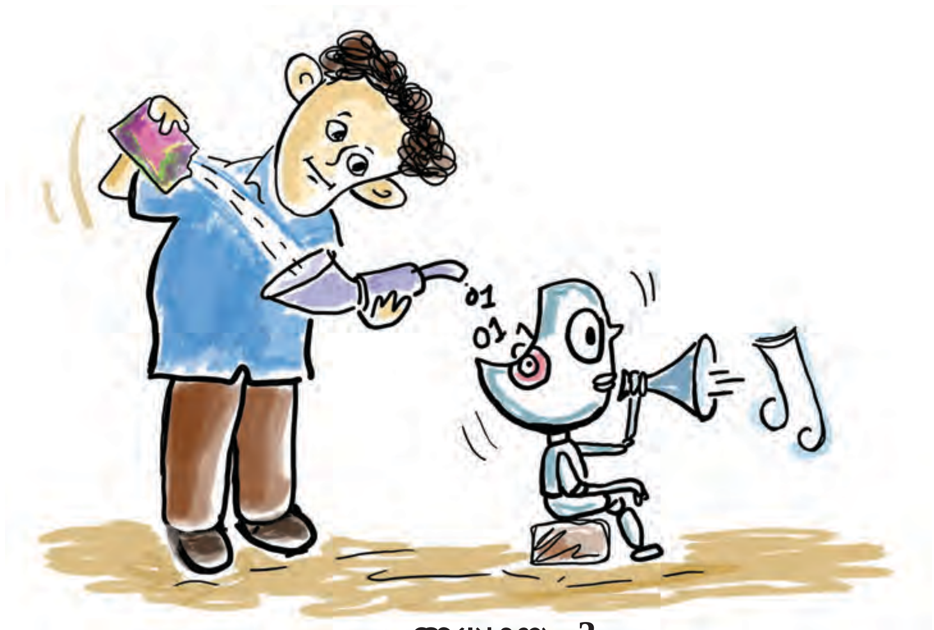
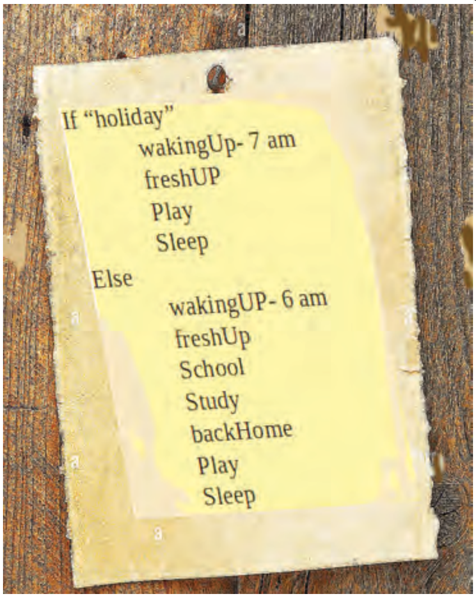


അധ്യായം 3

കമ്പ്യൂട്ടർഭാഷ

ചിത്രം 3.1 നോക്കി അവധിദിവസങ്ങളിലെ സന്ദീപിന്റെ ദിനചര്യയിലുള്ള വ്യത്യാസം കണ്ടെത്താമോ?


If "holiday"
wakingUp- 7 am
freshUP
Play
Sleep
Else
wakingUP- 6 am
freshUp
School
Study
backHome
Play
Sleep

ചിത്രം 3.1 ദിനചര്യയുടെ ഷെഡ്യൂൾ

എന്തെല്ലാമാണ് കണ്ടെത്തിയത്?

സാധാരണദിവസങ്ങളിൽ ചെയ്യുന്ന ചില കാര്യങ്ങൾ അവധിദിവസങ്ങളിൽ ചെയ്യാറില്ല, അല്ലേ?

പരിചയിച്ച ശീലങ്ങളുടെയും ലഭിച്ച നിർദേശങ്ങളുടെയും നേടിയ അറിവുകളുടെയും അടിസ്ഥാനത്തിലാണ് സന്ദീപിന്റെ ഓരോ ദിവസവും മുന്നോട്ടു പോവുന്നത്.

