

Begin Modern Programming

with

C
O
D
I
N
G
A

Pyi Soe

အခန်း ၁

စက်ရုပ်ကားရဲလိုဖြင့် ပရီဂရမ်းမင်းမိတ်ဆက်

ကွန်ပျိုတာတွေဟာ သက်မဲ့ စက်ပစ္စည်းတွေပါပဲ။ ကားတို့ လေယာဉ်တို့နဲ့ မတူတာက ကွန်ပျိုတာတွေဟာ စက်ချဉ်းသက်သက် ဘာအစွမ်းမှ မယ်မယ်ရရ မရှိဘူး။ ဒါပေမဲ့ ဆောင်ရွက်လိုတဲ့ ကိစ္စအဝေဝအတွက် ပရီဂရမ်းမျိုးမျိုး ထည့်ပေးလိုက်တဲ့ အခါမာ သူ့အစွမ်းက အတိုင်းအဆမဲ့ပဲ။ နေရာမျိုးစံ၊ နယ်ပါယ်မျိုးစံ မှာ အကူးအညီပေးနိုင်တဲ့ စွယ်စုံသုံး ပစ္စည်းတစ်ခုဖြစ်သွားတယ်။ ဂိတ်သံစဉ်တွေကို ဖွင့်ပေးနိုင်သလို အသံလည်းသွင်းပေးနိုင်တယ်။ ရုပ်ရှင်တည်းဖြတ် လုပ်ချင်တာလား။ ပြဿနာမရှိဘူး၊ ကူညီပေးနိုင်တယ်။ နျှော လီးယား ပါတ်ပေါင်းဖို့တွေကို ဖီမံနိုင်သလို မောင်းသူမဲ့ အုံပျုံတွေကိုလည်း ပဲထိန်းပေးနိုင်တယ်။

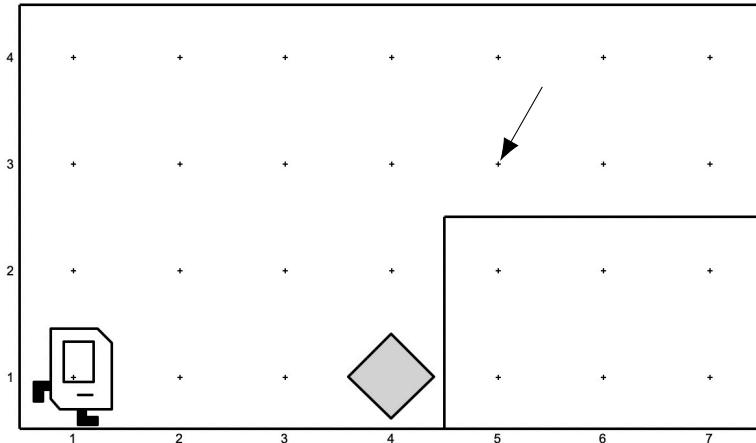
ကျွန်တော်တို့တွေ နိစ္စဓာဝ အသုံးပြုနေကြတဲ့ ကား၊ စမတ်ဖုန်း၊ လက်ပါတ်နာရီ၊ မိုက်ခရီးဝေဖွံ့ဖိုးဖိုး အဝတ်လျှော့စက် စတဲ့ စက်ပစ္စည်း အမျိုးမျိုးဟာလည်း ကွန်ပျိုတာတွေနဲ့ မကင်းပြန်ပါဘူး။ “ကွန်ပျိုတာနည်းပညာ အကူးအညီမပါတဲ့ အတိုင်းဆက်သံစုံထွင်မှုဆိုတာ မရှိဘူး” လို့ ဆိုနိုင်ပါတယ်။

တစ်ချက်တစ်ချက် ရိုက်ခတ်လိုက်တဲ့ ကွန်ပျိုတာနည်းပညာ လိုင်းလုံးကြီးတွေဟာ ကမ္မာတစ်စုံမှုးလုံး ပုံစံပြောင်းသွားလောက်အောင် အဟုန်ပြင်းထန်လာတယ်။ ဘီလီယံချွဲတဲ့ လူတွေ ဆီရှယ်မီဒီယာတွေပေါ်က နေ ရုပ်သံတွေနဲ့ ချိတ်ဆက်ပြောဆိုတာတွေကို ရစေတာဟာလည်း ကွန်ပျိုတာစနစ်တွေပါပဲ။ Artificial Intelligence (AI) နည်းပညာကြောင့် သက်ရှိတွေမှာပဲတွေ့ရတဲ့ ညာတ်ရည်မျိုးကို ကွန်ပျိုတာတွေမှာလည်း တွေ့လာရပါပြီ။ သံချွာပွဲတွေ ဖြေရှင်းခြင်း၊ စစ်တုရင်ထိုးခြင်း စတဲ့ ကိစ္စမျိုးတွေအပြင် ပန်းချီဆွဲခြင်း၊ ကဗျာရေးစပ်ခြင်း၊ သီချင်းရေးဖွဲ့ခြင်း ကဲ့သို့ အနုပညာဖန်တီးမှုတွေကိုပါ AI က လုပ်ဆောင်ပေးနိုင်ပါတယ်။ နှစ်ဆယ်တစ်ရာစုံ အထူးမြှားဆုံး AI နည်းပညာလိုင်းဟာ အရှိန်အဟုန်ပြင်းပြင်း ရိုတ်ခတ်ဖို့ အားယူစ ပြုနေပါပြီ။

‘ကွန်ပျိုတာ’ လိုပြောတဲ့ အခါ စက်ပစ္စည်းသက်သက် မဟုတ်ဘဲ ကွန်ပျိုတာမှုတ်ညာတ်ထဲက ပရီဂရမ်တွေလည်း ပါဝင်တယ်ဆိုတာ သတိချုပ်ရပါမယ်။ ကွန်ပျိုတာတွေ တစ်စုံတစ်ရာ စွမ်းဆောင်နိုင်စေတဲ့ ပရီဂရမ်တွေ ရေးတဲ့ အလုပ်ကို ပရီဂရမ်းမင်း(Programming) လို့ ခေါ်တယ်။

၁.၁ စက်ရုပ် ကားရဲလ်

ပရီဂရမ်းမင်းမင်းမျိုးတာ ဘယ်လိုမျိုးလဲ သဘောပေါက်အောင် စာတွေတစ်သီးပြီးရေး ရှင်းပြတာထက် ပရီဂရမ်လေးတွေ လက်တွေ့ ရေးကြည်လိုက်တာ ပိုပြီးထိရောက်ပါတယ်။ ဒါကြောင့် စက်ရုပ်ကားရဲလ်ကို ပရီဂရမ်လေးတွေရေးပြီး အလုပ်တွေခုံးကြည့်ကြမယ်။ ပုံ (၁.၁) မှာ တွေ့ရတာက ကားရဲလ် ရောက်ရှိနေတဲ့ နဗုံနာကမ္မာတစ်ခုပါ။ မီးခါးရောင် မှုန်ကူကုံးပုံလေးကို ဘိပါ (beeper) လို့ ခေါ်တယ်။ အဲဒီဘိပါကို မြားပြထားတဲ့ နေရာကို ရွှေခိုင်းချင်တယ်။ မျှေားမည်းအထူးတွေက နံရံတွေပါ။ ကားရဲလ်ကို ကိစ္စတစ်ခု ဆောင်ရွက်စေ



ပုံ ၁.၁ ခက်ရှုပ်လေး ကားရဲ့

ချင်တဲ့အခါ အခြေခံ ကားရဲ့လိုက် တွေ့ကို အသုံးပြုရပါတယ်။ ကွန်မန်းတွေ့ကို နှုတ်နှုံးပြောပြီး ခိုင်းရတာ မဟုတ်ဘဲ ပရိုကရမ်းပြီး ခိုင်းရတာပါ။ ကားရဲ့နားလည်တဲ့ ကွန်မန်းတွေ့ကို ကြည့်ကြရအောင်။

ကားရဲ့လိုက်မန်းများ

မဖြစ်မနေ သိထားရမဲ့ အခြေခံ ကားရဲ့လိုက်မန်း လေးခုပဲ ရှိတယ်။ move, turn_left, put_beeper နဲ့ pick_beeper တို့ဖြစ်တယ်။ အခြား ကားရဲ့လိုက်မန်း တွေ့လည်း ရှိပါသေးတယ်။ ဒါပေမဲ့ ကားရဲ့လိုက်များမင်း စလေ့လာဖို့ ဒီလေးခုနဲ့ပဲ လုံလောက်ပါပြီ။

move ကွန်မန်းက ကားရဲ့လိုက် ရှေ့တွေ့ကွန်နာကို ရွှေ့ခိုင်းတာ။ ကားရဲ့လိုက်မှာ တစ်ခုနဲ့တစ်ခု အကွာအဝေးတူ ခြားထားတဲ့ အတန်းလိုက် အတန်းလိုက် အစက်ကလေးတွေ့ဟာ ကွန်နာ (corner) တွေ ဖြစ်တယ်။ ကဲ့သို့ မျဉ်းမည်းအထူ နံရုံတွေ့နဲ့ ထောင့်မှန်စတုဂံပဲ ပါတ်လည် ဘောင်ခတ်ထားတယ်။ ကွန်နာတွေ့ကြားမှာလည်း နံရုံတွေ့ရှိနိုင်တယ်။ နူးနာကဲ့မှာ ဘေးတိုက် နံပါတ်စဉ် ၄ နဲ့ ၅ ကြား ထောင်လိုက် နံရုံတွေ့ခဲ့ အထက်အောက် နံပါတ်စဉ် ၂ နဲ့ ၃ ကြား အလျေားလိုက် နံရုံတွေ့ခဲ့ကို တွေ့ရှုပါမယ်။ ကွန်နာရှေ့မှာ နံရုံကေနေရင် ကားရဲ့လိုက် ပေါ်ခဲ့လို့မရပါဘူး။

put_beeper က ကားရဲ့လိုက် လက်ရှိ ရှိနေတဲ့ ကွန်နာမှာ ‘ဘိပါတစ်ခုချု’ ထားခိုင်းတာ၊ pick_beeper က ရပ်နေတဲ့ ကွန်နာမှာ ‘ဘိပါတစ်ခုကောက်’ ခိုင်းတာပါ။ ကွန်နာမှာ ဘိပါရှိနေမှ ကောက်ခိုင်းလို့ရမှုပါ။ မရှုရင် ကောက်ခိုင်းလို့ မရှုဘူး။ ဘိပါချိခိုင်းရင်လည်း ကားရဲ့လိုက် ဘိပါရှိမှ ချိခိုင်းလို့ရတယ်။ ကားရဲ့လိုက် ဘိပါတွေ့ လို့သလောက် ဖြည့်ပေးထားတယ်လို့ ယူဆပါ။ turn_left က ‘ဘယ်လှည့်’ ခိုင်းတာ။

ဘိပါကို ဘယ်လိုပြောခိုင်းမလဲ

ပုံ (၁.၁) အနေအထားကနေ ရေ့ကို သုံးနေရာရွှေ့ ဘိပါကောက်၊ ဘယ်ဘက်လှည့်၊ အပေါ် နှစ်နေရာရွှေ့ ညာဘက်လှည့်၊ ရှေ့တွေ့နေရာထပ်ရွှေ့ပြီး ဘိပါချုထားခိုင်းလိုက်ရင် အလုပ်ပြီးသွားပါပြီ။

ကားရဲ့လိုက် ညာဘက်လှည့်ခိုင်းဖို့ turn_right ကွန်မန်း မရှိဘူး။ ဒါပေမဲ့ ဘယ်သုံးခါလှည့်တာဟာ ညာဘက်လှည့်တာနဲ့ တူတူပါပဲ။ ဒါကြောင့် ညာဘက်ချင်တဲ့အခါ ဘယ်သုံးခါလှည့်ခိုင်းလို့ရတယ်။

၁.၂ ကားရဲလ် ပရိုဂရမ်

ပရိုဂရမ် ရေးတယ်ဆိုတာ ကွန်ပျိုတာကို ကိစ္စတစ်ခုခဲ့ ဆောက်ရွက်ပေးဖို့ ခိုင်းစေတဲ့ ညွှန်ကြားချက်တွေ ရေးတာပါပဲ။ ဒီလို ညွှန်ကြားချက်တွေကို ပရိုဂရမ်ကုဒ် (program code) လို ခေါ်တယ်။ ပရိုဂရမ်ကုဒ် တွေကို ကွန်ပျိုတာနားလည်တဲ့ programming language တစ်ခုခဲ့ ရေးရတယ်။ ဒီစာအုပ်မှာ အသုံးပါမဲ့ programming language ကတော့ Python ပါ။ Programming language တစ်မျိုးပဲ ရှိတာ မဟုတ်ပါဘူး။ ရာနဲ့ချို့ပြီး ရှိတာပါ။ လူဘာသာစကားတွေ အမျိုးမျိုးရှိသလိုပဲပေါ့။ Python ဟာ ဒီလို ရာနဲ့ချို့တဲ့တော့ လက်ရှိအသုံးအများဆုံး ထိပ်ဆုံးဆယ်ခု ထဲမှာ ပါဝင်တယ်။ Python နဲ့ ဘိပါန္တာခိုင်းတဲ့ ပရိုဂရမ်ကို လေ့လာကြည့်ရအောင်။ ကားရဲလ်နဲ့ ပထမဆုံး မိတ်ဆက်ပေးတဲ့ ပရိုဂရမ်မို့လို ဒီပရိုဂရမ် နံမည်ကို 'Meet Karel' လို ခေါ်ပါမယ်။

```
# File: meet_karel.py
# About: This is
from stanfordkarel import *

def main():
    """Karel code goes here!"""
    move()
    move()
    move()
    pick_beeper()
    turn_left()
    move()
    move()

    turn_left()
    turn_left()
    turn_left()

    move()
    put_beeper()
# End of main

if __name__ == "__main__":
    run_karel_program("meet_karel")
```

ဒါဟာ 'Meet Karel' ပရိုဂရမ်အတွက် Python နဲ့ရေးထားတဲ့ ပရိုဂရမ်ကုဒ် တွေဖြစ်ပါတယ်။ 'Python ကုဒ်' လို အတိုကျိုးပဲ ပြောတာများတယ်။ Python 'စာ/စကား' မတတ်ရင် ဒီ 'Python ကုဒ်' တွေကိုလည်း နားလည်မှာ မဟုတ်ဘူး။ ဒီတော့ Python 'စာ/စကား' အခြေခံက စပြီး လေ့လာဖို့လိုပါမယ်။

ကွန်းမန် (Comment)

ပထမဆုံး # သင်္ကာနဲ့ စတဲ့ စာကြောင်းတွေက ကွန်းမန်တွေပါ။ ကွန်းမန်တွေက ကွန်ပျိုတာ ဆောင်ရွက်ပေးရမဲ့ ညွှန်ကြားချက်တွေ မဟုတ်ဘူး။ ပရိုဂရမ်ကုဒ်နဲ့ ပါတ်သက်ပြီး ကုဒ် ဖတ်ရှုသူ အတွက် မှတ်ချက်ရေးတာ သို့မဟုတ် ရှင်းပြထားတာပါ။ တနည်းအားဖြင့် ဖတ်ရှုသူ (လူ) ပရိုဂရမ်မှာအတွက် ရည်ရွယ်တာ။ ကွန်ပျိုတာ (စက်) အတွက် ရည်ရွယ်တာ မဟုတ်ဘူး။ ကွန်ပျိုတာက ကုဒ်ထဲက ကွန်းမန်တွေ အားလုံးကို

လစ်လၢြရှုမှုပါ။ ဒါပေမဲ့ ပရိုကရမ်ကုဒ်ကို ဖတ်တဲ့လူ နားလည်ဖို့ အထောက်အကူးဖြစ်စေတဲ့အတွက် ကုန်းမန်ရေးတာကို ပေါ့ပေါ့တန်တန် အရေးမပါသလို သဘောထားလို့ မရပါဘူး။ မိမိရေးတဲ့ ကုဒ်ကို ရှင်းပြီးလိုအပ်ရင် ကုန်းမန်ရေးရပါမယ်။ ရေးသင့်တဲ့ နေရာတွေကိုလည်း မကြာခင်တွေရမှုပါ။

import စတိတ်မန္တ

```
from stanfordkarel import *
```

ကတေသာ အင်ပို့စတိတ်မန် ဖြစ်ပါတယ်။ “stanfordkarel လိုက်ဘရီမှ အာလုံးကို ထည့်သွင်းပေးပါ” လိုတောင်းဆိုတဲ့ အဓိပ္ပာယ်။ * သက်တာကို ‘အားလုံး’ လို့ ယူဆပါ။ stanfordkarel လိုက်ဘရီမှ ကားရဲ့လုပ်ပရိုကရမ်အတွက် လိုအပ်တာအားလုံး ပါဝင်တယ်။ ဒီလိုက်ဘရီကို အင်ပို့လုပ်ထားမှ ကားရဲ့လုပ်ကွန်းများတွေ သုံးလို့ရမှုပါ။ သီးခြား ကားရဲ့လုပ်ပရိုကရမ် တစ်ခုစီတိုင်းအတွက် အင်ပို့လုပ်ရမှာ ဖြစ်တယ်။

လိုက်ဘရီ

လိုက်ဘရီ (*library*) ဆိုတာ ပညာရပ်နယ်ပယ် တစ်ခုအတွက် ရည်ရွယ်ရေးထားတဲ့ ပရိုကရမ်ကုဒ်တွေပါပဲ။ သချုပ်အတွက်အချက် လိုက်ဘရီ ဂိမ်းရေးစွဲ လိုက်ဘရီ၊ 2D/3D ရေပ်စစ်ဆေးစွဲ လိုက်ဘရီ၊ အေအာင်အတွက် လိုက်ဘရီ စသည်ဖြင့် နယ်ပယ်အသီးသီး၊ ကိစ္စရပ်အဖို့ဖို့အတွက် သက်ဆိုင်ရာ ကျမ်းကျင်ပညာရှင်တွေ ထုတ်လုပ်ဖြန့်ချီပေးထားတဲ့ လိုက်ဘရီတွေရှိတယ်။ Matrix အော်ပရေးရှင်းတွေအတွက် numpy၊ ဂရပ်စွဲမယ်ဆိုရင် matplotlib စတဲ့ လိုက်ဘရီတွေကို အင်ပို့လုပ် အသုံးပြနိုင်ပါတယ်။ မူထရာစ် A ကို B နဲ့ မြောက်ရင် ဒီလိုပါ

```
from numpy import *

A = array([[1, 1, 2, 2],
           [2, 2, 1, 1],
           [2, 2, 1, 1]])
B = array([[2, 2],
           [2, 2],
           [1, 1],
           [2, 2]])

result = matmul(A, B)

print(result)
```

