ຄໍາສັ່ງ pseudocode ຂ້າງລຸ່ມນີ້. ຂຽນທຸກຢ່າງໃນພາສາຊີວິດປະຈຳວັນຍົກເວັ້ນຄໍາສັ່ງຂ້າງລຸ່ມນີ້. ເອົາໃຈໃສ່ກັບການຫຍໍ້ໜ້າເມື່ອໃຊ້ຄໍາສັ່ງ.

“ໄປຊື້ທາງທິດຕາເວັນຕົກຕາມສີ່ແຍກນີ້. ເຈົ້າຈະເຫັນ 4 ຕຶກ.
ເມື່ອເຈົ້າເຫັນຫ້ອງການໄປສະນີ, ລຽວຂວາກົງກັນຂ້າມກັບຫ້ອງການໄປສະນີ.
ໄປຊື້ ແລະ ເມື່ອເຈົ້າເຫັນຮ້ານອາຫານອິຕາລີ, ຕຶກທີ່ສາມແມ່ນໂຮງແຮມທີ່ເຈົ້າກຳລັງຊອກຫາ.”

Commands to use

- Input : read, obtain, get
- output: print, show, display
- Calculation: compute, calculate, determine
- Iteration: for, while
- Decision-making according to conditions: if-then-else
- Conditional iteration: repeat-until
- Correct, wrong: true, false



Write the entire code and the expected output results in the note.

|

Let's code

1. ສາມໂຄງສ້າງການປະຕິບັດຂອງ Python ທີ່ສາມາດນຳໃຊ້ກັບຊີວິດປະຈຳວັນ

ສາມໂຄງສ້າງຂອງໂປຣແກຣມ (sequence, selection ແລະ iteration structure) ທີ່ໄດ້ຮຽນໃນ Unit 1 ມີປະໂຫຍດ ບໍ່ພຽງແຕ່ໃນການຂຽນ code ເທົ່ານັ້ນ, ແຕ່ຍັງຖືກນຳໃຊ້ໃນຊີວິດຈິງ.

ກໍລະນີຂອງຊີວິດປະຈຳວັນ

“ໄປຊື້ທາງທິດຕາເວັນຕົກຕາມສີ່ແຍກນີ້.
ເຈົ້າຈະເຫັນ 4 ຕົກ.

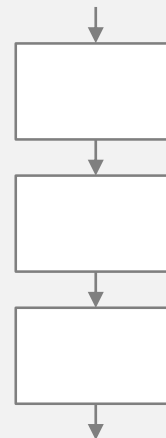
ເມື່ອເຈົ້າເຫັນຫ້ອງການໄປສະນີ, ລ້ຽວຂວາ
ຂ້າມຫ້ອງການໄປສະນີ.

ໄປຊື້ ແລະ ເມື່ອເຈົ້າເຫັນຮ້ານອາຫານອິຕາລີ
, ຕົກທີ່ສາມແມ່ນໂຮງແຮມທີ່ເຈົ້າກຳລັງ
ຊອກຫາ.”