നാമും ഇതുപോലെത്തന്നെയല്ലേ?

അനുഭവങ്ങളിൽനിന്നും പഠനങ്ങളിൽനിന്നും നേടിയ അറിവുകളും ശീലങ്ങളും, മറ്റുള്ളവരിൽനിന്ന് ലഭിച്ച

നിർദേശങ്ങളുമൊക്കെയാണ് ഓരോ ദിവസവും നമ്മെ മുന്നോട്ടു നയിച്ചുകൊണ്ടിരിക്കുന്നത്.

കമ്പ്യൂട്ടറുകൾ പ്രവർത്തിക്കുന്നതും ഇങ്ങനെ നൽകപ്പെട്ട നിർദേശങ്ങളും അതിലൂടെ നേടിയെടുത്ത അറിവുകളും കൊണ്ടാണെന്ന് നിങ്ങൾ മനസ്സിലാക്കിയിട്ടുണ്ടല്ലോ.

എങ്ങനെയാണ് കമ്പ്യൂട്ടറുകൾ നാം നൽകുന്ന നിർദേശങ്ങൾ മനസ്സിലാക്കുന്നത്?

കമ്പ്യൂട്ടർ ഒരു ഇലക്ട്രോണിക് ഉപകരണമാണല്ലോ. ഏതൊരു ഇലക്ട്രോണിക് ഉപകരണവുംപോലെ കമ്പ്യൂട്ടറുകൾക്കും ON, OFF എന്നീ അവസ്ഥകൾ മാത്രമേ മനസ്സിലാവുകയുള്ളൂ.

ON, OFF എന്നീ അവസ്ഥകളെ പ്രതിനിധീകരിക്കുന്ന 0, 1 എന്നീ ചിഹ്നങ്ങൾ മാത്രമുപയോഗിക്കുന്ന ***ബൈനറി ഭാഷ***യിലുള്ള കോഡുകളിലൂടെയാണ് കമ്പ്യൂട്ടറുകൾക്ക് നിർദേശങ്ങൾ നൽകേണ്ടത്.

പക്ഷേ, നമുക്ക് ബൈനറിഭാഷ വശമില്ലല്ലോ. പിന്നെ എങ്ങനെയാണ് കമ്പ്യൂട്ടറിന് ബൈനറിഭാഷയിൽ നിർദേശങ്ങൾ നൽകുക?

ഉദാഹരണത്തിന്, നിങ്ങൾക്ക് ബംഗാളിഭാഷ മാത്രമറിയാവുന്ന ഒരു സുഹൃത്ത് ഉണ്ടെന്നിരിക്കട്ടെ. നിങ്ങൾക്ക് ബംഗാളിഭാഷ അറിയുകയുമില്ല. എങ്ങനെയാണ് സുഹൃത്തുമായി ആശയവിനിമയം നടത്തുക?

ബംഗാളിയും മലയാളവും അറിയാവുന്ന ഒരു പൊതുസുഹൃത്തുണ്ടെങ്കിൽ ഈ പ്രശ്നം പരിഹരിക്കാമല്ലോ.

ഇത്തരത്തിൽ നമുക്ക് മനസ്സിലാവുന്ന ഭാഷയിൽ നൽകിയ നിർദേശങ്ങളെ ബൈനറിഭാഷയിലേക്കു മൊഴിമാറ്റാൻ കഴിയുന്ന പ്രോഗ്രാമുകൾ നമ്മുടെ കമ്പ്യൂട്ടറുകളിലുമുണ്ട്. ഹൈലെവൽ ഭാഷകൾ എന്നും ട്രാൻസ്‌ലേറ്റർ പ്രോഗ്രാമുകളെന്നും ഇവ അറിയപ്പെടുന്നു.

ട്രാൻസ്‌ലേറ്റർ പ്രോഗ്രാമുകൾ പ്രധാനമായും രണ്ടുതരത്തിലുണ്ട് - ***കമ്പയിലറുകളും ഇന്റർപ്രെട്ടറുകളും.***

ട്രാൻസ്‌ലേറ്റർ പ്രോഗ്രാമുകളുടെ സഹായത്തോടെ നമുക്ക് മനസ്സിലാവുന്ന High level ഭാഷയിലെഴുതിയ നിർദേശങ്ങൾ കമ്പ്യൂട്ടറുകൾ മനസ്സിലാക്കുകയും അതിനനുസരിച്ച് പ്രവർത്തിക്കുകയുമാണ് ചെയ്യുന്നത്.