ရလဒ် အခုလိုထွေက်ပါမယ်။

```
[[10 10]
 [11 11]
 [11 11]]
```

ဒါကတေသာ ဘားချွတ် အတွက် numpy နဲ့ matplotlib လိုက်ဘရီ သုံးထားတာပါ။

```
from matplotlib.pyplot import *
from numpy import *

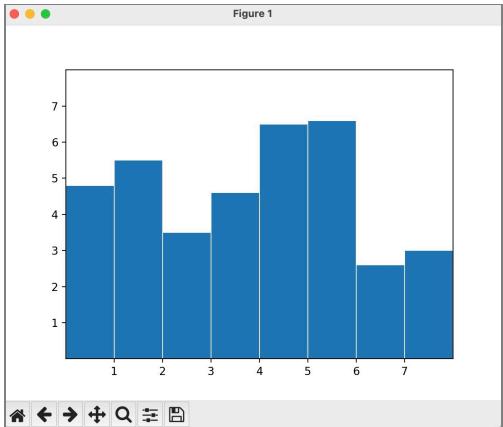
# make data:
x = 0.5 + arange(8)
y = [4.8, 5.5, 3.5, 4.6, 6.5, 6.6, 2.6, 3.0]
```

```
# plot
fig, ax = subplots()
bar(x, y, width=1, edgecolor="white", linewidth=0.7)

ax.set(xlim=(0, 8), xticks=arange(1, 8),
       ylim=(0, 8), yticks=arange(1, 8))

show()
```

ဘားချုတ်ကို ဒီလို ထုတ်ပေးပါတယ်။



ထိုး၏

လိုက်ဘရီတွေဟာ ပရိုကရမ်တွေ တည်ဆောက်ရာမှာ အင်မတန်မှ အရေးပါတယ်။ ဖော်ပြထားတဲ့ မေ့ထရစ် နဲ့ ဘားချုတ် ကုဒ်တွေကို (အခုတော့) နားလည်မှာ မဟုတ်သေးဘူးပေါ့။ ဒါပေမဲ့ သက်ဆိုင်ရာ လိုက် ဘရီတွေနဲ့ ဒီလိုကိစ္စတွေကို သိပ်မခက်ခဲဘဲ လုပ်လိုရနိုင်တယ်ဆိုတာ မြင်မယ် ထင်ပါတယ်။ လိုက်ဘရီတွေ သာမရှိရင် ပရိုကရမ်တွေကို အခုထက် အဆပေါင်းများစွာ အချိန်ပေးပြီး ရှုပ်ရှုပ်ထွေးထွေး ခက်ခက်ခဲ့လဲ တည်ဆောက်ကြရမှာပါ။

တံ့က်၏ စတိတ်မန်နှင့် ဆင်းတက်၏

ဂျို့နှင့် ပြင်သစ်စာ စတဲ့ လူဘာသာစကားတွေဟာ စကားလုံးတွေ ဝါကျတွေနဲ့ ဖွဲ့စည်းထားသလို ပရီ ဂရမ်ကုဒ်တွေဟာလည်း စကားလုံးတွေ ဝါကျတွေနဲ့ ဖွဲ့စည်းထားတာပါပဲ။ Programming language တွေမှာ စကားလုံးတွေကို တံ့က် (token) လိုခေါ်ပြီး ဝါကျတွေကိုတော့ စတိတ်မန် (statement) လို ခေါ်ပါတယ်။ ဝါကျတွေကို စကားလုံးတွေနဲ့ ဖွဲ့စည်းထားသလို စတိတ်မန်တွေကတော့ တံ့က်တွေနဲ့ ဖွဲ့စည်းထားတာပါ။ စတိတ်မန် ပုံစံတစ်မျိုးကို တွေ့ခဲ့ပြီးပါပြီ။ အဲဒါကတော့ ရေ့စာမျက်နှာက အင်ပိုစတိတ်မန်ပဲ ဖြစ်ပါတယ်။

လူဘာသာစကားတွေမှာ ဖွဲ့စည်းတည်ဆောက်ပုံ စထရက်ချာရှိသလို programming language တွေမှာလည်း စထရက်ချာရှိဖို့ လိုအပ်တာပေါ့။ ဖွဲ့စည်းပုံ စထရက်ချာ မှန်/မမှန်ကို သဒ္ဓါစည်းမျဉ်တွေနဲ့ ထိန်းကွပ်ထားတာပါ။ ပရိုကရမ်ကုဒ် စထရက်ချာ မှန်/မမှန် ထိန်းကွပ်ပေးတဲ့ သဒ္ဓါစည်းမျဉ်တွေကိုတော့ ဆင်းတက် (syntax) လိုခေါ်တယ်။

မြန်မာလိုရေးရင် မြန်မာသဒ္ဓါကို လိုက်နာရသလို Python နဲ့ ရေးရင် Python ဆင်းတက်စိုးကို

လိုက်နာရမှာပေါ့။ မြန်မာသွေ့မှားရင် ဖတ်တဲ့သူက သည်းခံနားလည် ပေးပေမဲ့ ဆင်းတက်စုံမှားရင်တော့ Python က လုံးဝ လက်ခံမှာ မဟုတ်ပါဘူး။ ဆင်းတက်စုံ စဉ်းကမ်းတွေဟာ ပိုပြီး တင်းကျပ်တယ်။ လွှဲချော်လို့ မရဘူး။ ဆင်းတက်စုံမှားနေတဲ့ ပရိုကရမ်ကို Python က run ခွင့် ပြုမှာမဟုတ်ဘဲ အမှားနဲ့ သက်ဆိုင်တဲ့ အယ်ရာမက်ဆွဲချုပ်တွေ ပြပေးမှာပါ။ ဖြစ်လေရှိတဲ့ ဆင်းတက်စုံအမှားတွေကို မကြာခင် တွေ ရပါမယ်။

Keywords

`from, import, def, if` စတာတွေဟာ *keyword* တွေဖြစ်တယ်။ Python ရေးတဲ့အခါ သူနေရာနဲ့ သူ အပိုပိုယ်ကိုယ်စိန့် အသုံးပြုရတဲ့ စကားလုံးတွေဖြစ်တယ်။ `from` နဲ့ `import` ကို လိုက်ဘရီ အင်ပိုလုပ် ဖို့ သုံးတယ်။ `def` ကို ဖန်ရှင် သတ်မှတ်တဲ့အခါ သုံးတယ်။ Python က သတ်မှတ်ထားတဲ့ နေရာတွေက လွှဲလို့ အခြားကိစ္စတေအတာက *keyword* တွေကို အသုံးပြုလို့ မရပါဘူး။ ဒါကြောင့် *keyword* တွေကို reserved word လို့လည်း ခေါ်တယ်။

main ဖန်ရှင်

‘Meet Karel’ ပရိုကရမ် အင်ပိုစတိတ်မန့် အပြီးမှာ တွေ့ရတာကတော့ `main` ဖန်ရှင်သတ်မှတ်ချက်ပါ။ ကြည့်ရအဆင်ပြေအောင် သူချည်းသီးသန့် တစ်ဖြတ် ပြန်ပြပေးထားပါတယ်။

```
def main():
    """Karel code goes here!"""
    move()
    move()
    move()
    pick_beeper()
    turn_left()
    move()
    move()

    turn_left()
    turn_left()
    turn_left()

    move()
    put_beeper()
# End of main
```

ဖန်ရှင် (*function*) ဆိုတာ ကိစ္စတေစုံ လုပ်ဆောင်ပေးဖို့အတွက် ယူနစ်တစ်ခုအနေနဲ့ ဖွဲ့စည်းထားတဲ့ ပရိုကရမ်ကုဒ် အစုအဝေးတစ်ခုပါပဲ။ ဖန်ရှင်ကို အသုံးပြုတဲ့အခါ ငင်းရဲ့ လုပ်ငန်းတာဝန်အတိုင်း ဖန်ရှင်က လုပ်ဆောင်ပေးမှာ ဖြစ်တယ်။ ဖန်ရှင်အသုံးပြုတာကို ‘ဖန်ရှင်ကောလ်’ (*function call*) လုပ်တာလို့ ပြောတယ်။

`main` ဖန်ရှင်သတ်မှတ်ချက်ကို အပိုင်းနှစ်ပိုင်းခဲ့ ကြည့်နိုင်တယ်။ ပထမတစ်ပိုင်း

```
def main():
```

ကို ဖန်ရှင်ခေါင်းစီး (function header) လို့ခေါ်တယ်။ ဖန်ရှင်ခေါင်းစီးမှာ ဖန်ရှင်နံမည်နဲ့ ဖန်ရှင်ပါရာ မိတာတွေကို ပိုက်ကွင်းထဲမှာ သတ်မှတ်ပေးရပြီး ကော်လုံးနဲ့ အဆုံးသတ်တယ်။ ဥပမာ x, y ပါရာ

မိတ္တ နဲ့ myfun ဖန်ရှင် အတွက်

| **def myfun(x, y):**

ပါရာမီတာမပါရင်လည်း ပိုက်ကွင်းအလွှတ် တစ်စုံ () တော့ပါရမယ်။ main ဖန်ရှင်မှာ ပါရာမီတာ မပါဘူး။ ပါရာမီတာတွေအကြောင်း နောက်ပိုင်းအခန်းတွေမှာ အသေးစိတ် လေ့လာရမှာပါ။ ကားချဲလ်ပရိုကရမ်တွေမှာ ပါရာမီတာအကြောင်း သိမ့်မလိုသေးပါဘူး။ ပါရာမီတာ မလိုတဲ့ ဖန်ရှင်တွေပဲ တွေ့ရမှာပါ။

ဖန်ရှင်သတ်မှတ်ချက် ဒုတိယပိုင်းကတော့ ဖန်ရှင်သော်ဒါပါ။ ဖန်ရှင်သော်ဒါ ရေးရင် ဘေးမျဉ်းကနေညာဘက်ကို စွာရေးရပါရမယ်။ ခွာတဲ့ အက္ဌာအဝေး တာပြေးညီးဖြစ်ရမယ်။ အင်ဒန်ထုတ် (indent) လုပ်တာလို့ ခေါ်တယ်။ ကုဒ်စထရှင်ချာကို ကြည့်လိုက်တာနဲ့ ထင်းကနဲ့ မြင်သာအောင် လုပ်ရတာပါ။ main ဖန်ရှင်မှာ ခေါင်းစီးအောက် အင်ဒန်ထုတ်လုပ်ထားတဲ့ ကုဒ်အားလုံးဟာ ဖန်ရှင်သော်ဒါပဲလို့ ချက်ချင်းသိနိုင်တယ်။ သော်ဒါ ပထမတစ်ကြောင်း

| *"""Karel code goes here!"""*

ဟာ docstring လို့ ခေါ်တဲ့ စာသား ဖြစ်တယ်။ ဖန်ရှင်နဲ့ ပါတ်သက်တဲ့ ရှင်းလင်းဖော်ပြချက်တွေ ရေးဖို့ အတွက် သုံးတာပါ။ ဒါကြောင့် docstring ကို ချမှတ် သုံးခုတဲ့ *"""* တစ်စုံကြား ညုပ်ရေးတဲ့ ကွန်းမန်း တစ်မျိုးလို့ ယူဆနိုင်တယ်။ လိုအပ်တဲ့အခါ အသေးစိတ် ထပ်ပြီး ဖော်ပြပေးမှာပါ။

Docstring အောက်မှာ တွေ့ရတာကတော့ ကားချဲလ်ကွန်းမန်းတွေဆိုတာ သိပါလိမ့်မယ်။ ကားချဲလ်ကွန်းမန်းတွေဟာ stanfordkarel လိုက်ဘရဲ့ ဖန်ရှင်တွေပါ။ တနည်းအားဖြင့် stanfordkarel လိုက်ဘရဲ့မှာ ကားချဲလ်ကွန်းမန်းတွေအတွက် ဖန်ရှင်သတ်မှတ်ချက်တွေ ပါဝင်တယ်။ ဖန်ရှင်တစ်ခုကို အသုံးပြုဖို့ အတွက် အဲဒီဖန်ရှင်ကို ခေါ်ပါတယ်။ ဒါကို function call 'ဖန်ရှင်ကောလ်' လုပ်တယ်လို့ ပြောတယ်။ ကားချဲလ်ကို ဘယ်ဘက်လုပ်စေချင်ရင် turn_left ဖန်ရှင်ကောလ် လုပ်ရပါမယ်။ ဘိပါကောက်ခိုင်းချင်ရင် pick_beeper ဖန်ရှင်ကောလ် လုပ်ရပါမယ်။ ဖန်ရှင်ကောလ် လုပ်တဲ့ ပုံစံက

| **turn_left()**
pick_beeper()

စသည်ဖြင့် ဖြစ်တယ်။

Python မှာ အင်ဒန်ထုတ်ကို ဖြစ်ကတတ်ဆန်း လုပ်လို့မရဘူး။ ဘေးမျဉ်းကနေ စွာတဲ့ အက္ဌာအဝေး မည်းတာနဲ့ ဆင်းတက်စိအမှား ဖြစ်တယ်။ မလိုတဲ့နေရာမှာလည်း စွာရေးလို့ မရဘူး။ ခေါင်းစီးကို ဘေးမျဉ်းနဲ့ စွာထားကြည့်ပါ။ အယ်ရာဖြစ်တာကို တွေ့ရမယ်။ အင်ပိုဒ်တိတ်မန်းလည်း ဘေးမျဉ်းနဲ့ ကွာေနလို့မရဘူး။ အခြား language တွေမှားလည်း အင်ဒန်ထုတ်လုပ် ရေးကြပေမဲ့ Python မှာလောက် မတင်းကျပ်ဘူး။ အင်ဒန်ထုတ်မလုပ်လည်း ဆင်းတက်စိမှားတာ မဖြစ်ဘူး။

ကားချဲလ်ပရိုကရမ်တစ်ခုမှာ main ဟာ အထူးတာဝန်တစ်ခု လုပ်ဆောင်ပေးရတယ်။ အဲဒါကတော့ ပရိုကရမ်ဝင်းဒီးမှာ **Run Program** ခလုတ်နှိပ်လိုက်ရင် တုံ့ပြန် လုပ်ဆောင်ပေးရတာပါ။ ဒါကြောင့် ကွန်းမန်းတွေဟာ ခလုတ်နှိပ်တော့မှုပဲ စအလုပ်လုပ်တာ ဖြစ်တယ်။

Entry Point (ပရိုကရမ်ခုမှတ်)

'Meet Karel' ပရိုကရမ်မှာ main ဖန်ရှင်နောက် အောက်ခုံးအပိုင်းဟာ ပရိုကရမ် run တဲ့အခါ ပထမဆုံး စတင်လုပ်ဆောင်ပေးရမဲ့ ဖန်ရှင်ကို ဖော်ပြပေးတာပါ။ 'အန်ထရီဗိုင့်' လို့ခေါ်တယ်။

```

if __name__ == "__main__":
    run_karel_program("meet_karel")

```

run_karel_program ဖန်ရှင်ဟာ ကားရဲပရိုကရမ် တစ်ခုအတွက် အန်ထရီပိုင့် ဖြစ်တယ်။ ပရိုကရမ် တက်လာတာနဲ့ တစ်ပါတည်း ခေါ်တင်ချင်တဲ့ ကမ္ဘာကို ဒီဖန်ရှင်မှာ ထည့်ပေးတယ်။ `meet_karel.w` ကမ္ဘာကို စစ်ချင်းခေါ်တင်ထားချင်ရင် "meet_karel" ထည့်ပေးရမယ်။ ဖိုင်မရှိတဲ့ ကမ္ဘာကို ထည့်ထားရင် အယ်ရာတက်ပြီး ပရိုကရမ်ပွင့်လာမှာ မဟုတ်ဘူး။ ကမ္ဘာမထည့်ပေးထားဘဲ ဒီလို

```
run_karel_program()
```

ဆိုရင် 8×8 အရွယ် default ကမ္ဘာကို တင်ပေးပါတယ်။

ကားရဲလ်ကမ္ဘာတစ်ခုကို လိုချင်တဲ့ ပုံး ဒီဇိုင်းဆွဲပြီး ဖိုင်နဲ့ သိမ်းထားလိုရတယ်။ ကမ္ဘာ ပုံစံချကဲ့ ပရိုကရမ်လည်း ရှိတယ်။ ကမ္ဘာဖိုင်တွေက `.w` ဖိုင် အိပ်စ်တန်းရှင်းနဲ့ ဖြစ်တယ်။ ဒီစာအုပ်မှာပါတဲ့ ဥပမာတွေလေ့ကျင့်ခန်းတွေ အားလုံးအတွက် လိုအပ်တဲ့ ကမ္ဘာတွေကို အဆင်သင့်ပေးထားမှာပါ။ ကိုယ့်ဟာကို လုပ်ဖို့ မလိုဘူး။

၁.၃ ကားရဲလ် ပရိုကရမ် run ခြင်း

ଓପାର୍କିଂ ଜ

title

နောက်ဆက်တဲ့ က

လိုအပ်သည့် ဆောဖံ့ဌများ ထည့်သွင်းခြင်း

အခြား အင်ဂျင်နီယာ/သိပ္ပံဌ ပညာရပ်တွေလိုပဲ ပရိုကရမ်မင်းလေ့လာတဲ့အခါ လက်တွေ့လုပ်ကြည့်ဖို့ အင်မတန်အရေးကြိုးပါတယ်။ လက်တွေ့ရေးမကြည့်ဘဲ စမ်းသပ်မကြည့်ဘဲ သဘောတရားပိုင်းဆိုင်ရာတွေကို အမှန်တကယ်နားလည်ဗုံး မဟုတ်ပါဘူး။ အခြားပညာရပ်တွေထက် ကွန်ပျိုးတာ ပရိုကရမ်မင်းရဲ့ အားသာချက်တစ်ခုကတော့ လက်တွေ့စမ်းသပ်ခန်းကြိုးတွေ ရှိစရာမလိုတာပါပဲ။ စမ်းသပ်ပစ္စည်းတွေ လည်း များများစားစား မလိုအပ်ဘူး။ ကွန်ပျိုးတာ တစ်လုံးနဲ့ လိုအပ်တဲ့ ဆောဖံ့ဌပဲတဲ့ ရှိရင်ရပြီ။ ဆောဖံ့ဌတွေကလည်း ပိုက်ဆံကုန်စရာမလိုဘူး။ မပေးဘဲ သုံးလို့ရတာ။