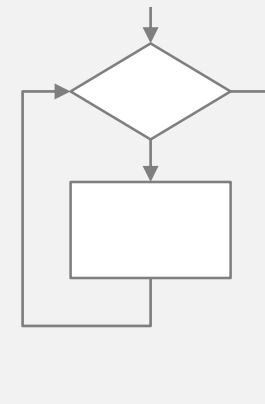
ນຳໃຊ້ໃນ
Algorithm

ສາມປະເພດຂອງໂຄງສ້າງການຂຽນໂປຣແກຣມ

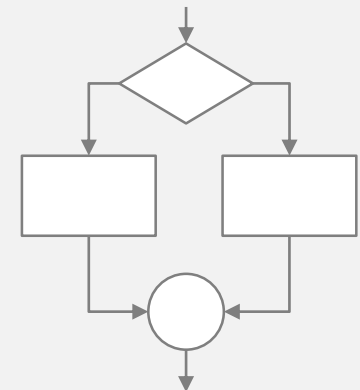
Sequence



Iteration

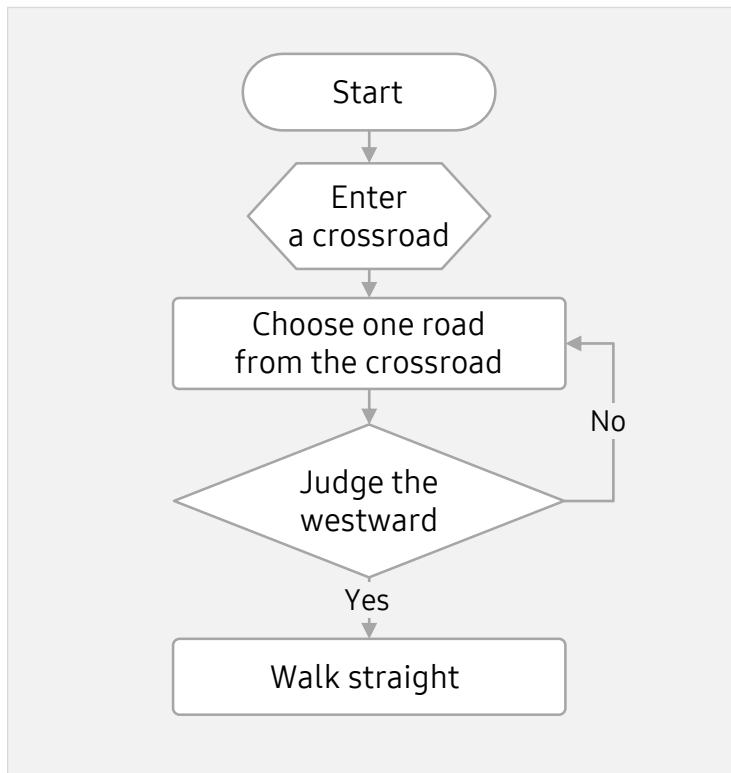


Selection



2. ແຕ້ມແຜນຜັງ (flowchart) ບອກທິດທາງ

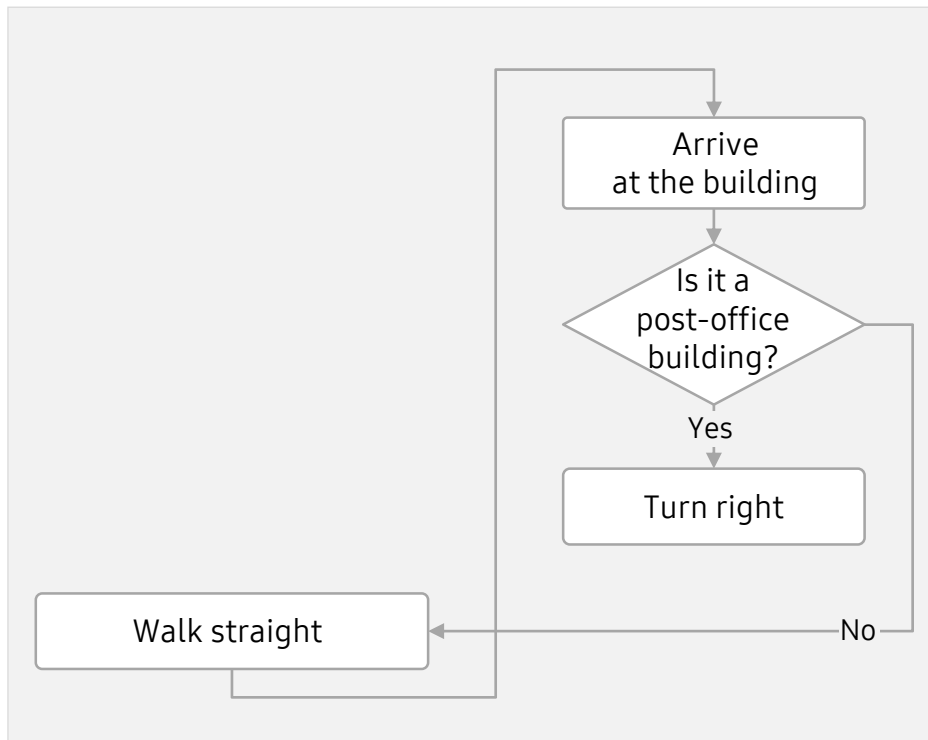
2.1. ເລີ່ມທິດທາງ ແລະ ຊອກຫາທິດທາງຕາເວັນຕົກ