ബൈനറിഭാഷ

ON, OFF എന്നീ സങ്കേതങ്ങൾ മാത്രം ഉപയോഗിക്കുന്ന ഭാഷയെ ബൈനറിഭാഷ എന്നാണ് പറയുക.

ഈ അവസ്ഥകളെ HIGH, LOW അല്ലെങ്കിൽ 1, 0 അല്ലെങ്കിൽ TRUE, FALSE എന്നിങ്ങനെയെല്ലാം സൂചിപ്പിക്കാറുണ്ട്.

കമ്പയിലറുകളും ഇന്റർപ്രെട്ടറുകളും

നിർദേശങ്ങൾ പൂർണ്ണമായും ബൈനറിഭാഷയിലേക്ക് മൊഴിമാറ്റി ഒരു പ്രത്യേക ഫയലാക്കി നൽകുന്നവയാണ് കമ്പയിലറുകൾ. എന്നാൽ നിർദേശങ്ങൾ പ്രവർത്തിപ്പിക്കുന്ന ഘട്ടത്തിൽ മാത്രം ഓരോ വരിയും മൊഴിമാറ്റം ചെയ്ത് നൽകുന്നവയാണ് ഇന്റർപ്രെട്ടറുകൾ.

കമ്പയിലറുകൾ നൽകുന്ന ബൈനറി ഫയലുകൾ (Executable files) പിന്നീട് പ്രവർത്തിപ്പിക്കുന്നതിന് അവയുടെ ഹൈലെവൽ ഭാഷയിലെഴുതിയ നിർദേശങ്ങൾ (Source Code) ആവശ്യമില്ല. എന്നാൽ ഇന്റർപ്രെട്ടറുകൾ ഉപയോഗിച്ച് പ്രവർത്തിപ്പിക്കുമ്പോൾ Source Code കമ്പ്യൂട്ടറിൽ ആവശ്യമാണ്.

C, C++, ജാവ മുതലായ ഹൈലവൽ പ്രോഗ്രാമിങ് ഭാഷകൾ കമ്പയിലറുകൾ ഉപയോഗിച്ചാണ് ബൈനറിയിലേക്ക് മൊഴിമാറ്റം ചെയ്തു പ്രവർത്തിപ്പിക്കുന്നത്. പൈത്തൺ ഒരു Interpreted പ്രോഗ്രാമിങ് ഭാഷയാണ്.

ഇത്തരത്തിലുള്ള ഒരു ഹൈലെവൽ ഭാഷയാണ് ഈ പാഠഭാഗത്ത് നാം പരിചയപ്പെടുന്നത്.

പാഠഭാഗത്തിന്റെ തുടക്കത്തിൽ നൽകിയിരിക്കുന്ന സന്ദീപിന്റെ ദിനചര്യകൾ ശ്രദ്ധിച്ചല്ലോ. ഓരോ ദിവസത്തിനുമനുസരിച്ച് ലഭിച്ച നിർദേശങ്ങൾ ഓരോന്നായി ചെയ്തുവരുകയാണ് സന്ദീപ് ചെയ്യുന്നത്.

ഇതുപോലെ ഓരോ പ്രശ്നവും പരിഹരിക്കാനുള്ള കൃത്യമായ നിർദേശങ്ങൾ കമ്പ്യൂട്ടറുകൾക്ക് നൽകി ഇവയെ പ്രശ്നനിർധാരണത്തിനായി ഉപയോഗിക്കാം. ഇത്തരത്തിൽ, കമ്പ്യൂട്ടറുകൾക്ക് നിശ്ചിത ജോലി (Task) ചെയ്യുന്നതിനാവശ്യമായ നിർദേശങ്ങൾ നൽകുന്നതിനെ ***പ്രോഗ്രാമിങ്*** എന്നു പറയുന്നു.