ပရိုကရမ်လက်တွေ့ရေး လေ့ကျင့်ဖို့အတွက် လိုအပ်တဲ့ ဆောဖံ့ဌတွေ ထည့်ထားရပါမယ်။ Python programming language အတွက် Python ဆောဖံ့ဌ ထည့်ရပါမယ်။ Python ဆောဖံ့ဌမရှိဘဲ Python ကုဒ်တွေ၊ Python ပရိုကရမ်တွေ run လို မရပါဘူး။ Python အပြင် ပရိုကရမ်ကုဒ် ရေးဖို့ အတွက် အထောက်အကူဗြာ ဆောဖံ့ဌပဲတစ်ခုလည်း လိုအပ်တယ်။

ပရိုကရမ် ကုဒ်ရေးဖို့အတွက် PyCharm သိမဟုတ် Visual Studio Code (VS Code) ကို အသုံးပြုနိုင်ပါတယ်။ အက်ဆေးတစ်ပုဒ်ရေးတဲ့အခါ မိုက်ခရိုဆောဖံ့ဌ Word နဲ့ရေးလိုရသလို ရှိုးရှိုးရှင်းရှင်း Notepad လောက်နဲ့ ရေးလိုလည်း ရတာပါပဲ။ အက်ဆေးရဲ့ အဓိပ္ပာယ်ကသာ အစိုကပါ။ ပရိုကရမ်ကုဒ် ရေးတဲ့အခါမှာလည်း ခီးသောပါပဲ။ ဆောဖံ့ဌပဲရေးကြိုးတွေ တည်ဆောက်တဲ့အခါ လိုအပ်မဲ့ ဖီချာတွေ အားလုံး စုံစုံလင်လင်ပါပြီးသား PyCharm လို Integrated Development Environment(IDE) ဆောဖံ့ဌပဲနဲ့ ကုဒ်တွေရေးလိုရသလို ပါ့ပေါ့ပါ့ပါ့နဲ့ လိုအပ်မှုပဲ လိုတဲ့ဖီချာအတွက် extensions (plugin လိုလည်းခေါ်ယ်) ထည့်သွင်းရတဲ့ Visual Studio Code (VS Code) လို ကုဒ်အသိဒ်ဘာ (Code Editor) ဆောဖံ့ဌပဲ့ သုံးပြီး ရေးရင်လည်း ရတာပါပဲ။ နောက်ဆုံး ကုဒ်နည်နည်း၊ ဖိုင်နည်းနည်း ရှိုးရှင်းတဲ့ ပရိုကရမ်လေးတွေဆုံးရင် Notepad နဲ့ ရေးလိုတောင် ရပါတယ်။ ကုဒ်တွေများမယ် ဖိုင်တွေ များမယ်၊ အသင်းအစွဲလိုက် ပူးပေါင်းရေးရတဲ့ ပရိုကရမ်မျိုးတွေ ဆိုရင်တော့လည်း Notepad လောက်နဲ့ အဆင်မပြနိုင်တော့ဘူးပေါ့။

PyCharm ရော VS Code ထည့်သွင်းနည်းပါ ဖော်ပြပေးပါမယ်။ မိမိ နှစ်သက်ရာ အဆင်ပြောရာ သို့မဟုတ် နီးစပ်ရာ ပရိုကရမ်ဟာ အသိမိတ်ဆွေ အကြုပြုတဲ့ တစ်ခုကို ရွေးချယ်သုံးပါ။ စလေ့လာသူအနေနဲ့ PyCharm ကို အသုံးပြုတာ ပိုအဆင်ပြောမယ်လို ထင်တယ်။ PyCharm သုံးကြည့်လို မိမိကွန်ပျိုးတာ မှာ နေးလွန်းတယ်ဆိုရင် VS Code ကိုစမ်းကြည့်ပါ။ နှစ်ခုလုံး စက်အရမ်းကောင်း/မြင့် ဖို့ မလိုပါဘူး။ တော်ရုံး အတန်အသင့်ကောင်းတဲ့ စက်လောက်နဲ့ အဆင်ပြောပါတယ်။

မိမိကိုယ်တိုင်က ကွန်ပျိုးတာ အသုံးပြုပဲ အခြေခံ အားနည်းပြီး ဖော်ပြပေးထားတဲ့ အတိုင်း တစ်ဆင့် ချင်း အင်စတောလ် လုပ်ဘာလည်း အဆင်မပြောဖြစ်နေရင် ဒီဘာအုပ်ရဲ့ အောက်ပါ ဖော်သွာတ်ခံနဲ့ ယူကျိုး။

ချုပ်နယ်တွေမှာ ကြည့်ရှုမေးမြန်း အကူအညီ တောင်းဆိုင်ပါတယ်။ ဒါမှမဟုတ် အတွေးအကြိုးတဲ့ နီးစပ်ရာ အသိမိတ်ဆေး/ညီကိုမောင်နှစ်မဲ တစ်ယောက်ယောက်ရဲ့ အကူအညီယူပြီး အင်စတောလ်လုပ်ပါ။

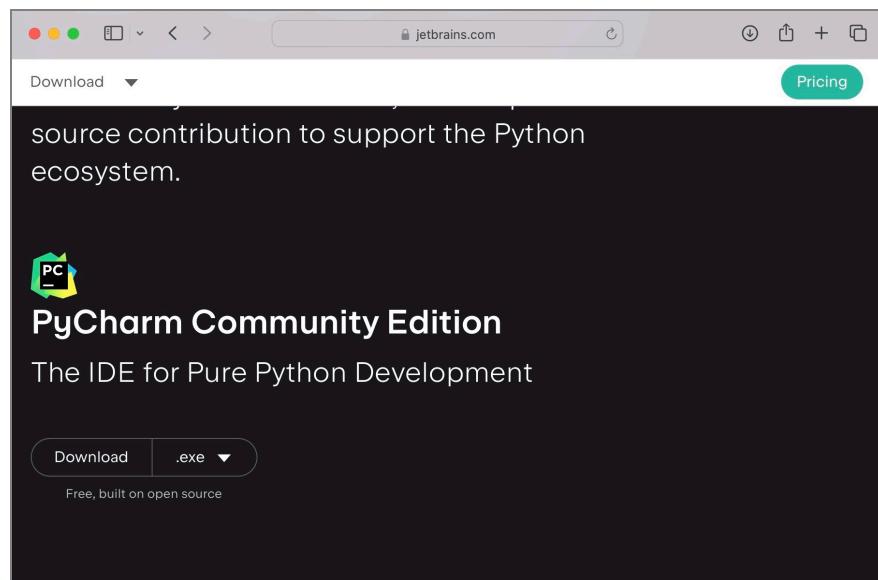
<https://www.facebook.com/bpwp>

<https://www.youtube.com/bpwp>

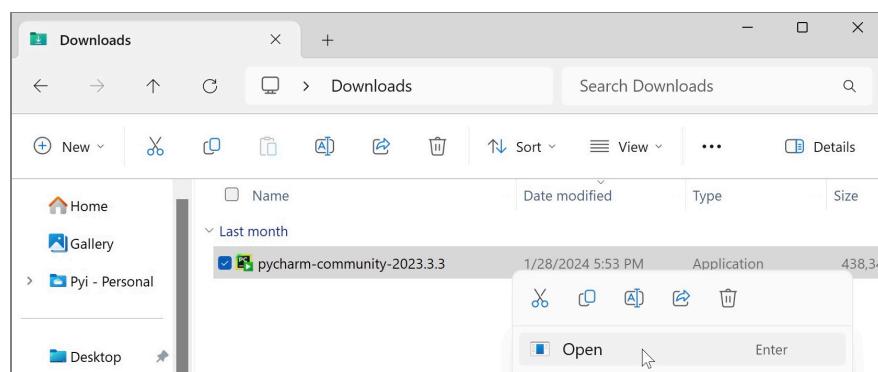
PyCharm အတွက် အရင်ဖော်ပြပေးပါမယ်။ VS Code အတွက် စာမျက်နှာ ၂၄ မှာ ကြည့်ပါ။

Python နှင့် PyCharm IDE ဆည့်သွင်းခြင်း

<https://www.jetbrains.com/pycharm/download/> လင့်ကိုဖွင့်ပါ။ ဝဘ်စာမျက်နှာ အောက်ဘက် နည်းနည်း ဆွဲချက်ရင် PyCharm Community Edition ခေါင်းလုပ်ခလုပ်ကို တွေ့ရပါမယ်။ ပုံ (၁/၁) ကိုကြည့်ပါ။



ပုံ ၁/၁



ပုံ ၂/၂

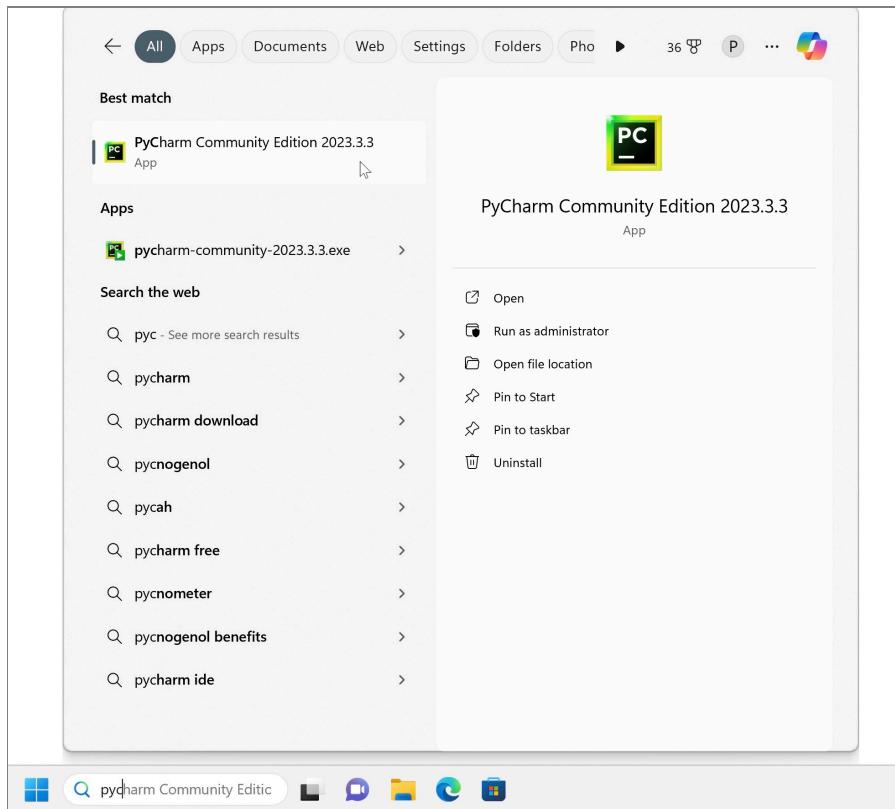
PyCharm Community Edition ကို အောင်လုပ်လုပ်ပါ။ (ဝဘ်စာမျက်နှာ အပေါ်ပိုင်းက ၀၂၍ သုံးရတဲ့ PyCharm Professional ကို အောင်လုပ် မှားမလုပ်မိန့် သတိပြုပါ)။ အင်စတော်လာဖိုင်ကို ညာကလစ်နှုပ်ပြီး Open လုပ်ပါ (ပုံ ၂/၂ ကိုကြည့်ပါ)။ Yes/No မေးတဲ့အခါ Yes နှုပ်ပါ။ Next > ကို နှုပ်၍ ရှေ့ကိုဆက်သွားပြီး နောက်ဆုံးမှာ Install နှုပ်ပြီး ကွန်ဖန်လုပ်ပါ။ အင်စတော်ပြီးသွားရင် Finish နှုပ်ပါ။

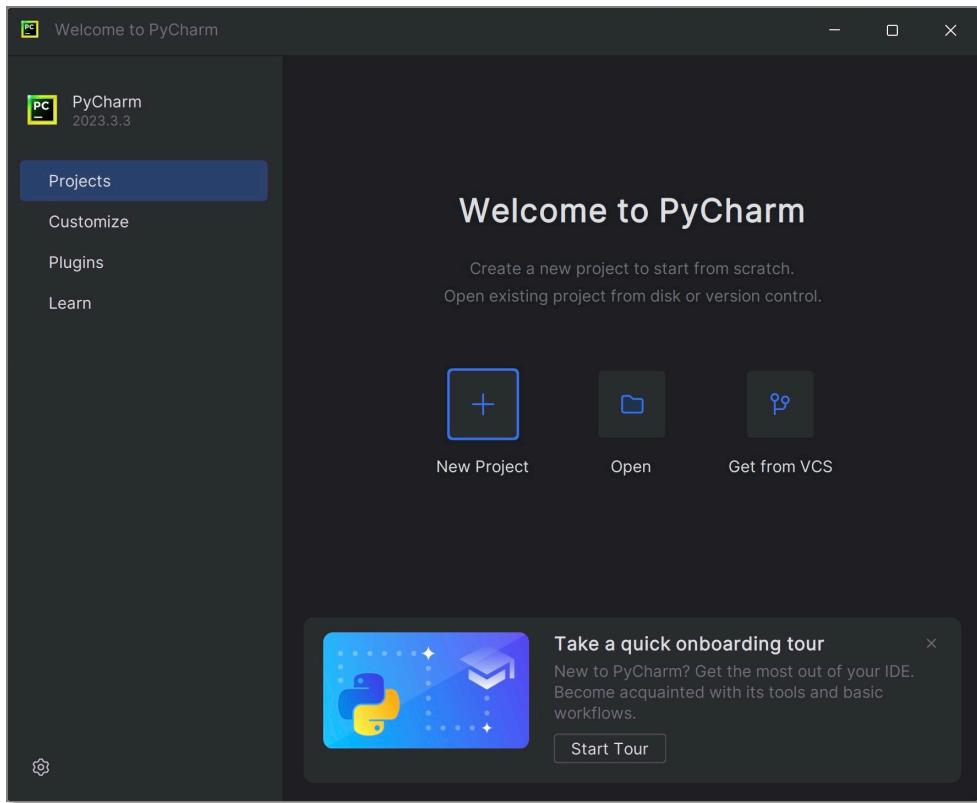
၀၂:၃၃ Taskbar Search ကနေ PyCharm ကိုရှာပြီးဖွံ့ဖြိုးပါ (ပုံ ၂/၃ ကို ကြည့်ပါ)။ သဘောတူ ကြောင်း ကွန်ဖန်လုပ်ခိုင်းရင် ချက်ချက်ဘောက်စ် ချက်ချက်လုပ်ပြီး Continue နှုပ်ပါ။ ဒေတာပို့ချင်လား ထပ် မေးပါလိမ့်မယ်။ Don't Send နှုပ်ပါ။ Welcome စာရင်ကိုပေါ်လာမယ်။ ပုံ (၂/၄) မှာ ကြည့်ပါ။

ဒီစာရေးနေချိန် လက်ရှိ PyCharm ဗားရှင်းက ၂၀၂၃ ပါ။ သိပ်မကြာခင် ၂၀၂၄ ထွက်ပါတော့ မယ်။ အကယ်၍ လက်ရှိဗားရှင်းထက် နိမ့်တဲ့ဗားရှင်းတွေကို အောင်လုပ် လုပ်ချင်ရင် အောက်ပါ လင့်ကို သွားပါ။

<https://www.jetbrains.com/pycharm/download/other.html>

ဗားရှင်း ၂၀၂၄/၂၅ ထွက်ပြီးတဲ့ အချိန်မှာ ၂၀၂၃ ဗားရှင်းကို လိုချင်ရင် ဝဘ်စာမျက်နှာမှ ရှာပြီး အောင်လုပ် လုပ်ပါ။ ၂၀၂၃ မှာလည်း ဗားရှင်းအခွဲတွေ ရှိပါသေးတယ်။ လက်ရှိအမြင့်ဆုံး ဗားရှင်းအခွဲ (ဥပမာ ၂၀၂၃.၃.၃) ကို သုံးလိုပါတယ်။ ၂၀၂၄/၂၅ သုံးမယ်ဆုံးရင်လည်း ပြဿနာတော့ မရှိပါဘူး။ Update ဗားရှင်းဖြစ်တဲ့အတွက် ပုံတွေမှာ ပြထားတာနဲ့တော့ ကွားခြားချက်တရာ့၏ ရှိကောင်းရှိနှင့်ပါတယ်။