- ▶ ພວກເຮົາຈະເລີ່ມຊອກຫາທາງໄປໂຮງແຮມ. ເລີ່ມຕົ້ນ, ເຮົາຄວນເຂົ້າໄປຢູ່ຈຸດທາງສີ່ແຍກກ່ອນ. [Sequence]
- ▶ ກ່ອນຈະໄປທາງທິດຕາເວັນຕົກທີ່ທາງແຍກ, ໃຫ້ກວດເບິ່ງເສັ້ນທາງເທື່ອລະເສັ້ນ ແລະ ພິຈາລະນາແຕ່ລະເສັ້ນວ່າ ຢູ່ທາງທິດຕາເວັນຕົກ ຫຼື ບໍ່. ຖ້າແມ່ນທາງທິດຕາເວັນຕົກ, ໃຫ້ຍ່າງຊື່. ຖ້າບໍ່ແມ່ນ, ໃຫ້ກວດເບິ່ງເສັ້ນທາງຕໍ່ໄປ. [Selection]

2. ແຕ້ມແຜນຜັງ (flowchart) ບອກທິດທາງ

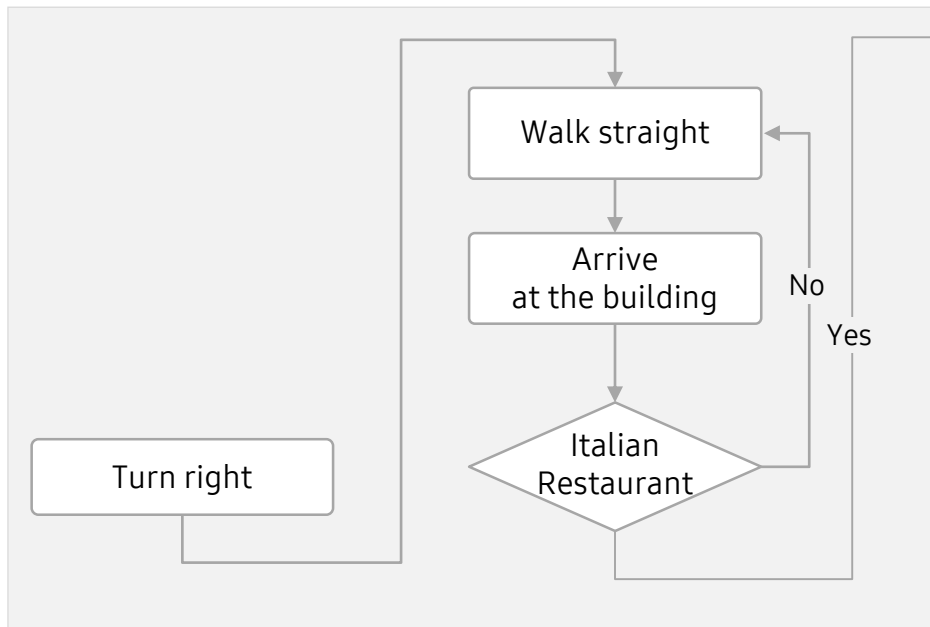
2.2. ຊອກຫາອາຄານຫ້ອງການໄປສະນີ



- ▶ ໄປທາງທິດຕາເວັນຕົກ, ແລ້ວຢ່າງຊື່ຕໍ່ໄປ. ຈະມາຮອດອາຄານທຳອິດ. [Sequence]
- ▶ ກວດສອບວ່າອາຄານທຳອິດແມ່ນຫ້ອງການໄປສະນີ ຫຼື ບໍ່. ຖ້າແມ່ນໃຫ້ລ້ຽວຂວາ ຖ້າບໍ່ແມ່ນ, ໃຫ້ໄປຊື່ ແລະ ກວດເບິ່ງອາຄານຕໍ່ໄປ. [Selection]