ഉദാഹരണമായി, ഒരു വ്യക്തിയുടെ പ്രായം കണക്കാക്കുന്നതിനുള്ള പ്രോഗ്രാം തയ്യാറാക്കാനാവശ്യമായ നിർദേശങ്ങൾ കമ്പ്യൂട്ടറിനു നൽകണമെന്നിരിക്കട്ടെ.

എന്തെല്ലാം വിവരങ്ങൾ ലഭിച്ചാലാണ് പ്രായം കണ്ടെത്താൻ കഴിയുക?

- ഏതു വർഷമാണ് ജനിച്ചത് എന്ന് അറിയണം.
- ....

ജനിച്ച വർഷം 2011 ആയ ഒരാളുടെ പ്രായം എത്രയായിരിക്കും?

2025 ആണ് നിലവിലെ വർഷമെങ്കിൽ അയാളുടെ വയസ്സ് 14 ആയിരിക്കും, അല്ലേ. ഇത് എങ്ങനെയാണ് കണ്ടെത്തിയത്?

നിലവിലെ വർഷത്തിൽനിന്നു ജനിച്ച വർഷം കുറച്ചാൽ മതിയല്ലോ.

അതായത്,

പ്രായം = നിലവിലെ വർഷം - ജനിച്ച വർഷം

മേൽ സൂചിപ്പിച്ച കാര്യങ്ങൾ ഘട്ടംഘട്ടമായി എഴുതുകയാണെങ്കിൽ,

- നിലവിലെ വർഷം (current_year)=2025
- ജനിച്ച വർഷം (birth_year)=2011
- പ്രായം = നിലവിലെ വർഷം - ജനിച്ച വർഷം

  (age=current_year - birth_year)
- പ്രായം കമ്പ്യൂട്ടറിൽ പ്രദർശിപ്പിക്കുക.

ഇത്തരത്തിൽ പ്രശ്നപരിഹാരത്തിനുള്ള ഘട്ടങ്ങൾ ക്രമത്തിൽ എഴുതുന്നതിനെ ***അൽഗോരിതം*** എന്നു പറയുന്നു.

മേൽപ്പറഞ്ഞ നിർദേശങ്ങൾ അനുസരിച്ച് ഒരു വ്യക്തിയുടെ പ്രായം കണക്കാക്കുന്നതിനുള്ള ഒരു കമ്പ്യൂട്ടർപ്രോഗ്രാം തയ്യാറാക്കിയാലോ.

ഏതു കമ്പ്യൂട്ടർഭാഷയാണ് ഈ പ്രോഗ്രാം തയ്യാറാക്കാൻ നാം തിരഞ്ഞെടുക്കേണ്ടത്?

നിരവധി ഭാഷകൾ കമ്പ്യൂട്ടർ പ്രോഗ്രാമുകൾ തയ്യാറാക്കുന്നതിനായി ഇന്ന് ലഭ്യമാണ്.

ഏതെല്ലാം കമ്പ്യൂട്ടർഭാഷകളെക്കുറിച്ച് നിങ്ങൾ കേട്ടിട്ടുണ്ട്?

- ....
- ....
- ....
- ....
- ....

ജാവ, C, C++, പൈത്തൺ, Ruby, PHP തുടങ്ങി അനേകം പ്രോഗ്രാമിങ്ഭാഷകൾ നിലവിലുണ്ട്. പൈത്തൺ ഭാഷ ഉപയോഗിച്ചാണ് നാമിവിടെ പ്രോഗ്രാമുകൾ എഴുതുന്നത്.

പൈത്തൺ

പ്രോഗ്രാമിങ് മേഖലയിലേക്ക് കടന്നുവരാൻ ആഗ്രഹിക്കുന്ന ചെറിയ കുട്ടികൾക്കു മുതൽ നിർമ്മിതബുദ്ധി സങ്കേതങ്ങളും റോബോട്ടിക്സ്, ഡാറ്റാ സയൻസ് പോലെയുള്ള സങ്കീർണ്ണമായ സംവിധാനങ്ങൾ വികസിപ്പിച്ചെടുക്കുന്നവർക്കുവരെ യോജിച്ച ഒരു പ്രോഗ്രാമിങ്ഭാഷയാണ് പൈത്തൺ. വളരെ ലളിതമായ Syntax ആണ് പൈത്തണിനുള്ളത്.