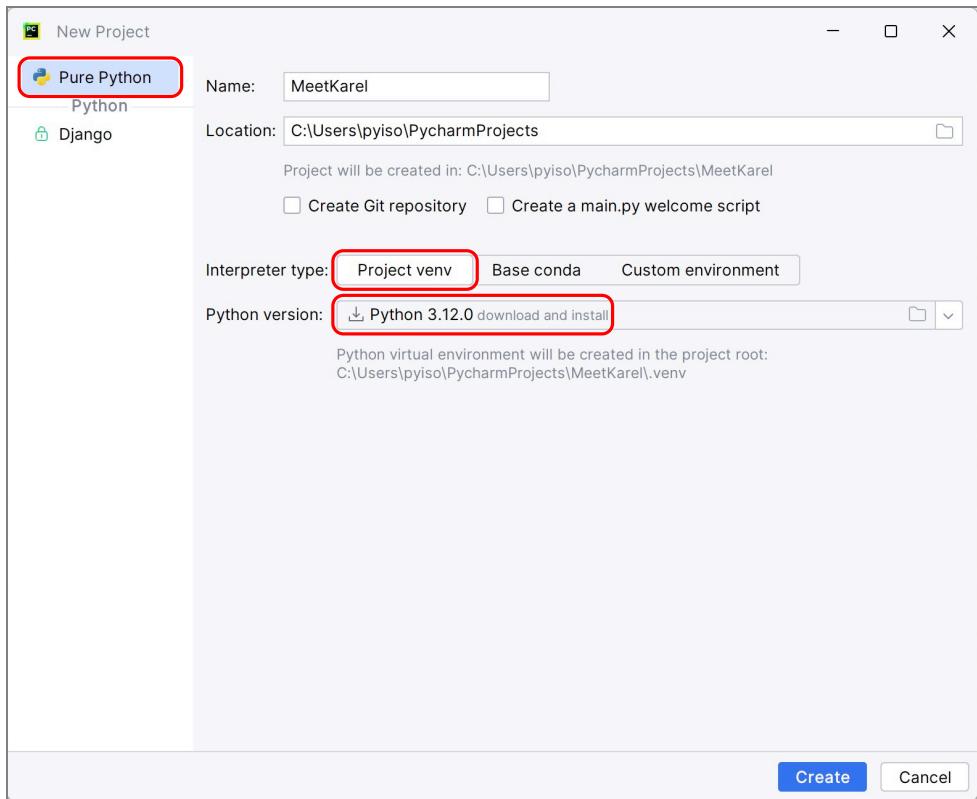




ပုံ ၂/၄

PyCharm IDE ဆိတာဘာလ

PyCharm ဟာ Python နဲ့ ဆောင်ဝဲ ရေးဖို့အတွက် အထောက်အကူဗြိ Integrated Development Environment(IDE) ဆောင်ဝဲဖြစ်ပါတယ်။ စာစီတရုက်လုပ်တဲ့အခါ Microsoft Word ကို အသုံးပြု ကြသလိုပဲ Python ကုဒ်ရေးဖို့ PyCharm ကိုထုတဲ့ သဘောပေါ့။ PyCharm IDE ၏ Python ပရောဂျက်တွေ အတွက် အခိုက်ရည်ရွယ်တယ်။ အထောက်အဦးတစ်ခု၊ တံတားတစ်ခု ဆောက်လုပ်တာ ကို ပရောဂျက်လို့ ပြောလေ့ရှိသလို ပရိုကရမ်/ဆောင်ဝဲတစ်ခု တည်ဆောက်တာကိုလည်း ပရောဂျက်လိုပဲ သုံးနှုန်းပါတယ်။ တစ်ဦးတစ်ယောက်တည်း ရေးတဲ့ ပရိုကရမ်အသေးလေးတွေ အတွက် PyCharm ကို အသုံးပြုနိုင်သလို ပရိုကရမ်မာတွေ အဖွဲ့လိုက်နဲ့ တည်ဆောက်ရတဲ့ ပရောဂျက်ကြီးတွေ အတွက်လည်း သုံးပါတယ်။ ဒီစာအုပ်မှာတော့ PyCharm ရဲ့ အဆင့်မြင့်ဖို့ချာတွေကို အသုံးပြုမှာ မဟုတ်ပါဘူး။ ပရိုကရမ်းမင်း စလေ့လာသူတွေကို လွှာယ်ကူးအဆင်ပြေစေတဲ့ အခြေခံ ဖို့ချာတွေလောက်ပဲ အသုံးပြုမှုပါ။



ပုံ ၁/၅ ပရောဂျက် အသစ်ယူခြင်း

PyCharm ပရောဂျက်ဆောက်ခြင်း

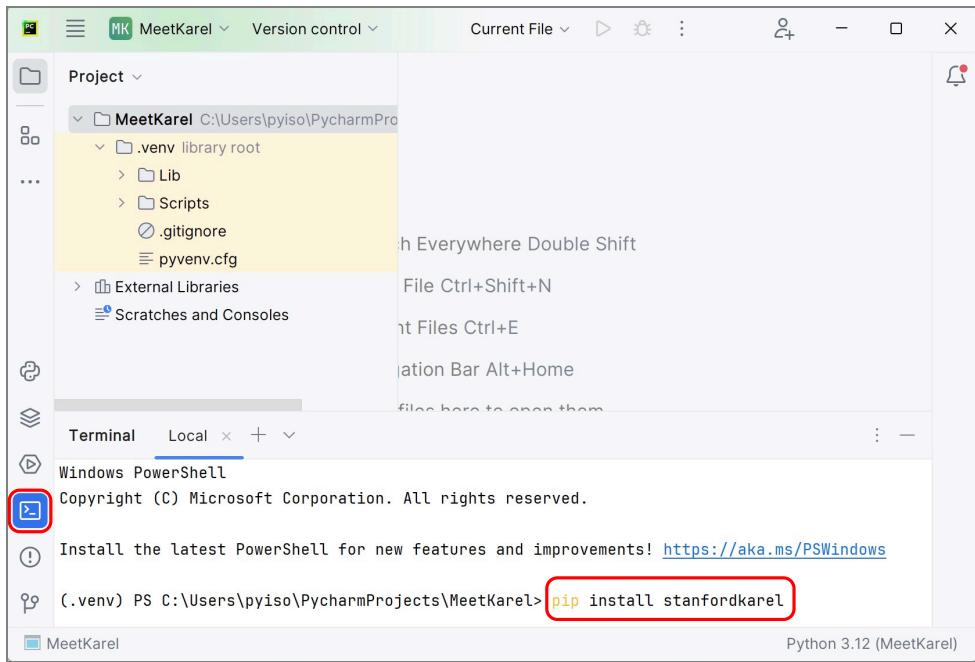
အင်စတောလုပ်ပြီးရင် PyCharm IDE ကိုဖွံ့ဖြိုး စေရင်မှ သို့မဟုတ် File မိန္ဒီးမှ New Project နှင့်ပြီး ပရောဂျက် အသစ်ယူပါ (ပြီးခဲတဲ့ ပုံ ၁/၄ ကို ပြန်ကြည့်ပါ)။ နံမည်ကို MeetKarel သေးပါ။ ပုံ (၁/၅) မှာ တွေ့ရတဲ့အတိုင်း Pure Python, Proj venv နဲ့ Python ဗားရှင်း 3.12.xx ကို ရွေးပါ။ xx က အမြားကဏ္ဍား ဖြစ်နေနိုင်တယ်။ အစိက ဗားရှင်း 3.12 သာ ဖြစ်ပါခေါ်။ Create ခလုတ်နှင့်ပါ။ ပရောဂျက်အတွက် Python ကို အင်စတောလုပ်ပါလိမ့်မယ်။

မိမိလေ့လေနေတဲ့ အခန်းတစ်ခုချင်းစီအတွက် ပရောဂျက်တစ်ခု ဆောက်နိုင်ပါတယ်။ အခန်း (၂) အတွက် Chapter02၊ အခန်း (၃) အတွက် Chapter03 စသည်ဖြင့်။ သက်ဆိုင်ရာအခန်းအလိုက် ကုဒ္ဓိုင်တွေကို ပရောဂျက်တစ်ခုစီမှာ ထားတဲ့အတွက် ဖိုင်တွေများပြီး ရှုပ်ထွေဖောင်းပွဲနေတဲ့ ပြဿနာ မရှိတော့ဘူး။ ကုဒ္ဓိုင်တွေကို ပရောဂျက်တစ်ခုခဲ့မှာပဲ ဖိုဒ်အလိုက်ခဲ့ထားလို့ရပေမဲ့ စလေ့လာသူအနေနဲ့ ပရောဂျက်တစ်ခုချင်း ခဲ့ထားတာလောက် မလွှယ်ကူဘူး။

ပရောဂျက် အသစ်ယူတဲ့အခါ တည်နေရာ Location: ကို သူ့ရှိခို့အတိုင်း ထားနိုင်သလို မိမိထားချင်တဲ့နေရာကို ညာဘာက်စွန်း ဖိုဒ်အိုင်ကွန်လေးနှင့်ပြီး ပြောင်းလို့ရတယ်။

stanfordkarel လိုက်ဘရီ အင်စတောလုပ်ခြင်း

ပုံ (၂/၆) မှာ အနီရောင် ဝိုင်းပြထားတဲ့ အိုင်ကွန်ကို နှင့်ပြီး Terminal ကိုဖွံ့ဖြိုး ဖြစ်ပါ။ Terminal မှာ အောက်ပါ ကွန်မန်းဖြင့်



ပုံ ၂/၆ Karel လိုက်ဘရိ အင်စတောလ်လုပ်ခြင်း

```
pip install stanfordkarel
```

ကားရဲလ်လိုက်ဘရိကို အင်စတောလ်လုပ်ပါ။ ပုံ (၂/၆) မှာ အနီရောင် စိုင်းပြထားပါတယ်။ ခက္ကာတဲ့ အခါ အခုလို မက်ဆော်တွေ ကျလာပါလိမ့်မယ်။

```
Collecting stanfordkarel
  Downloading stanfordkarel-0.2.7-py3-none-any.whl (51 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 51.9/51.9 kB 443.1 kB/s
    eta 0:00:00
Installing collected packages: stanfordkarel
Successfully installed stanfordkarel-0.2.7
```

```
[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
(.venv) PS C:\Users\pyiso\PycharmProjects\MeetKarel>
```

ဟိုက်လိုက်ပြထားတဲ့ မက်ဆော်တွေရရင် အင်စတောလ်လုပ်တာ အောင်မြင်လိုပါ။

မှတ်ချက်။ ပရောဂျက် အသစ်ဆောက်တိုင်း လိုအပ်တဲ့ လိုက်ဘရိကို တစ်ခါတပ်ပြီး အင်စတောလ် လုပ်ရပါမယ်။ ကားရဲလ်အခန်း ပရောဂျက်တစ်ခုစီအတွက် stanfordkarel လိုက်ဘရိကို အထက်ဖော်ပြပါအတိုင်း အင်စတောလ် လုပ်ဖို့လိုတယ်။

နမူနာ ကားရဲလ် ကဗ္ဗာနှင့် ပရိုဂရမ်ကုဒ် ဖိုင်များထည့်ခြင်း

meet_karel.zip ဖိုင်ကို ဒီလင့် <http://tinyurl.com/3mmmm9c7j> ကနေ ဒေါင်းလုဒ်လုပ်ပါ။ ငါး zip ဖိုင်ကို extract လုပ်ပါ။ meet_karel နံမည်နဲ့ ဖိုဒါတစ်ခု ရလာပါမယ်။ ငါးဖိုဒါထဲမှ အောက်ပါ worlds ဖိုဒါနှင့် .py ဖိုင်အားလုံးကို ကော်ပီလုပ်ပါ။

- worlds
 - ▶ meet_karel.w
 - ▶ move_beeper_to_other_side.w
- meet_karel.py
- move_beeper_to_other_side.py
- world_editor.py

MeetKarel ပရောဂျက်ထဲတွင် ကူးထည့်ပါ။ ပင်မ ပရောဂျက် MeetKarel (ပုံ က/၇ မှာ မြှားပြထား) ပေါ်မှာ ညာကလစ်နိုပ်ပြီး Paste လုပ်ရမှာပါ။ ကော်ပီကူးထည့်လိုက်တဲ့ ဖိုင်တွေက ပုံမှာ တွေ့ရတဲ့ အတိုင်း MeetKarel ဖိုဒါအောက်မှာ ရှိသင့်ပါတယ်။

အခန်းအလိုက် နမူနာ ကုဒ်ဖိုင်တွေ ထည့်ပေးထားတဲ့ .zip ဖိုင်တွေကိုလည်း အထက်ပါအတိုင်း အလားတူ လုပ်ရပါမယ်။ ပရောဂျက်အသစ်ဆောက်၊ .zip ဖိုင်ကို ဖြည့်၊ ရလာတဲ့ ဖိုဒါထဲက ဖိုင်တွေကို ပင်မ ပရောဂျက် ဖိုဒါထဲ ကော်ပီကူးထည့် ရုံပါပဲ။

meet_karel.py ဖိုင်ကို ကလစ်နှစ်ချက်နှင့် ဖွင့်ပါ။ ပုံ (က/၇) မှာ အနီးပိုင်းထားတဲ့ ကုဒ်အယ်ဒီတာ (code editor) ပွင့်လာပါမယ်။ အဲဒီ မူရင်းဖိုင်မှာ အောက်ပါအတိုင်း ဆက်လက် ပြင်ဆင်ဖြည့်စွက်ပါ။

```
from stanfordkarel import *

def main():
    """Karel code goes here!"""
    move()
    move()
    move()
    pick_beeper()
    turn_left()
    move()
    move()

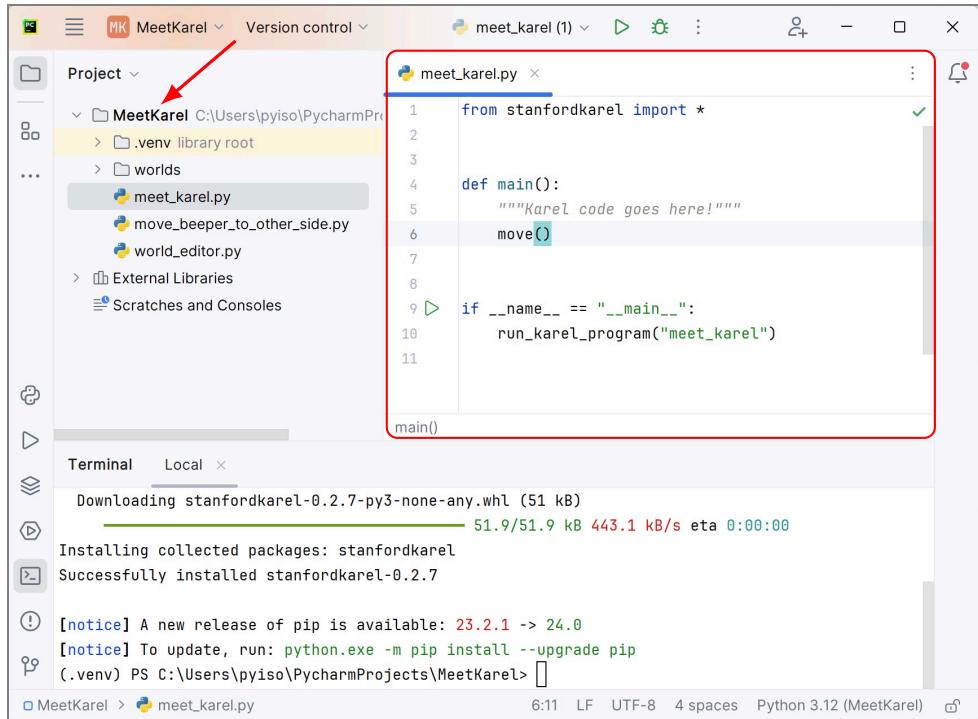
    turn_left()
    turn_left()
    turn_left()

    move()
    put_beeper()

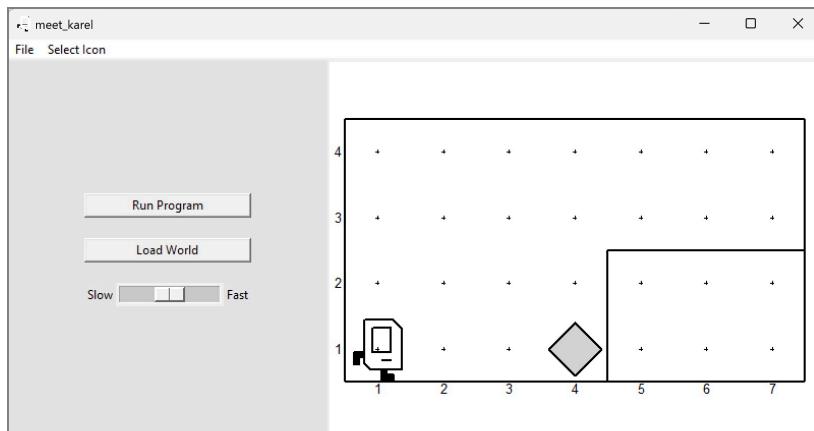
if __name__ == "__main__":
    run_karel_program("meet_karel")
```

run ထားတဲ့ ပရိုကရမ်တစ်ခုကို တစ်ခါထပ် run ရင် Cancel (သို့) Stop and Rerun လုပ်မှာ လူမေးတယ်။ Stop and Rerun လုပ်ရပါတယ်။

meet_karel.py ဖိုင်ကို ညာကလစ်နှင့်ပြီး Run 'meet_karel' လုပ်ပါ (အယ်ဒီတာ ဧရိယာမှာ ညာကလစ်နှင့်ပြီး Run 'meet_karel' လုပ်လိုလည်း ရပါတယ်) ။ ပုံ (က/၈) က ကားရဲ့ပုံ ပရိုကရမ် တက်လာရင် Run Program ခလုတ်ကိုနိုပ်ပါ။ ဘိပါကို ကားရဲ့လ်က ရွှေပေးပါလိမ့်မယ်။



ပုံ ၃/၃



ပုံ ၃/၈

ဆင်းတက်စ်အမှားများ

အကယ်၍ ပရိုက်မဲ့ run ပေါ်လို့မရရင် ကုဒ်ရေးတာမှားနေလို့ ဖြစ်နိုင်တယ်။ မမိရေးထားတာကို စာမျက်နှာ (၁၇) က ပရိုက်မဲ့ကုဒ်နဲ့ နှင့်ယူဉ် စစ်ဆေးကြည့်ပါ။ PyCharm အယ်ဒီတာမှာ အနိုင်တွန်းလေးတွေ (ပုံ က/၉) ပြတဲ့နေရင် အဲဒီနေရာတွေမှာ ဆင်းတက်စ်မှားနေလို့ (သို့) လိုက်ဘရှိမထည့်ရသေးလို့ပဲ။

လိုက်ကွင်းကျို့နေတာက အဖြစ်များတဲ့ အမှားပါ။ ကျို့ခဲ့လို့ မရပါဘူး။ အင်ဒန်တေးရှင်း (indentation) လုပ်ရမဲ့နေရာမှာ မလုပ်ထားရင်လည်း ပြဿနာဖြစ်တယ်။ move, turn_left တွေကို ဘေးမျဉ်းလာဘက်ဆွဲပြီး အင်ဒန်လုပ်ပေးရမယ်။ အဲဒါတွေ ကရာမစိုက်မိရင် ဆင်းတက်စ်အမှားဖြစ်ပြီး ပရိုက်မဲ့ run လို့ မရနိုင်ဘူး။