2. ແຕ້ມແຜນຜັງ (flowchart) ບອກທິດທາງ

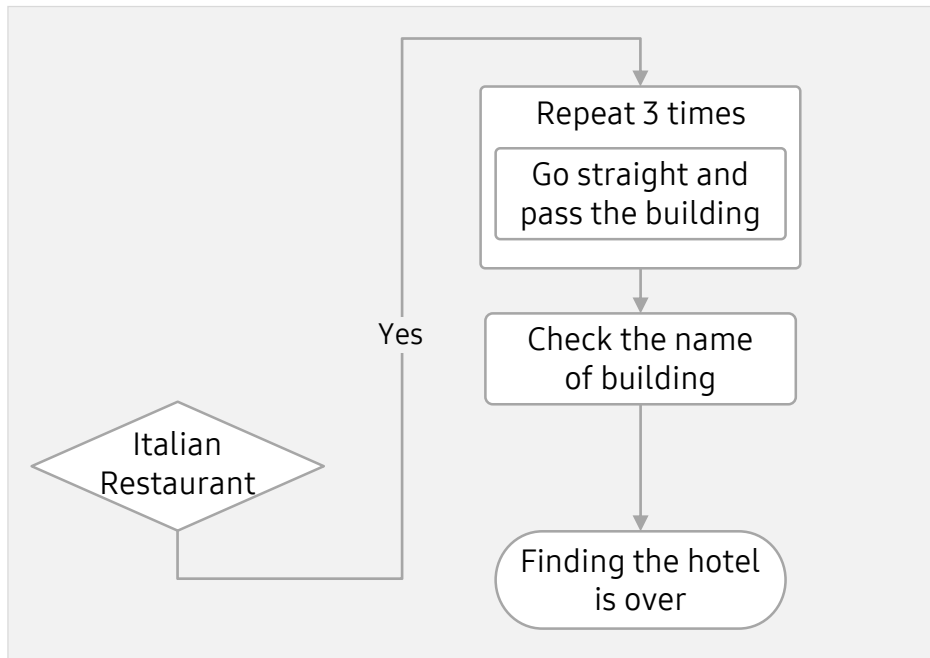
2.3. ຊອກຫາຮ້ານອາຫານອິຕາລີ



- ▶ ຫຼັງຈາກຜ່ານຕຶກໄປສະນີ, ໃຫ້ລ້ຽວຂວາ ແລະ ສືບຕໍ່ຍ່າງຊື່ ຈະມາຮອດອາຄານທຳອິດ. **Sequence**
- ▶ ກວດສອບວ່າອາຄານທຳອິດນີ້ແມ່ນຮ້ານອາຫານອິຕາລີບໍ່. ຖ້າແມ່ນ, ໃຫ້ເລີ່ມຊອກຫາໂຮງແຮມ. ຖ້າບໍ່ແມ່ນ, ໃຫ້ໄປຊື່ ແລະ ກວດເບິ່ງວ່າອາຄານຕໍ່ໄປແມ່ນຮ້ານອາຫານອິຕາລີບໍ່. **Selection**