ചിത്രം 3.2 Guido van Rossum

ചിത്രം 3.3 പൈത്തൺ ലോഗോ

1990 ൽ ഗൈഡോ വാൻ റോസ്സം (ചിത്രം 3.2) എന്ന നെതർലാൻഡുകാരനായ കമ്പ്യൂട്ടർ എൻജിനീയറാണ് പൈത്തൺഭാഷ വികസിപ്പിച്ചെടുത്തത്. ഒരു ഓപ്പൺസോഴ്സ് പ്രോഗ്രാമിങ് ഭാഷയാണ് പൈത്തൺ.

പൈത്തൺ പ്രോഗ്രാം തയ്യാറാക്കാം

നേരത്തേ നാം സൂചിപ്പിച്ച പ്രായം കണക്കാക്കുന്നതിനുള്ള പ്രോഗ്രാം പൈത്തൺഭാഷയിൽ തയ്യാറാക്കുന്നതെങ്ങനെയെന്ന് നോക്കാം.

ഇതിനായി നമ്മുടെ കമ്പ്യൂട്ടറിലുള്ള Text Editor എന്ന ആപ്ലിക്കേഷനാണ് ഉപയോഗിക്കേണ്ടത്. തയ്യാറാക്കിയ പ്രോഗ്രാം ബൈനറിഭാഷയിലേക്കു മൊഴിമാറ്റുന്നതിന് പൈത്തൺഭാഷയുടെ ഇന്റർപ്രെട്ടറും ആവശ്യമാണ്.

ചുവടെ നൽകിയിരിക്കുന്ന പ്രോഗ്രാം നോക്കൂ. നാം നേരത്തേ, അൽഗോരിതമായി എഴുതിയ കാര്യങ്ങൾ പൈത്തൺഭാഷയിലേക്കു മാറ്റിയിരിക്കുകയാണിവിടെ ചെയ്തിരിക്കുന്നത്. ഈ പ്രോഗ്രാം, കമ്പ്യൂട്ടറിലുള്ള Text Editor എന്ന ആപ്ലിക്കേഷൻ തുറന്ന് തെറ്റുകൂടാതെ ടൈപ്പ് ചെയ്തു നിങ്ങളുടെ ഫോൾഡറിൽ സേവ് ചെയ്യൂ.

```
birth_year=2011

current_year=2025

age = current_year - birth_year

print("Your age is : ", age)
```

പൈത്തൺഭാഷയിലെഴുതുന്ന പ്രോഗ്രാം ഫയലുകൾക്ക് .py എന്ന Extension ആണ് ഫയൽനാമത്തിന്റ കൂടെ ചേർക്കേണ്ടത്. വയസ്സു കണ്ടെത്തുന്ന ഈ പ്രോഗ്രാം സേവ് ചെയ്യുമ്പോൾ Find_age.py എന്നു പേരു നൽകാം.

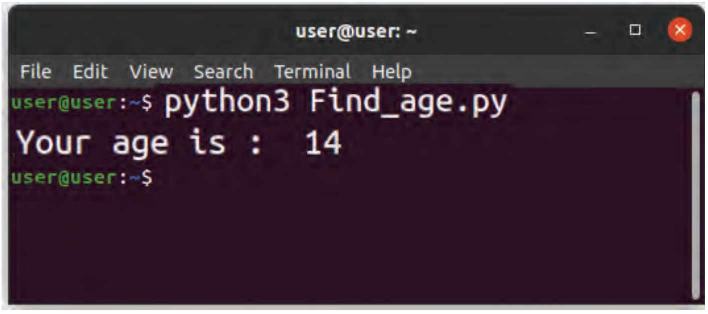
ഫയൽ സേവ് ചെയ്തില്ലേ. ഇനി Python3 എന്ന സോഫ്റ്റുവെയർ ഉപയോഗിച്ച് Find_age.py എന്ന ഫയൽ പ്രവർത്തിപ്പിച്ചുനോക്കൂ.