Terminal မှာ ထုတ်ပေးတဲ့ မက်ဆွဲချုပေါ်တွေကို ကြည့်ပြီးတော့လည်း ဘာပြဿနာဖြစ်နေလဲ မှန်းဆ လို့ရနိုင်တယ်။ ဘာကြောင့်ဖြစ်နိုင်လဲ ဆက်စပ်စဉ်းစားလို့ ရတယ်။ ဥပမာ ဖြစ်တဲ့ပြဿနာအလိုက် အခုလို တွေ့ရပါမယ်။

```
File "c:\Users\pyiso\VS Code\meet_karel\meet_karel.py", line 6
    move(
        ^
SyntaxError: '(' was never closed

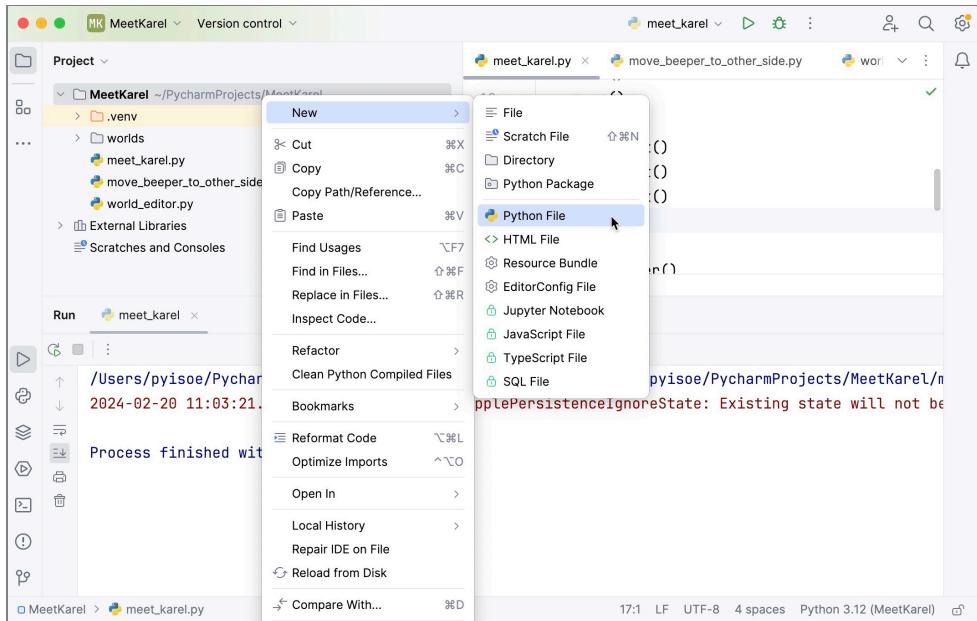
File "c:\Users\pyiso\VS Code\meet_karel\meet_karel.py", line 7
    move()
        ^
IndentationError: unexpected indent

Traceback (most recent call last):
  File "c:\Users\pyiso\VS Code\meet_karel\meet_karel.py", line 1,
    in <module>
      from stanfordkarel import *
ModuleNotFoundError: No module named 'stanfordkarel'
```

Python ဖိုင် အသစ်ယူခြင်း

MeetKarel ပင်မ ပရောဂျက်ဖို့ဒါပေါ်မှာ ညာကလစ်နှုပ်ပြီး Python ဖိုင် အသစ်ယူနိုင်ပါတယ်။ Python ဖိုင်တွေက .py အိုပ်စ်တန်းရှင်းနှုပါ။ ကားရဲလ်ပရိုဂရမ်တစ်ခုကို Python ဖိုင်တစ်ခု ထားပါမယ်။ ပင်မ ပရောဂျက်ဖို့ဒါ အောက်မှာပဲ တိုက်ရိုက်ရှိရပါမယ်။

နောက်ပိုင်း အဆင့်မြင့်လာရင် ပရိုဂရမ်တစ်ခုအတွက် ပရောဂျက်တစ်ခု ထားနိုင်တွေ အပြင် ပရိုဂရမ်အတွက် လိုအပ်တဲ့ ရုပ်ပုံတော့၊ အခြားဖိုင်တွေ (config ဖိုင်၊ setting ဖိုင် စသည်ဖြင့်) လည်း ပါနိုင်တယ်။ ပင်မပရောဂျက် အောက်မှာပဲ ဖိုင်တွေက တိုက်ရိုက်ရှိဖို့လည်း မလိုတော့ဘူး။ ဆက် စပ်ရာ ဖိုင်တွေကို အမျိုးအစားအလိုက် ဖန်ရှင်အလိုက် ဖို့ဒါတွေခဲ့ပြီး စနစ်ကျ စီစဉ်ဖွဲ့စည်း ထားရမှာပါ။ ပရောဂျက်တစ်ခုမှာ ဖိုင်တွေကို စနစ်တကျ စုဖွဲ့ထားဖို့ အရေးကြီးပါတယ်။



ပုံ ၂/၁၀

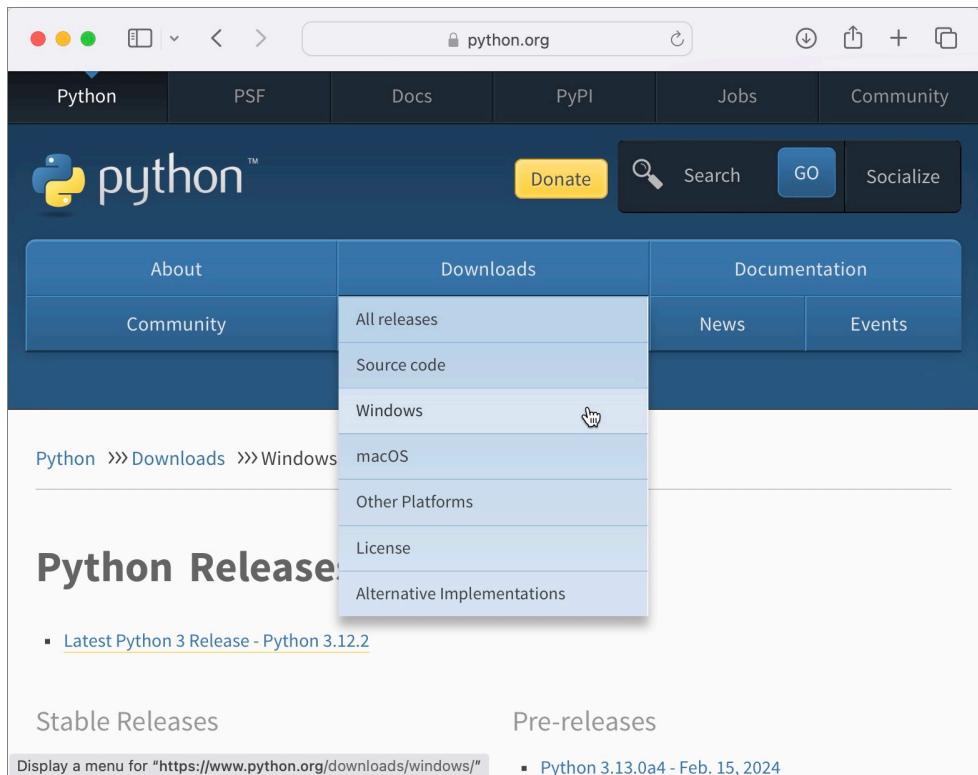
Visual Studio Code နှင့် Python အင်စတောလ်လုပ်ခြင်း

Visual Studio Code(VS Code) ဟာ ပရိုဂရမ်မာအများစုံ ကြိုက်နှစ်သက်တဲ့ မောဒန် ကုဒ်အယ်ဒီတာ တစ်ခုပါ။ Python, JavaScript, C++ စတဲ့ programming language အမျိုးမျိုးအတွက် အသုံးပြု နိုင်ပါတယ်။

PyCharm မှာတော့ ပရောဂျက်အသစ်ဆောက်ရင် Python ပါ တစ်ခါတည်း ဒေါင်းလုဒ်လုပ်ပြီး အင်စတောလ် လုပ်လို့ရတယ်။ VS Code နဲ့ Python ရေးမယ်ဆိုရင် Python Programming Language ကို သီး၌ ဒေါင်းလုဒ်လုပ်ပြီး အင်စတောလ် လုပ်ရပါမယ်။

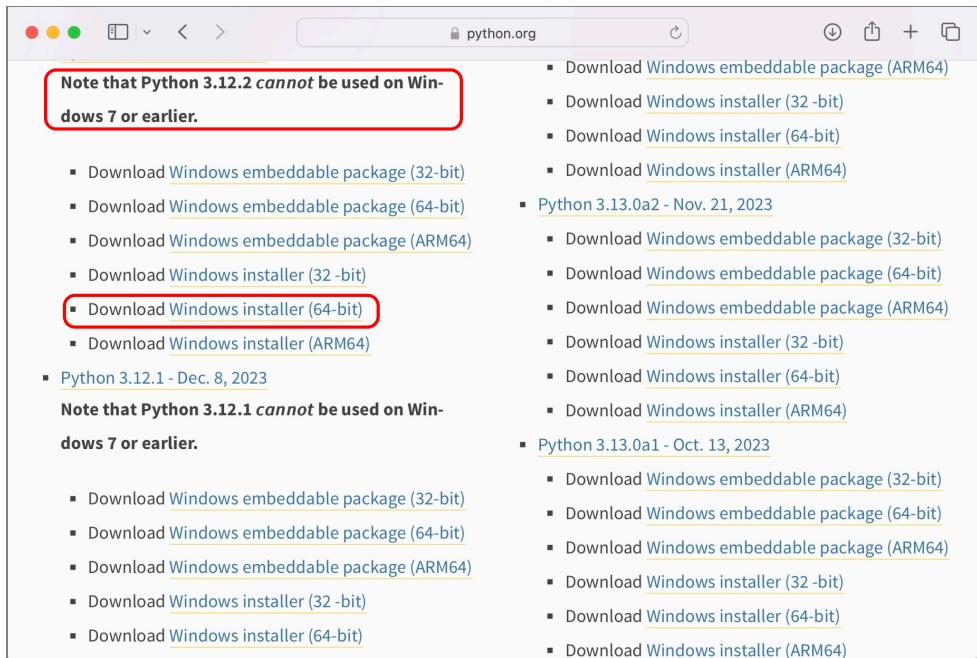
Python အင်စတောလ်လုပ်ခြင်း

ဒီလုပ် <https://www.python.org/> ကိုဖွေ့စ်ပါ။ Download မိန္ဒီးမှ Windows နိုပ်ပါ (ပုံ က/၁၁ ကို ကြည့်ပါ)။ Apple ကွန်ပျိုးတာအတွက် ဆိုရင် macOS ရွှေးရပါမယ်။ လူများစုသုံးတဲ့ ပိုက်ခရီးဆော် ဝင်း ခြီးအတွက် အင်စတောလ်လုပ်နည်းကို အခိုက်ပြေားမှာပါ။ ပုံ က/၁၂ မှာလို့ တွေ့ရပါလိမ့်မယ်။ အင်စတော်လာ ဗားရှင်း 3.12 ထဲက လက်ရှိအမြင့်ဆုံး ကိုရွေးပါ။ ဒီစာရေးချိန်မှာ 3.12.2 ဟာ အမြင့်ဆုံးဗားရှင်းပါ။ Windows Installer (64-bit) ကိုနိုပ်ပြီး ဒေါင်းလုဒ်လုပ်ပါ။



ပုံ က/၀၁

ဒေါင်းလုဒ်ပြီးရင် အင်စတော်လာဖိုင်ကို ညာကလစ်နိုပ်ပြီး Run as administrator လုပ်ပါ။ ပုံ (က/၁၃) ကိုကြည့်ပါ။ ပုံ (က/၁၄) မှာလို့ ဒိုင်ယာလော် ဆောက်စ် ပွင့်လာပါမယ်။ အနိဂုင်းထားတဲ့ ချက်ချေ ဆောက်စ်နှစ်ခုကို ချက်ချေလုပ်ပြီး Install Now နိုပ်ပါ။ အင်စတောလ် ပြီးလို့ Setup was successful



Note that Python 3.12.2 cannot be used on Windows 7 or earlier.

- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (ARM64)
- Download Windows installer (32-bit)
- Download Windows installer (64-bit)**
- Download Windows installer (ARM64)

Python 3.12.1 - Dec. 8, 2023

Note that Python 3.12.1 cannot be used on Windows 7 or earlier.

- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (ARM64)
- Download Windows installer (32-bit)
- Download Windows installer (64-bit)

Python 3.13.0a2 - Nov. 21, 2023

- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (ARM64)
- Download Windows installer (32-bit)
- Download Windows installer (64-bit)
- Download Windows installer (ARM64)

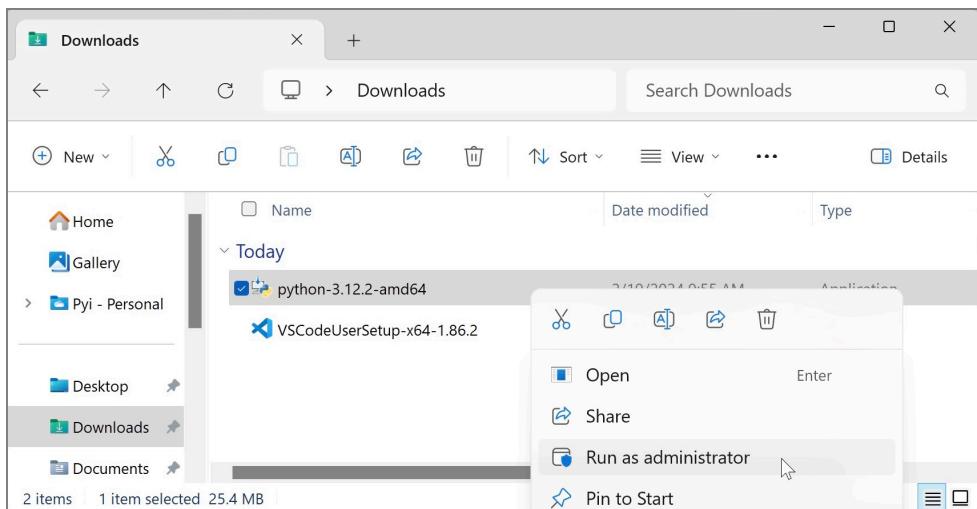
Python 3.13.0a1 - Oct. 13, 2023

- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (ARM64)
- Download Windows installer (32-bit)
- Download Windows installer (64-bit)
- Download Windows installer (ARM64)

ဗိ က/၁၂

ပေါ်လာရင် Close ခလုတ်နိပ်ပြီး ပိတ်ပါ။

ဝင်းဒီး command prompt (cmd) မှာ `python --version` run ရင် ဖုံး (က/၁၃) မှာလို အင် စတော်လုပ်ထားတဲ့ ဗားရှင်းကို ပြပေးသင့်ပါတယ်။



Downloads

Name Date modified Type

python-3.12.2-amd64

VSCodeUserSetup-x64-1.86.2

Open Enter

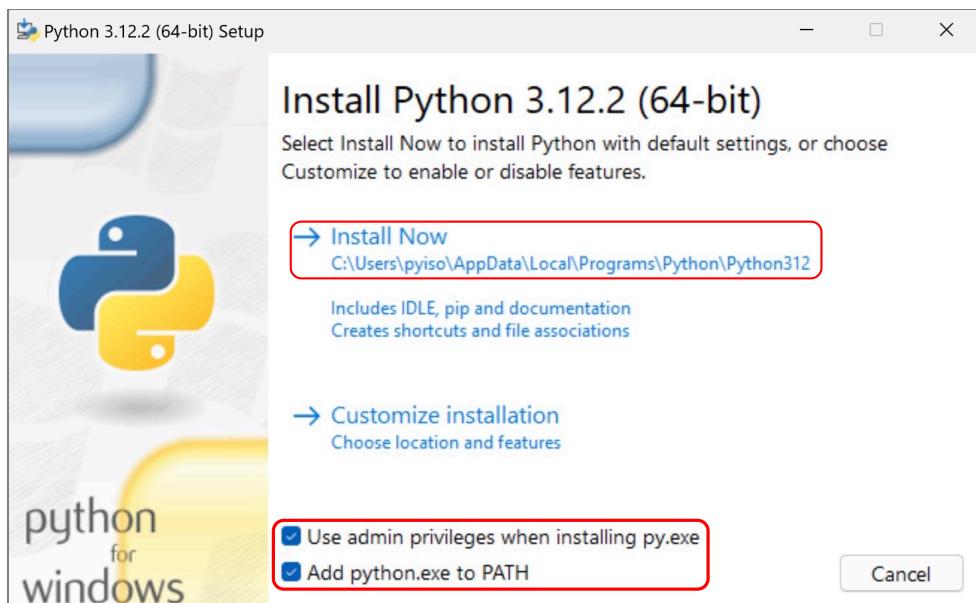
Share

Run as administrator

Pin to Start

ဗိ က/၁၃

JR



ଓ গো

The image shows a Microsoft Windows Command Prompt window. The title bar reads "Command Prompt". The window displays the following text:
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.
C:\Users\pyiso>python --version
Python 3.12.2
C:\Users\pyiso>