2. ແຕ້ມແຜນຜັງ (flowchart) ບອກທິດທາງ

2.4. ຊອກຫາທີ່ພັກ ແລະ ເສັ້ນທາງສິ້ນສຸດ

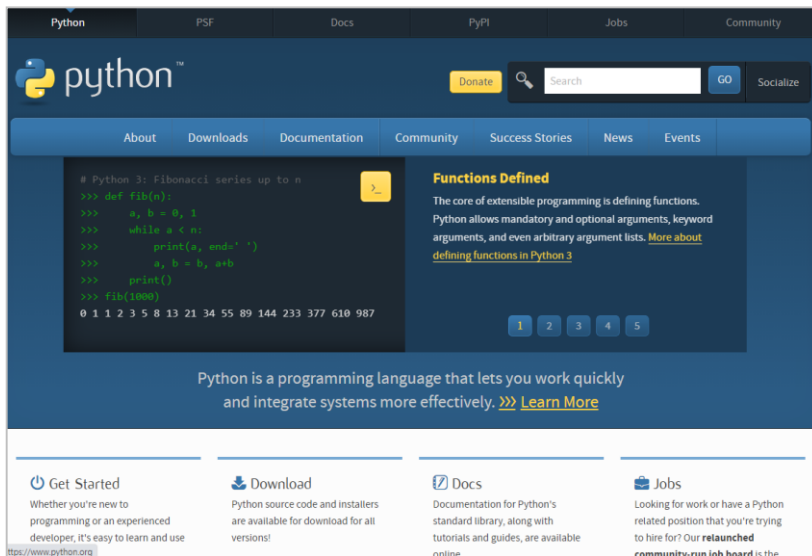


- ▶ ເມື່ອພົບຮ້ານອາຫານອິຕາລີ, ຈະເລີ່ມຊອກຫາໂຮງແຮມ. [Sequence]
- ▶ ອາຄານທີສາມແມ່ນອາຄານທີ່ກຳລັງຊອກຫາ. ດັ່ງນັ້ນ, ເຮັດຊ້ຳຂຶ້ນຕອນຂອງການໄປຊື້ແລະຜ່ານອາຄານສາມເທື່ອ. [Iteration]
- ▶ ເມື່ອມາຮອດຕຶກທີສາມ, ໃຫ້ກວດເບິ່ງຊື່ຂອງອາຄານເພື່ອເບິ່ງວ່າຖືກຕ້ອງ ຫຼື ບໍ່, ແລະ ຫຼັງຈາກນັ້ນສຳເລັດການຄົ້ນຫາ. [Sequence]

One more step

1. ການແນະນຳແຫຼ່ງອ້າງອີງການຮຽນຮູ້ Python

- | ມີແຫຼ່ງອ້າງອີງທີ່ດີໃນການຮຽນ Python ດ້ວຍຕົນເອງ ຫາກຕ້ອງການຮຽນເພີ່ມເຕີມກ່ຽວກັບ Python ຈຶ່ງເຂົ້າໄປທີ່ເວັບໄຊຕາມລິ້ງຂ້າງລຸ່ມນີ້.
- | ມູນນິທິຊອບແວ Python (Python Software Foundation)



- ▶ ມູນນິທິທີ່ບໍ່ຫວັງຜົນກຳໄລທີ່ຮັບຜິດຊອບສຳລັບການພັດທະນາ ແລະ ການແຈກຢາຍ Python.
- ▶ ສາມາດດາວໂຫລດເວີຊັນຫຼ້າສຸດຂອງ Python ໄດ້ ບໍ່ມີຄ່າໃຊ້ຈ່າຍ.
- ▶ ສະໜອງ Tutorials ແລະ ເອກະສານສຳລັບການຮຽນຮູ້ Python.
- ▶ ແນະນຳຂ່າວສານ ແລະ ກິດຈະກຳຫຼ້າສຸດ.
- ▶ ໃຫ້ບາງຕົວຢ່າງທີ່ປະສົບຜົນສຳເລັດຂອງໂຄງການ Python ທີ່ຍິ່ງໃຫຍ່



One more step

1. ການແນະນຳແຫຼ່ງອ້າງອີງການຮຽນຮູ້ Python

■ ມູນນິທິຊອບແວ Python (Python Software Foundation)

The screenshot shows the Python Tutorial page from the Python Software Foundation documentation. The page title is "The Python Tutorial". The left sidebar contains links for "Previous topic: Changelog", "Next topic: 1. Whetting Your Appetite", and "This Page: Report a Bug, Show Source". The main content area starts with a paragraph: "Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms." This is followed by a paragraph about the Python interpreter and standard library, and another paragraph about the Python interpreter's extensibility. The page concludes with a paragraph about the tutorial's purpose and a link to the Python Standard Library.