പൈത്തൺ ഫയൽ പ്രവർത്തിപ്പിക്കാൻ

- ഫയൽ സേവ് ചെയ്ത ഫോൾഡർ തുറക്കുക.
- ഫോൾഡറിൽ ഒഴിഞ്ഞ സ്ഥലത്ത് Right Click ചെയ്ത് Open in Terminal തുറക്കുക.
- തുടർന്ന്, ടെർമിനലിൽ താഴെ നൽകിയ നിർദേശം ടൈപ്പ്ചെയ്ത് എന്റർ ചെയ്യുക.

```
python3 Find_age.py
```

പ്രോഗ്രാം റൺ ചെയ്തപ്പോൾ വയസ്സ് എത്രയാണെന്ന് ടെർമിനലിൽ പ്രിന്റ് ചെയ്തുവന്നില്ലേ. (ചിത്രം 3.4)



ചിത്രം 3.4 ടെർമിനൽ ജാലകം

ഈ പ്രോഗ്രാമിലെ print("Your age is : " , age) എന്ന വരി എങ്ങനെയാണ് പ്രവർത്തിച്ചതെന്ന് ടെർമിനൽ പരിശോധിച്ച് കണ്ടെത്തൂ.

Your age is : എന്ന് പ്രിന്റ് ചെയ്തശേഷം ഗണിതക്രിയ ചെയ്തു ലഭിച്ച വിലയാണ് age ന്റെ സ്ഥാനത്ത് ഇവിടെ പ്രിന്റ് ചെയ്തിരിക്കുന്നത്.

ഇതേ മാതൃകയിൽ, നിശ്ചിത നീളവും വീതിയുമുള്ള ഒരു ചതുരത്തിന്റെ പരപ്പളവ് കാണുന്നതിനുള്ള പ്രോഗ്രാം ചുവടെ നൽകിയിരിക്കുന്നു.

ചരങ്ങൾ (Variables)

ഗണിതത്തിൽ ഉപയോഗിക്കുന്നതുപോലെ, കമ്പ്യൂട്ടർ പ്രോഗ്രാമുകളിലും ചരങ്ങൾ ഉപയോഗിക്കാം. ഒരു പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുന്നതിനാവശ്യമായ ഡാറ്റ താൽക്കാലികമായി ശേഖരിക്കുന്നതിനാണ് ഈ സങ്കേതം ഉപയോഗിക്കുക.

പ്രോഗ്രാമിൽ ചരങ്ങളുപയോഗിക്കുമ്പോൾ അവയ്ക്കു നൽകുന്ന പേര് അക്ഷരങ്ങളിലാണ് (Alphabet) തുടങ്ങേണ്ടത്. ഈ പേരിൽ Underscore(_) ഒഴികെയുള്ള Special characters ഉണ്ടാവരുത്.

ഓരോ വേരിയബിളും ഏതുതരത്തിലുള്ള ഡാറ്റയാണ് ഉൾക്കൊള്ളുന്നത് എന്നതിനെ ***Datatype*** എന്ന പദം കൊണ്ട് സൂചിപ്പിക്കുന്നു. Integer, float, string, image, audio, date, time എന്നിവ വിവിധ ഡാറ്റാടൈപ്പുകളാണ്. എന്നാൽ, പൈത്തൺ ഭാഷയിൽ വേരിയബിളുകൾ നിർമ്മിക്കുമ്പോൾ ഡാറ്റാടൈപ്പ് പ്രത്യേകം സൂചിപ്പിക്കേണ്ടതില്ല.

```
length=35
breadth=65
. . . . . . . . . . . . . . .
print("Area of Rectangle is: ", area)
```

ഈ പ്രോഗ്രാമിൽ വിട്ടുപോയ വരിയിൽ എന്താണ് ചേർക്കേണ്ടത്?

ടെക്സ്റ്റ് എഡിറ്ററിൽ പുതിയൊരു ഡോക്യുമെന്റ് തുറന്ന്, പ്രോഗ്രാം പൂർത്തിയാക്കി സേവ് ചെയ്ത് പ്രവർത്തിപ്പിച്ചു നോക്കൂ.