ଓ গো

VS Code အင်စတောလ်လုပ်ခြင်း

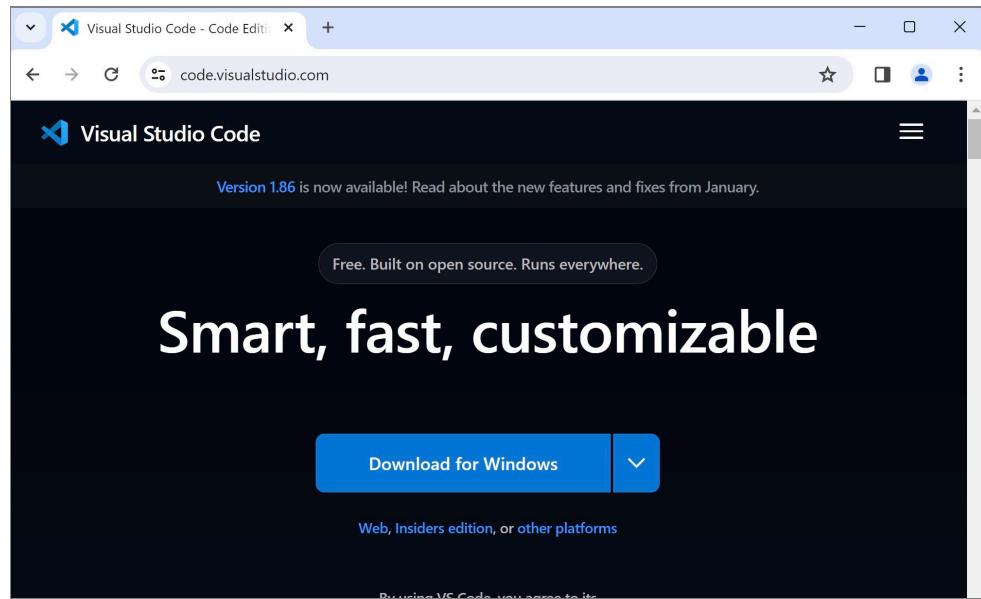
အခုနိရင် Python programming language အောင်မြင်စွာ ထည့်ပြီးသွားပါပြီ။ VS Code အင်စတောလ် ဆက်လုပ်ပါမယ်။ အင်စတောလ် ဒေါင်းလုပ်လုပ်ရန် ဝဘ်စာမျက်နှာကို အောက်ပါလုပ်လုပ်ပါ။

<https://code.visualstudio.com>

မှတစ်ဆင့် သွားပါ။ ပုံ (က/၁၆) ဝဘ်စာမျက်နှာကို တွေ့ရပါမယ်။ [Download for Windows](#) နှိပ်၍ ဒေါင်းလုပ်လုပ်ပါ။

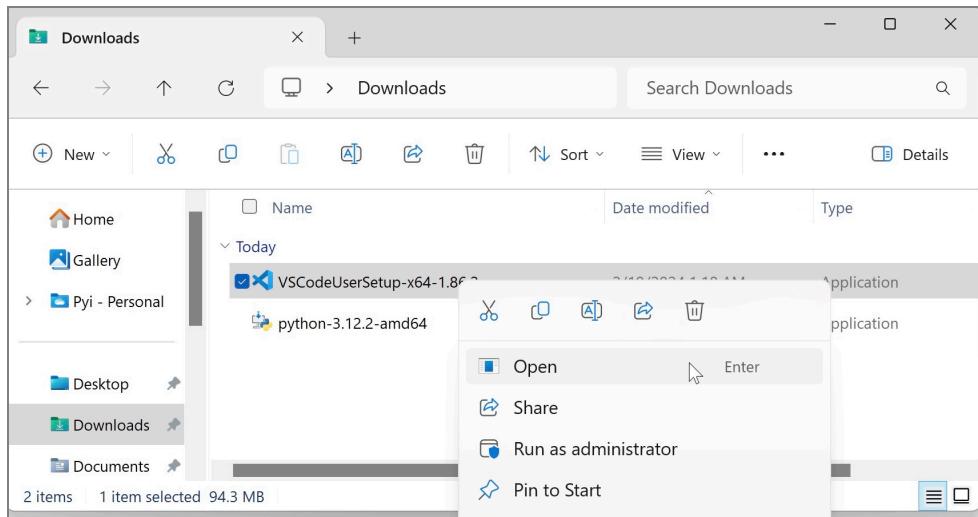
ပြီးတဲ့အခါ အင်စတောလ်လုပ်ကို ညာကလစ်နှုပ်ပြီး ဖွင့်ပါ (ပုံ က/၁၇ မှာ ပြထားပါတယ်)။ အင်စတောလ် ဒိုင်ယာလောက်ဘောက်စွဲ ပွုင့်လာရင် [I accept the agreement](#) ကို ချက်ချွဲလုပ်၍ [Next >](#) တစ်ခု ပြီးတစ်ခု ဆက်နှုပ်သွားပြီး နောက်ဆုံးမှာ [Install](#) နှိပ်ပါ။ အင်စတောလ်လုပ်နေတာကို ခကေတာင့်ပြီး၊ ပြီး သွားရင် [Finish](#) နှိပ်ပါ။

Welcome စခရင်ကို ပုံ (က/၁၈) လို တွေ့ရပါမယ်။ [Dark Modern](#) (သို့) [Light Modern](#) နှစ်သက်ရာ သီးမှစ် ရွေးပါ။ ဒါဆိုရင် VS Code လည်း အင်စတောလ် လုပ်ပြီးသွားပါပြီ။ ကားရဲ့လုပ်ရုံးမှာ ရေးဖို့ ဆက်လုပ်ရပါမယ်။

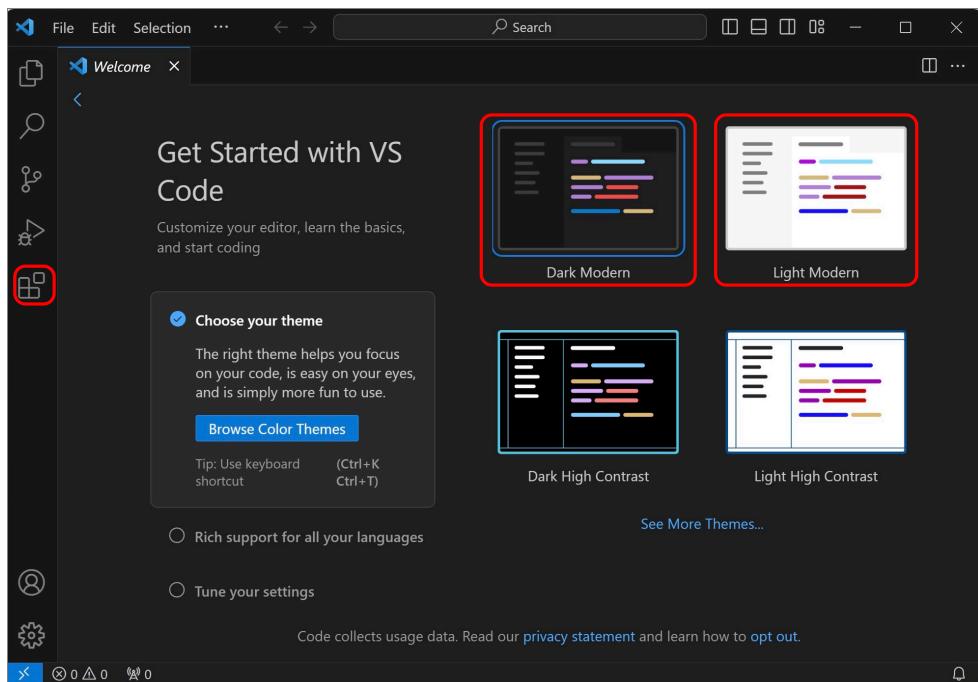


ပုံ က/၁၆

၂၂



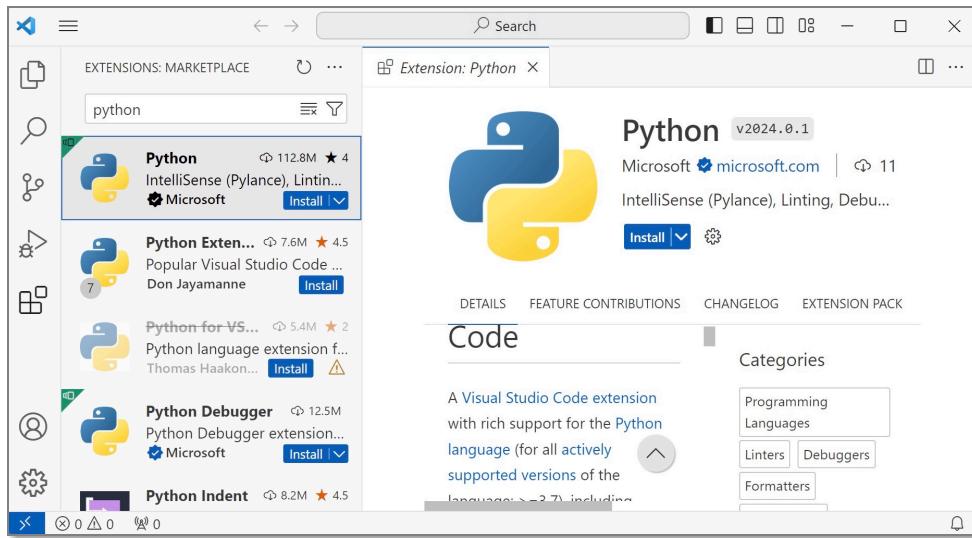
ပုံ ၃/၀၄



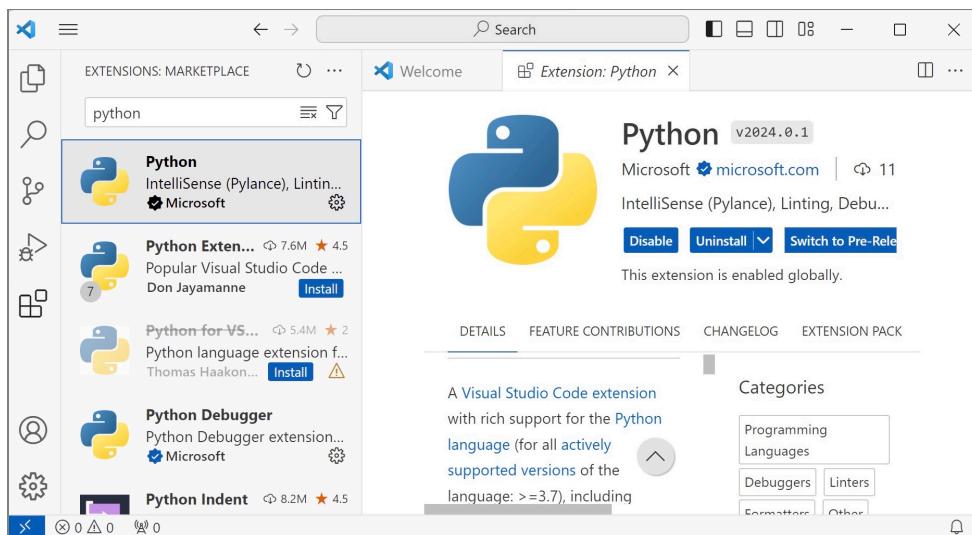
ပုံ ၃/၀၅

VS Code Python Extension တည်ခြင်း

Python ပရိဂုဂ္ဂရေးဖို့အတွက် VS Code က ဒီအတိုင်းဆိုရင် သိပ်အဆင်မပြေသေးပါဘူး။ Python အတွက် extension အင်စတော်လုပ်ပေးရပါအိုးမယ်။ ပုံ (က/၁၈) ဘယ်ဘက်ဘောင်နားမှာ အနီးပိုင်းထားတဲ့ အိုင်ကွန်လေးကို နှိပ်ပါ။ Python extension ကိုရှာပါ။ ပုံ (က/၁၉) မှာ ဘယ်လိုရှာရမလဲ ပြထားပါတယ်။ ပုံမှာတွေ့ရတဲ့ Microsoft က ထုတ်တဲ့ extension ကို အင်စတော်လုပ်ပါ။ အောင်မြင်ရင် ပုံ (က/၂၀) မှာလို ဖြစ်သွားပါမယ်။ Python နဲ့ VS Code ကိစ္စတော့ ပြီးသွားပြီ။ ကားရဲ့လဲ example run ဖို့ ဆက်လပ်ရမယ်။



ပုံ က/၁၉



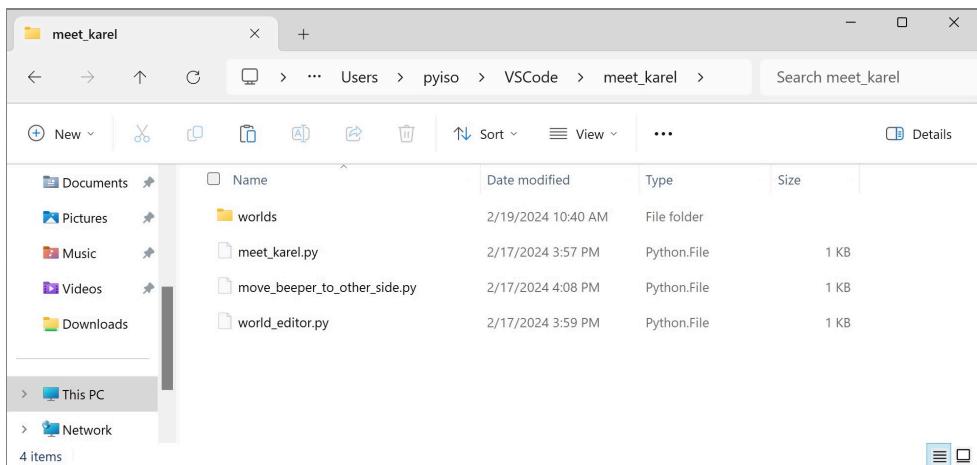
ပုံ က/၂၀

နမူနာ ကားရဲလ် ကဗ္ဗာနှင့် ပရိုဂရမ်ကို ပိုင်များထည့်ခြင်း

meet_karel.zip ဖိုင်ကို ဒီလင့် <http://tinyurl.com/3mm9c7j> ကနေ ဒေါင်းလုပ်လုပ်ပါ။ ငှါး zip ဖိုင်ကို extract လုပ်ပါ။ meet_karel နံမည်နဲ့ ဖို့ဒါတစ်ခု ရလာပါမယ်။ ဖို့ဒါထဲ ဝင်ကြည့်ရင် အောက်ပါ အတိုင်း ရှိသင့်ပါတယ်။

■ worlds

- ▶ meet_karel.w
 - ▶ move_beeper_to_other_side.w
- meet_karel.py
 - move_beeper_to_other_side.py
 - world_editor.py



ပုံ က/ဂ

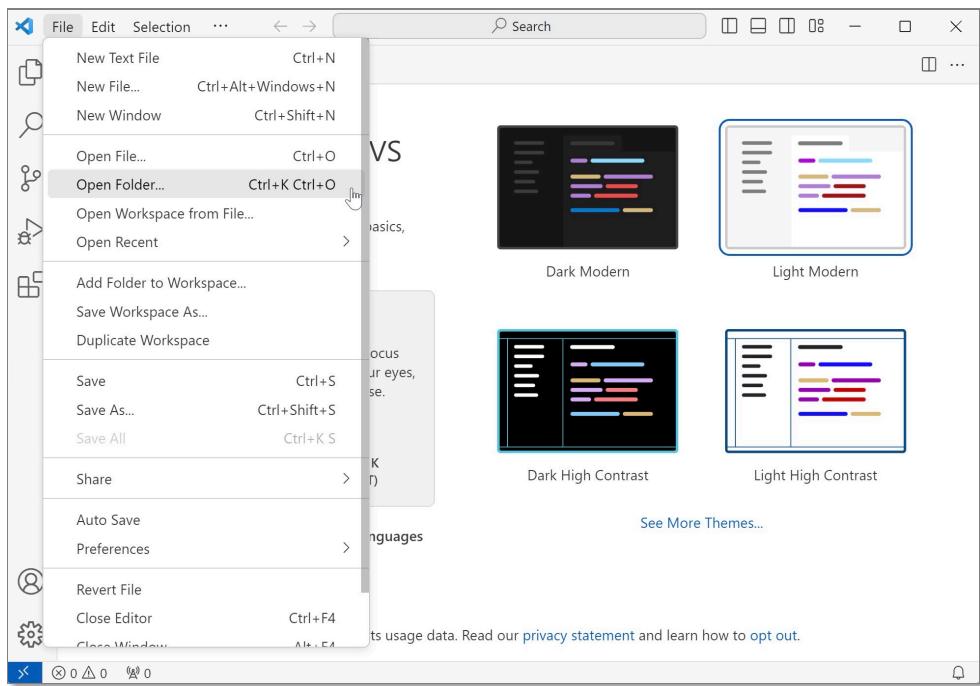
VS Code အတွက် ဖို့ဒါတစ်ခုကို မိမိအတွက် အဆင်ပြေမဲနေရာမှာ သီးသန့်တည်ဆောက် ထားသင့်တယ်။ ဥပမာ

C:\Users\yourname\VS Code

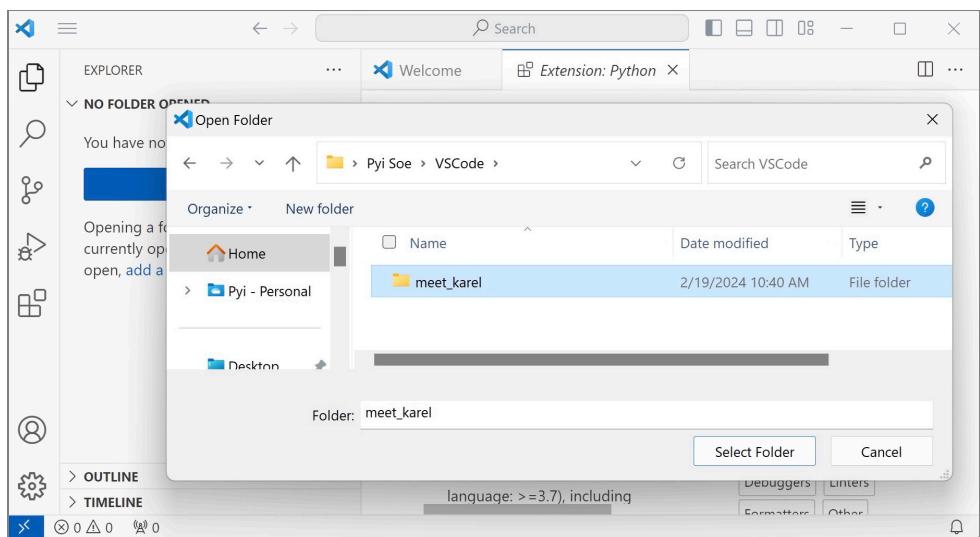
မိမိ လက်ရှိ Home ဖို့ဒါကို C: drive ရဲ့ Users ဖို့ဒါထဲမှာ တွေ့နိုင်ပါတယ်။ Win + R ရေးကတ် နှိပ်ပြီး %userprofile% ရှိက်ထည့်။ Ok လုပ်ပြီး Home ဖို့ဒါကို သွားနိုင်ပါတယ်။ အကယ်၍ မသွားတတ်ရင်လည်း ပြဿနာမရှိပါဘူး။ ကိုယ့်အတွက် လွယ်ကူမဲ့ Desktop, Downloads, Documents တစ်ခုထဲမှာ VS Code အတွက် ဖို့ဒါတစ်ခု ထားလည်းရတယ်။

meet_karel ဖို့ဒါ (.zip ဖိုင်မဟုတ်ပါ) ကို အထက်ပါအတိုင်း အသစ် ဆောက်ထားတဲ့ VS Code သီးသန့်ဖို့ဒါထဲကို ကော်ပီကူးထည့်ပါ။ ငှါး meet_karel ဖို့ဒါကို VS Code File မိုးမှ Open Folder နှိပ်၍ ဖွေ့စွဲပါ။ ပုံ (က/ဂ) တွင်ကြည့်ပါ။