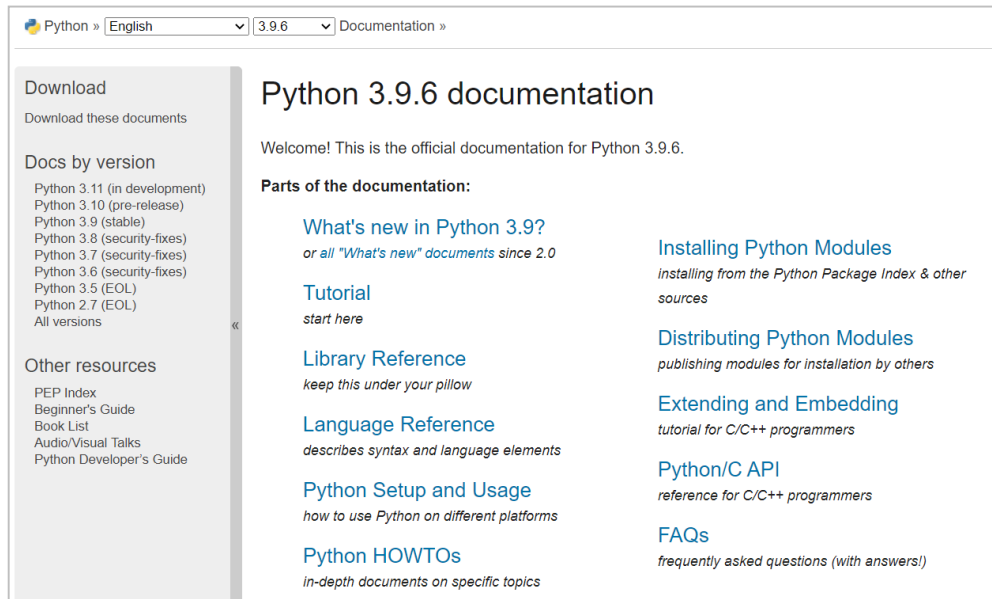
<https://docs.python.org/3/tutorial/index.html>

► ສາມາດຮຽນຮູ້ພື້ນຖານຂອງ Python ໃນຫນ້າ Python Tutorial.

One more step

1. ການແນະນຳແຫຼ່ງອ້າງອີງການຮຽນຮູ້ Python

■ ມູນນິທິຊອບແວ Python (Python Software Foundation)



The screenshot shows the Python 3.9.6 documentation page. At the top, there are dropdown menus for 'Python »', 'English', '3.9.6', and 'Documentation »'. The main heading is 'Python 3.9.6 documentation'. Below it, a welcome message states: 'Welcome! This is the official documentation for Python 3.9.6.' The page is organized into sections: 'Download' (with a link to 'Download these documents'), 'Docs by version' (listing versions from Python 3.11 down to 2.7, and 'All versions'), and 'Other resources' (listing 'PEP Index', 'Beginner's Guide', 'Book List', 'Audio/Visual Talks', and 'Python Developer's Guide'). The 'Parts of the documentation:' section lists various topics with brief descriptions: 'What's new in Python 3.9?' (or all 'What's new' documents since 2.0), 'Installing Python Modules' (installing from the Python Package Index & other sources), 'Distributing Python Modules' (publishing modules for installation by others), 'Extending and Embedding' (tutorial for C/C++ programmers), 'Python/C API' (reference for C/C++ programmers), 'FAQs' (frequently asked questions (with answers!)), 'Tutorial' (start here), 'Library Reference' (keep this under your pillow), 'Language Reference' (describes syntax and language elements), 'Python Setup and Usage' (how to use Python on different platforms), and 'Python HOWTOs' (in-depth documents on specific topics).

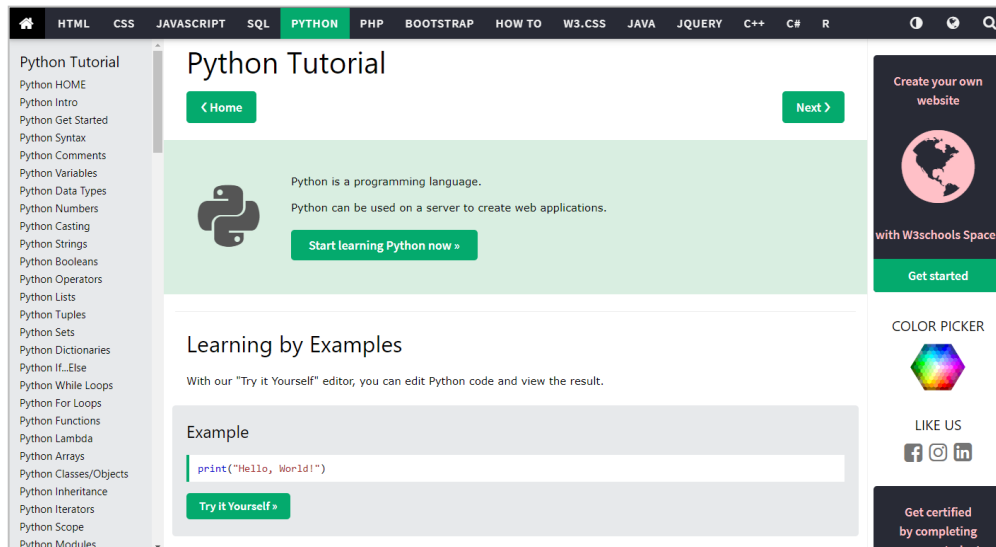
<https://docs.python.org/3/>

- ▶ ຫນ້າເອກະສານ Python (Python documentation) ສະໜອງເອກະສານລາຍລະອຽດກ່ຽວກັບ Python.

One more step

1. ການແນະນຳແຫຼ່ງອ້າງອີງການຮຽນຮູ້ Python

W3Schools



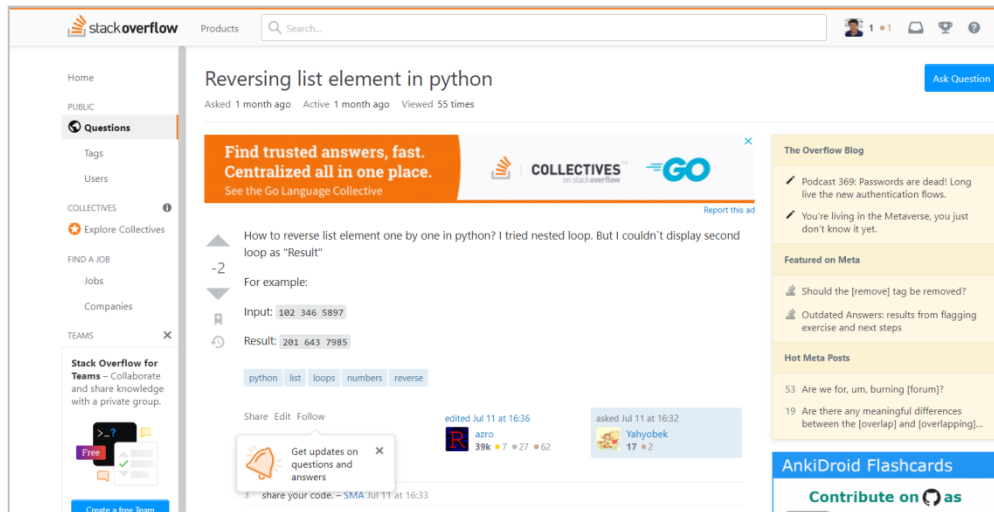
<https://www.w3schools.com>

- ▶ ເຮົາຄົ້ນຫາສິ່ງທີ່ມີປະໂຫຍດໃນການເອົາ Python ໄປໃຊ້ຢ່າງກວ້າງຂວາງຫຼາຍຂຶ້ນໄດ້ແນວໃດ?
- ▶ ທີ່ W3Schools, ທ່ານສາມາດຮຽນຮູ້ພາສາການຂຽນໂປຣແກຣມຕ່າງໆດ້ວຍ Python.
- ▶ ໂດຍສະເພາະ, ທ່ານສາມາດ run ຕົວຢ່າງ code Python ໃນເວັບໄຊໄດ້ເລີຍ.