အခန်းအလိုက် နမူနာ ကုဒ်ဖိုင်တွေ ထည့်ပေးထားတဲ့ .zip ဖိုင်တွေကိုလည်း အထက်ပါအတိုင်း လုပ်ရပါမယ်။ .zip ဖိုင်ကို ဖြည့်၊ ရလာတဲ့ ဖို့ဒါကို သီးသန့်ဖို့ဒါတစ်ခုထဲကို ကော်ပီကူးထည့်၊ VS Code နဲ့ အဲ ဒီဖို့ဒါကို ဖွေ့စွဲပါပဲ။



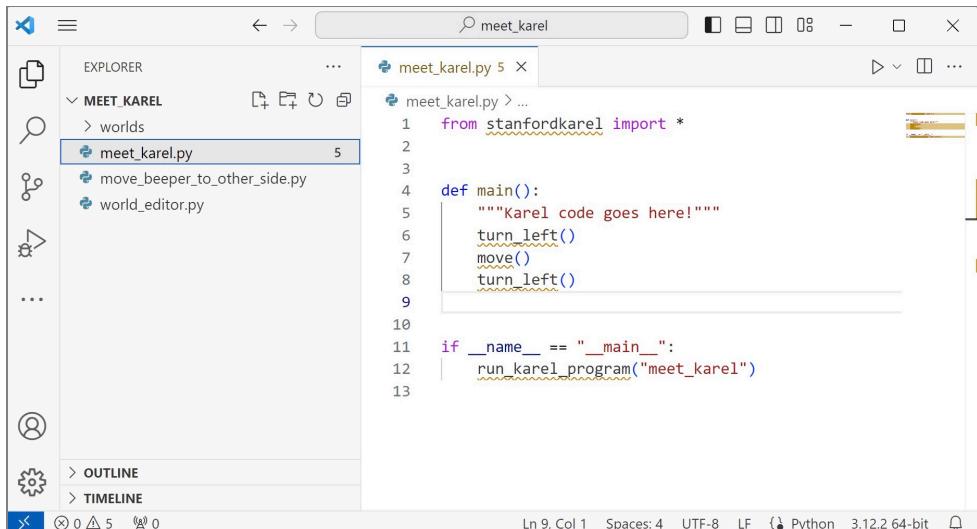
ဗုဒ္ဓန/JJ



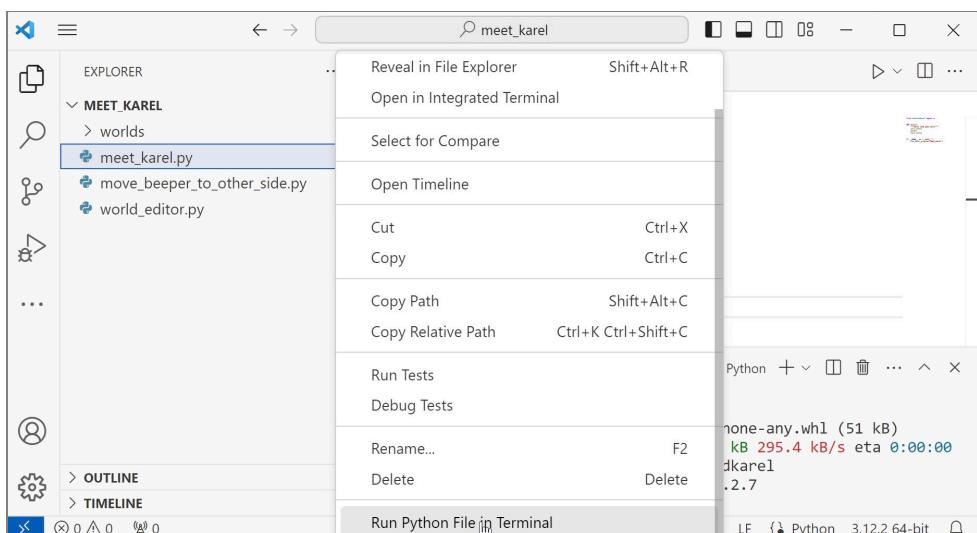
ဗုဒ္ဓန/JR

stanfordkarel လိုက်ဘရီ အင်စတောလ်လုပ်ခြင်း

`meet_karel.py` ဖိုင်ကို ကလစ်နှစ်ချက်နှင့် ဖွင့်ပါ။ ကုဒ်အယ်ဒီတာ ပွင့်လာမယ် (ပုံ ၂/၂၃)။ အဲဒီ ကုဒ် အယ်ဒီတာပေါ် (သို့) `meet_karel.py` ဖိုင်ကို ညာကလစ်နှင့်ပြီး `Run Python File in Terminal` လုပ်ပါ (ပုံ ၂/၂၃)။ Terminal ပွင့်လာပြီး အယ်ရာမက်ဆောင်တွေ ပြလိမ့်မယ်။ ပုံ (၂/၂၄) မှာ ကြည့်ပါ။ ကားရုံးပရှုံးရပ်အတွက် လိုအပ်တဲ့ `stanfordkarel` လိုက်ဘရီ အင်စတောလ် မလုပ်ရသေးပါဘူး။ ဒါ ကြောင့် အယ်ရာဖြစ်နေတာ။



ပုံ ၂/၂၄



ပုံ ၂/၂၅

ခုနကပွင့်လာတဲ့ Terminal မှာပဲ အောက်ပါကွန်မန်းကို run ပြီး `stanfordkarel` လိုက်ဘရီကို အင်စတောလ်လုပ်ပါ။

```

EXPLORER              meet_karel.py 5
MEET_KAREL
  worlds
  meet_karel.py
  move_beeper_to_other_side.py
  world_editor.py

PROBLEMS 5   OUTPUT  TERMINAL  ...
PS C:\Users\pyiso\VSCode\meet_karel> & C:/Users/pyiso/AppData/Local/Programs/Python/Python312/python.exe c:/Users/pyiso/VSCode/meet_karel/meet_karel.py
Traceback (most recent call last):
  File "c:/Users/pyiso/VSCode/meet_karel/meet_karel.py", line 1,
    in <module>
      from stanfordkarel import *
ModuleNotFoundError: No module named 'stanfordkarel'
PS C:\Users\pyiso\VSCode\meet_karel> pip install stanfordkarel

```

ပုံ ၃/၂၆

`pip install stanfordkarel`

ပုံ (၃/၂၆) မှာ အနိဂုံးထားတာကို ကြည့်ပါ။ အဲဒီအတိုင်းရိုက်ထည့်ပြီး Enter ကိုနှိပ်ပါ။ ခက္ကာတဲ့ အခါ အခုလို မက်ဆောင်တွေ ကျလာပါလိမ့်မယ်။

```

ModuleNotFoundError: No module named 'stanfordkarel'
PS C:\Users\pyiso\VSCode\meet_karel> pip install stanfordkarel

```

Collecting stanfordkarel

Downloading stanfordkarel-0.2.7-py3-none-any.whl (51 kB)

51.9/51.9 kB 295.4 kB/s eta 0:00:00

Installing collected packages: stanfordkarel

Successfully installed stanfordkarel-0.2.7

PS C:\Users\pyiso\VSCode\meet_karel>

သိုက်လိုက်ပြထားတဲ့ မက်ဆောင်တွေ တွေ့ရင် အင်စတောလ် အောင်မြင်လိုပါ။ ပုံ (၃/၂၆) အယ်ဒီတာမှာလို သတိပေး လိုင်းတွေနဲ့မျဉ်းတွေ မရှိသင့်တော့ဘူး။ stanfordkarel လိုက်ဘရဲ့ အင်စတောလ် လုပ်ပြီးပြီးလို့ သတိပေးတာတွေ ပျောက်သွားသင့်တယ်။

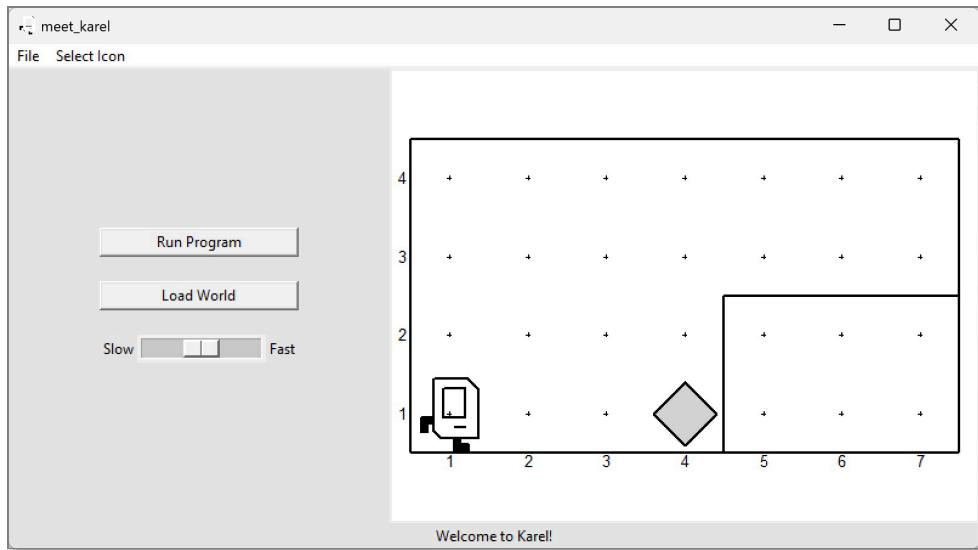
meet_karel.py ဖိုင်ကို ညာကလစ်နှုပ်ပြီး Run Python File in Terminal ပြန်လုပ်ကြည့်ပါ။ ပုံ (၃/၂၇) မှာပြထားတဲ့ ကားရဲလ်ပရိုကရမ် ပွင့်လာသင့်ပါတယ်။ Run Program နှုပ်ကြည့်ပါ။ စက်ရှုပ် လေး ကားရဲလ် နေရာရွှေသွားတာ တွေ့ရမယ်။ meet_karel.py ကို အောက်ပါအတိုင်း ဖြည့်စက်ရေးပါ။

```

from stanfordkarel import *

def main():
    """Karel code goes here!"""
    move()
    move()
    move()
    pick_beeper()

```



ဦး က/ဂျ

```

turn_left()
move()
move()

turn_left()
turn_left()
turn_left()

move()
put_beeper()

if __name__ == "__main__":
    run_karel_program("meet_karel")

```

`meet_karel.py` ဖိုင်ကို ညာကလစ်နိုပ်ပြီး `Run Python File in Terminal` ပြန်လုပ်ကြည့်ပါ။ ကား ရဲလ်ပရိုကရမ် ပွင့်လာရင် `Run Program` နိုပ်ကြည့်ပါ။ ဘိပါလေးကို နေရာရွှေ့ပေးပါလိမ့်မယ်။ အကယ်၍ ကားရဲလ်ပရိုကရမ် မတက်လာရင် ကုဒ်ရေးတာမှားနေလို့ ဖြစ်နိုင်တယ်။ အပေါ်ကုဒ်နဲ့ နှင့်ယူဉ်ပြီး ကြည့်ပါ။ VS Code အယ်ဒီတာမှာ အနိုင်းတွေ့လေးတွေ ပြတဲ့နေရင် အဲဒီနေရာတွေမှာ ဆင်းတက်စုံမှားနေတာ ဖြစ်နိုင်တယ်။

VS Code အယ်ဒီတာမှာ ပရိုကရမ်ကုဒ်ပြင်ပြီး ပြန် run တဲ့အခါ ပထမ run ထားတဲ့ ပရိုကရမ် ကို အရင်ပိတ်ဖို့လိုပါတယ်။ ဆိုလိုတာက `meet_karel.py` ကို run ထားတယ်ဆိုပါတယ့်။ ပုံ (က/ဂျ) က ဝင်းဒီးပွင့်လာမယ်။ `meet_karel.py` ကုဒ်ကို ပြင်ပြီး ပြန် run ချင်ရင် အဲဒီ ဝင်းဒီးကို အရင်ပိတ်ရ မယ်။ မဟုတ်ရင် ပြင်ထားတဲ့ ပရိုကရမ်က ချက်ချင်း ပွင့်မလာဘူး။ ပထမ ဟာကို ပိတ်တော့မှုပဲ နောက် run တဲ့ဟာ ပွင့်လာမှာ။

ဂိုက်ကွင်းကျွန်နေတာက အဖြစ်များတဲ့ အမှားပါ။ ကျွန်ခဲ့လို့ မရပါဘူး။ အင်ဒန်တေးရှင်း (indentation)

tion) လုပ်ရမဲ့နေရာမှာ မလုပ်ထားရင်လည်း ပြဿနာဖြစ်တယ်။ `move`, `turn_left` တွေကို ဘေးမျှပုံးလှာဘက်ဆွဲပြီး အင်ဒန်လုပ်ပေးရမယ်။ အဒါတွေ ဂရာမစိုက်မိရင် ဆင်းတက်စ်အမှားဖြစ်ပြီး ပရိုကရမ် `run` လို့ မရနိုင်ဘူး။

Terminal မှာ ထုတ်ပေးတဲ့ မက်ဆွဲချုပ်တွေကို ကြည့်ပြီးတော့လည်း ဘာပြဿနာဖြစ်နေလဲ မှန်းဆလို့ရနိုင်တယ်။ ဘာကြောင့်ဖြစ်နိုင်လဲ ဆက်စပ်စဉ်းစားလို့ ရတယ်။ ဥပမာ ဖြစ်တဲ့ပြဿနာအလိုက် အခုလို တွေ့ရပါမယ်။

```
File "c:\Users\pyiso\VS Code\meet_karel\meet_karel.py", line 6
    move(
        ^
SyntaxError: '(' was never closed

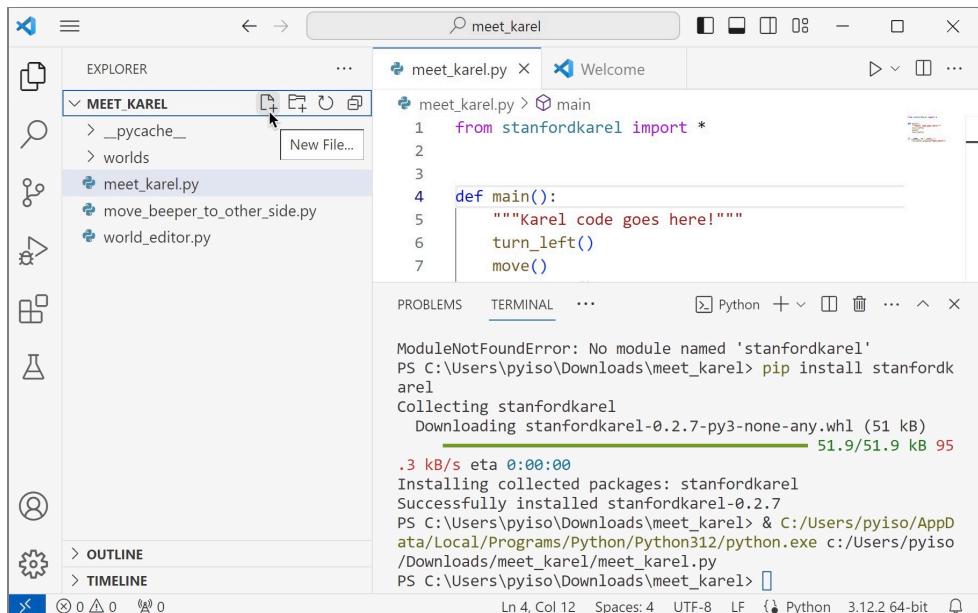
File "c:\Users\pyiso\VS Code\meet_karel\meet_karel.py", line 7
    move()
        ^
IndentationError: unexpected indent

Traceback (most recent call last):
  File "c:\Users\pyiso\VS Code\meet_karel\meet_karel.py", line 1,
    in <module>
      from stanfordkarel import *
ModuleNotFoundError: No module named 'stanfordkarel'
```

VS Code တွင် Python ဖိုင် အသစ်ယူခြင်း

MEET_KAREL ပင်မ ပရောဂျက်ဖိုဒ်အပေါ်မှာ ကလစ်နိုင်ပါ။ ပုံမှာ ပြထားတဲ့ **New File** အင်ကွန်ကို့နှင့် ဖိုင်နံပည်ဖြည့်တဲ့ ဘောက်စံလေး ပေါ်လာမယ်။ Python ဖိုင်တွေက .py အိပ်စံတန်းရှင်းနဲ့ ဖြစ်ရပါမယ်။ ဒါကြောင့် နံပည် ဖြည့်တဲ့အခါ .py နဲ့ အဆုံးသတ်ပေးရပါမယ် (ဥပမာ hello.py)။ ကားချုလ်ပရိုဂရမ်တစ်ခုကို Python ဖိုင်တစ်ခု ထားပါမယ်။ ပင်မ ပရောဂျက်ဖိုဒ်အပေါ်မှာပဲ တိုက်ရှိကြရပါမယ်။