One more step

1. ການແນະນຳແຫຼ່ງອ້າງອີງການຮຽນຮູ້ Python

Stack Overflow



<https://stackoverflow.com/>

- ▶ ໃນຂະນະທີ່ຂຽນໂປຣແກຣມ, ເຮົາຈະພົບກັບຄວາມຜິດພາດ ແລະ ບັນຫາຫຼາຍຢ່າງທີ່ແກ້ໄຂຍາກ. ຈະແກ້ໄຂບັນຫາເຫຼົ່ານີ້ໄດ້ແນວໃດ?
- ▶ Stack Overflow ແມ່ນເວັບໄຊແບ່ງປັນຂໍ້ມູນທີ່ນິຍົມທີ່ສຸດໃນໂລກສໍາລັບນັກພັດທະນາ.
- ▶ ຄື້ນຫາ “ຢ້ອນກັບເບິ່ງ (reverse) ອົງປະກອບລາຍການໃນ python" ຢູ່ໃນເວັບໄຊນີ້. ຈະມີຫຼາຍຄໍາຖາມ ແລະ ການສົນທະນາທີ່ກ່ຽວຂ້ອງກັບລາຍການທີ່ເຮົາຈະຮຽນຮູ້ໃນພາຍຫຼັງ.

| Pair programming



Pair Programming Practice

Guideline, mechanisms & contingency plan

Preparing pair programming involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns “driving the mouse.” Effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, it is important to make it clear that the active student will not be punished because the pairing did not work well.

Pairing similar, not necessarily equal, abilities as partners

Pair programming can be effective when students of similar, though not necessarily equal, abilities are paired as partners. Pairing mismatched students often can lead to unbalanced participation. Teachers must emphasize that pair programming is not a “divide-and-conquer” strategy, but rather a true collaborative effort in every endeavor for the entire project. Teachers should avoid pairing very weak students with very strong students.

Motivate students by offering extra incentives

Offering extra incentives can help motivate students to pair, especially with advanced students. Some teachers have found it helpful to require students to pair for only one or two assignments.



Pair Programming Practice

I Prevent collaboration cheating

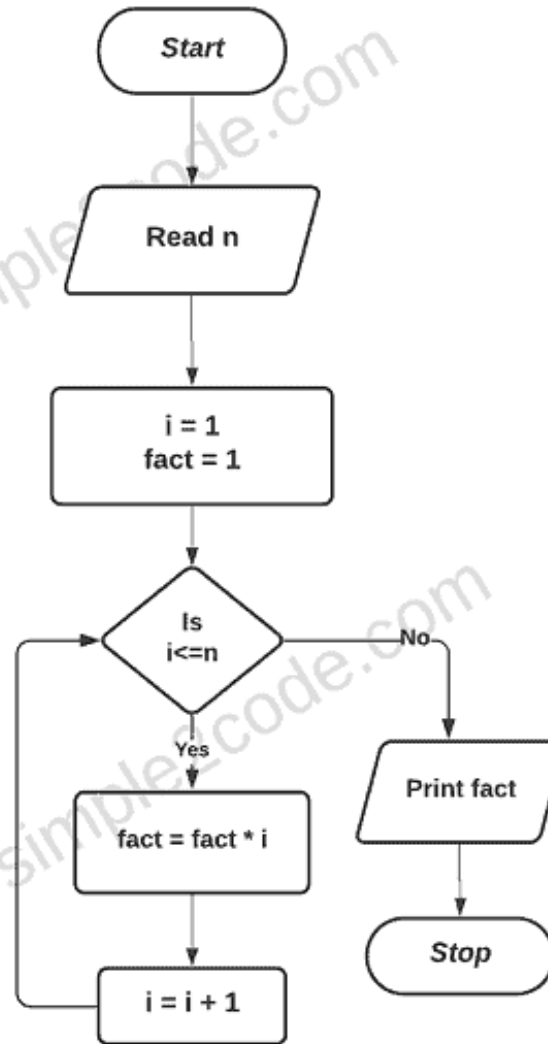
The challenge for the teacher is to find ways to assess individual outcomes, while leveraging the benefits of collaboration. How do you know whether a student learned or cheated? Experts recommend revisiting course design and assessment, as well as explicitly and concretely discussing with the students on behaviors that will be interpreted as cheating. Experts encourage teachers to make assignments meaningful to students and to explain the value of what students will learn by completing them.

I Collaborative learning environment

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

Q1. ຂຽນ flowchart ຂອງ factorizes 18, 39, 63, 126, 792

Flowchart for Factorial Number



<https://simple2code.com/c-tutorial/algorithm-and-flowchart-to-find-factorial-of-a-number/>