နောက်ပိုင်း အဆင့်မြင့်လာရင် ပရိုဂရမ်တစ်ခုအတွက် ပရောဂျက်တစ်ခု ထားနိုင်တယ်။ ကုဒ်ဖိုင်တွေ အပြင် ပရိုဂရမ်အတွက် လိုအပ်တဲ့ ရုပ်ပုံတွေ၊ အခြားဖိုင်တွေ (config ဖိုင်၊ setting ဖိုင် စသည်ဖြင့်) လည်း ပါနိုင်တယ်။ ပင်မပရောဂျက် အောက်မှာပဲ ဖိုင်တွေက တိုက်ရှိကြရဖို့လည်း မလိုတော့ဘူး။ ဆက်စပ်ရာ ဖိုင်တွေကို အမျိုးအစားအလိုက် ဖန်ရှင်အလိုက် ဖို့ဒါတွေခဲ့ပြီး စနစ်ကျ စီစဉ်ဖွဲ့စည်း ထားရမှာပါ။ ပရောဂျက်တစ်ခုမှာ ဖိုင်တွေကို စနစ်တကျ စွဲထားဖို့ အရေးကြီးပါတယ်။



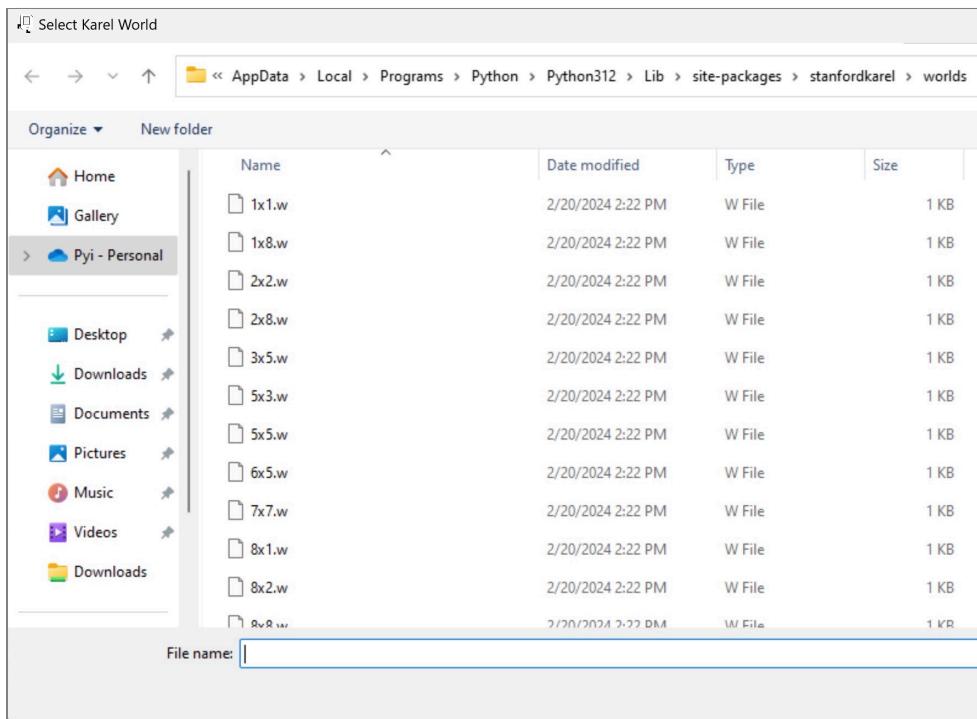
ပုံ က/၂

နောက်ဆက်တဲ့ ခ

ကားရဲလ်ပရီဂရမ် ဖီချာများ

ကားရဲလ် ကဗ္ဗာဖိုင်များ

ကားရဲလ်ပရီဂရမ် ဝင်းခွဲးမှာ **Load World** ခလုတ်နှုပ်ပြီး ကဗ္ဗာဖိုင်အသစ် တင်လိုရတယ်။ ခလုတ်နှုပ်လိုက် ရင် အခုလို ဖိုင် ခိုင်ယာလော့ဂုံးပုံင့်လာမယ်။



ပုံ ၉/၁

ဒါက stanfordkarel လိုက်ဘရဲ သူနိုင်အရိုအတိုင်း ပါတဲ့ worlds ဖို့ပါ။ ဖိုင်တွေက **.w** နဲ့ ဆုံးပါတယ်။ ကဗ္ဗာဖိုင်တွေကို ပင်မ ပရောဂျက်အောက် worlds ဖို့ပါထဲမှာ ထားလိုလည်းရတယ်။ အခြားနေရာတွေမှာ ထားလိုတော့ မရဘူး။

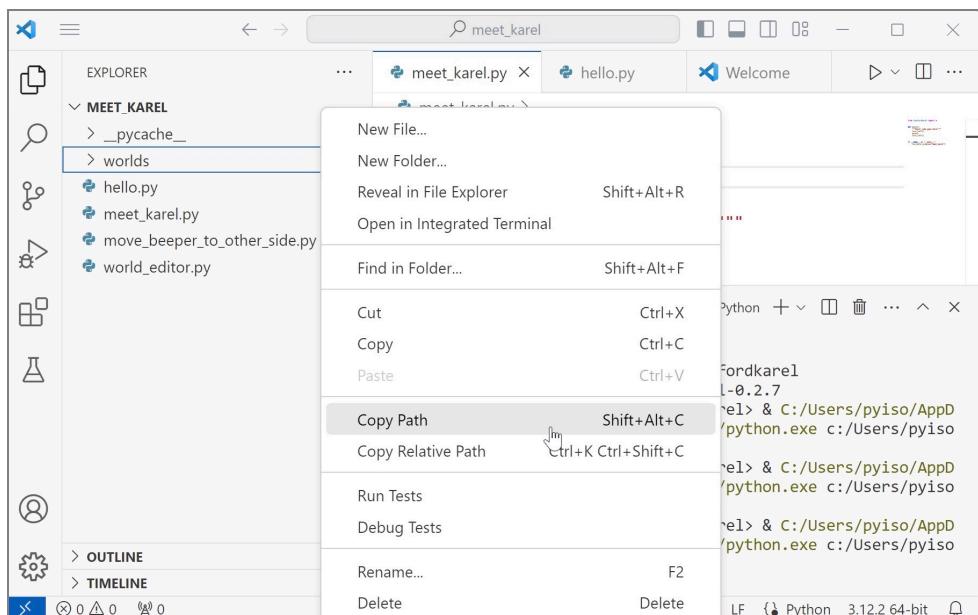
စာအုပ်ပါ ဥပမာတွေ၊ လေ့ကျင့်ခန်းတွေအတွက် ကမ္မာဖိုင်တွေကို ပရောဂျက် တစ်ခုချင်းအလိုက် သီးခြား worlds ဖို့ဒါထဲမှာ ထည့်ပေးထားမှာပါ။ ပရိုဂ်ရမ်တစ်ခုဟာ ကမ္မာတစ်ခုတည်းမှာပဲ အလုပ်လုပ်တာမဟုတ်ဘဲ အလားတူ ကမ္မာအမျိုးမျိုးအတွက် အလုပ်လုပ်အောင် ရေးပေးရတာ။ အခြားတာကို နားမလည်ရင် အခန်း (၂) ဖတ်ပြီးရင် နားလည်သွားမှာပါ။

Load World လုပ်တဲ့အခါ ပွင့်လာတဲ့ ဖိုင် ခိုင်ယာလော်ကာ ကိုယ်လိုချင်တဲ့ worlds ဖို့ဒါ မဖြစ်နေဘူး။ သူနိုင်ပါတဲ့ worlds ဖို့ဒါ ဖြစ်နေတယ်။ ကိုယ်ခေါ်တင်ချင်တဲ့ ဖိုင်တွေရှိတဲ့ လက်ရှိပရောဂျက်၏ worlds ဖို့ဒါကို သွားရမယ်။ ဥပမာ PyCharm/VS Code အတွက် MeetKarel/meet_karel ပရောဂျက် worlds ဖို့ဒါ လမ်းကြောင်း အပြည့်အစုံက

C:\Users\yourname\VSCode\meet_karel\worlds
C:\Users\yourname\PycharmProjects\MeetKarel\worlds

ဖြစ်ပါမယ်။ ဖိုင်ခိုင်ယာလော်ကနေ ဖော်ပြပါ လက်ရှိပရောဂျက် worlds ဖို့ဒါကို တစ်ဆင့်ချင်း သွားပြီး တင်ချင်တဲ့ ကမ္မာဖိုင် (.w ဖိုင်) ကို ရွေးရမှာပါ။

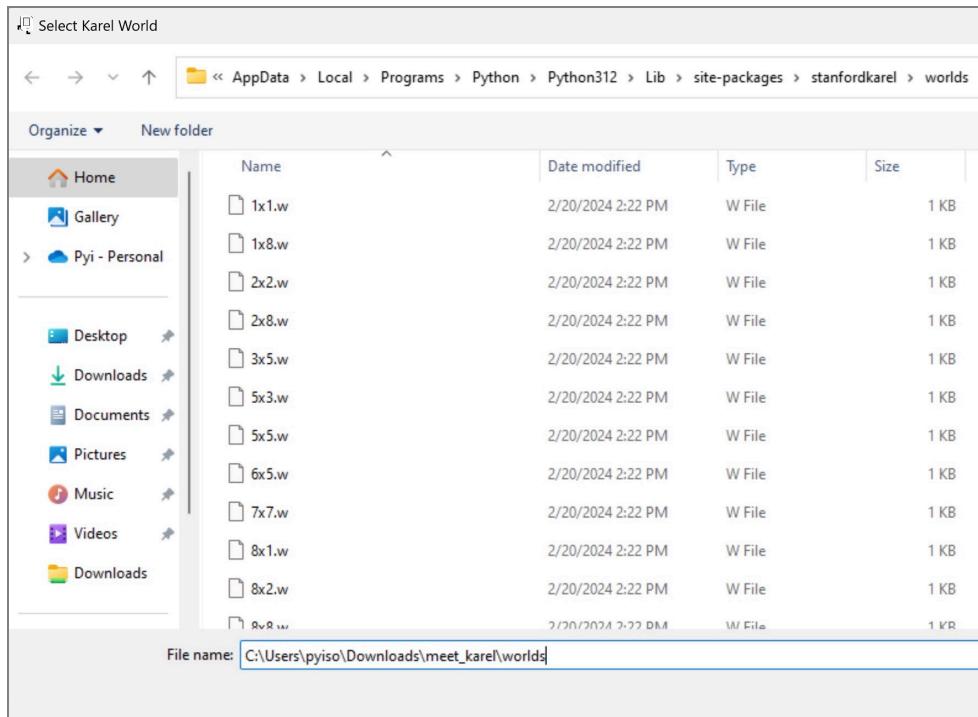
အပေါ်ကနည်းနဲ့ အဆင်မပြောရင် အခုလုံစမ်းကြည့်ပါ။ လက်ရှိပရောဂျက် worlds ဖို့ဒါကို ညာကလစ်နှုပ်ပြီး Copy Path လုပ်ပါ (ပုံ ၁/၂)။ ဖိုင် ခိုင်ယာလော် **File name** မှာ ကူးထည့်ပါ (ပုံ ၂/၂)။ **Enter** ကိုနှိပ်ပါ။ ပရောဂျက် worlds ဖို့ဒါကိုရောက်သွားပါမယ်။ လိုတဲ့ကမ္မာဖိုင် ရွေးတင်ရုံပါပဲ။ ပုံ (၁/၂) မှာ meet_karel worlds ကို နူးနှုန်းပြထားပါတယ်။



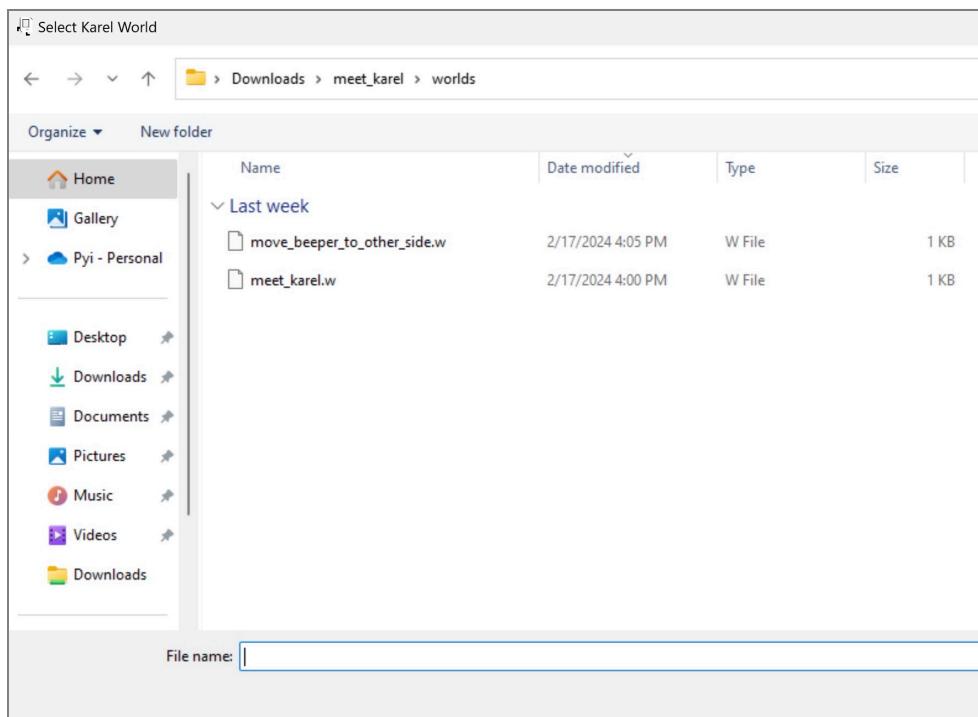
ပုံ ၁/၂

အကယ်၍ အထက်ဖော်ပြပါနည်းတွေက ရှုပ်နေတယ်ထင်ရင် မှတ်ရ/သွားရ လွယ်တဲ့ Desktop, Downloads, Documents လို့ နေရာတစ်ခုခုမှာ သီးသန့်ဖို့ဒါတစ်ခု အောက်ပြီး ပရောဂျက်အားလုံး ထည့်ထားတာ အရှင်းဆုံးပါပဲ။ ပရောဂျက်ဖို့ဒါနေရာ သိရင် ဖိုင်ခိုင်ယာလော်ကနေ ဘယ်လိုမဆိုရောက်အောင် သွားလို့ရပါတယ်။

၃၃



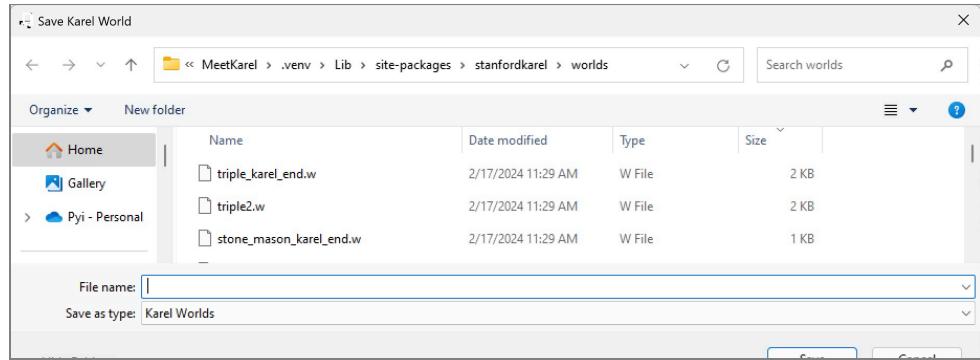
ပုံ ၅/၃



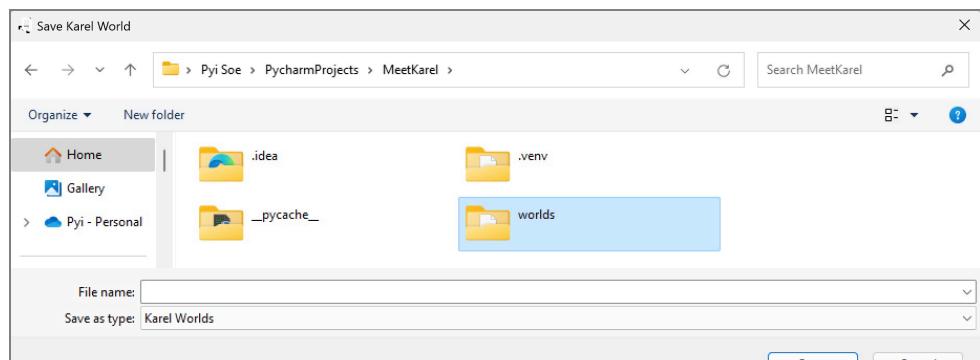
ပုံ ၅/၄

ကိုယ်ပိုင် ကားရဲလ်ကမ္ဘာ ဆောက်ခြင်း

ကားရဲလ်ရဲ ကမ္ဘာ အသစ်တစ်ခုဆောက်မယ်ဆိုရင် world_editor.py ဖိုင်ကို ညာကလစ်နှင့် Run ပါ။ Would you like to load an existing world? လို ပေါ်လာပြီး Yes/No ရွှေ့ခိုင်ပါလိမ့်မယ်။ No ကို နှိပ်ပါ။ ကမ္ဘာအရွယ်အစားအတွက် ကော်လံ ဘယ်နှစ်ခုလဲ row ဘယ်နှစ်ခုလဲ ထည့်ပေးပါ။ ကိုယ်ပိုင် ကားရဲလ်ကမ္ဘာ တည်ဆောက်လိုရတဲ့ ဝင်းဒီးပွင့်လာပါလိမ့်မယ်။ ကားရဲလ် မျက်နှာမူရာအရပ်၊ ဘိပါအိတ်ထဲရှိ ဘိပါအရေအတွက်၊ နံရုံဆောက်/ဖျက်တာ၊ ဘိပါထည့်/ဖယ်ထုတ်တာ စတာတွေ လုပ်နိုင်ပါတယ်။ Save World နှိပ်ပြီး သိမ်းနိုင်ပါတယ် ဖိုင်ကိုသိမ်းတဲ့အခါ သုန္တရှိ သိမ်းခိုင်းတဲ့ဖို့ (default world folder) ထဲမှာ သို့မဟုတ် ပင်မ ပရောဂျက်ဖို့တဲ့က worlds ဖို့ဒါထဲမှာ သိမ်းရပါမယ်။



ပုံ ၉/၅



ပုံ ၉/၆