പ്രായം കണക്കാക്കുന്നതിനുള്ള പ്രോഗ്രാം പ്രവർത്തിപ്പിച്ചപ്പോൾ നാം നൽകിയ വർഷങ്ങൾക്കനുസരിച്ചുള്ള പ്രായമാണ് ഔട്ട്പുട്ട് ആയി ലഭിച്ചത്. ഈ പ്രോഗ്രാം ഏത് സമയത്തും ആര് പ്രവർത്തിപ്പിച്ചാലും വയസ്സ് 14 ആണ് ലഭിക്കുക.

ഇതിനു പകരം പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുന്ന ആളുടെ വയസ്സ് പ്രദർശിപ്പിക്കുന്ന പൈത്തൺ പ്രോഗ്രാം ആക്കി മാറ്റാൻ എന്താണ് ചെയ്യേണ്ടത്?

പ്രവർത്തിപ്പിക്കുന്ന സമയത്ത്, നിലവിലെ വർഷവും ജനനവർഷവും പ്രോഗ്രാമിലേക്കു നൽകാൻ സാധിച്ചാലോ? അപ്പോൾ ഇത് എല്ലാ കാലത്തും ഏതൊരാളുടെ പ്രായം കാണാനും ഉപയോഗിക്കാം, അല്ലേ?

പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുമ്പോൾ, വ്യക്തിയുടെ പേരും ജനനവർഷവും നിലവിലെ വർഷവും ഇൻപുട്ട് ചെയ്യുന്ന രീതിയിൽ ഈ പ്രോഗ്രാം പരിഷ്കരിക്കുന്നതെങ്ങനെയെന്നു നോക്കാം.

നേരത്തേ തയ്യാറാക്കിയ പ്രോഗ്രാമിൽ birth_year=2011 എന്നതിനു പകരം,

```
birth_year=input("Enter your year of birth :")
```

എന്നു മാറ്റിയെഴുതി പ്രോഗ്രാം ടെർമിനൽ ഉപയോഗിച്ച് പ്രവർത്തിപ്പിച്ചുനോക്കൂ.

എന്താണ് വ്യത്യാസം അനുഭവപ്പെട്ടത്?

....

....

ഇതേ മാതൃകയിൽ, നാം തയ്യാറാക്കിയ പ്രോഗ്രാമിന്റെ തുടക്കത്തിൽ താഴെ നൽകിയ വരികൾ കൂടി ചേർത്തുനോക്കാം.

```
name= input("What is your name?:")

current_year=input("Enter current year:")
```

ഈ പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുമ്പോൾ നിങ്ങളുടെ ജനനവർഷം, പേര്, നിലവിലെ വർഷം എന്നിവ ചോദിക്കുകയും Keyboard വഴി ടൈപ്പ് ചെയ്യുന്ന ജനനവർഷം, പേര്, നിലവിലെ വർഷം എന്നിവ യഥാക്രമം birth_year, name, current_year എന്നീ ചരങ്ങളിൽ സൂക്ഷിക്കുകയും ചെയ്യുന്നു.

ഈ പ്രോഗ്രാം സേവ് ചെയ്തു പ്രവർത്തിപ്പിച്ചു നോക്കൂ.

പുതുക്കിയെഴുതിയ പ്രോഗ്രാം പ്രവർത്തിപ്പിച്ചപ്പോൾ നിങ്ങൾ ആഗ്രഹിക്കുന്ന ഔട്ട്പുട്ട് ലഭിച്ചുവോ?

നാം നൽകിയ ഇൻപുട്ടുകൾ ഉപയോഗിച്ച് ഗണിതക്രിയകൾ സാധ്യമാവാത്തതിനാലാണ് ഈ പ്രശ്നം ഉണ്ടായത്. ലഭിച്ച വിവരങ്ങൾ സംഖ്യകളിലേക്കു മാറ്റിയെങ്കിൽ മാത്രമേ അവയുടെ വ്യത്യാസം കണക്കാക്കാൻ കഴിയൂ.

ഇൻപുട്ട് (input) ഫങ്ഷൻ ഉപയോഗിക്കുമ്പോൾ

പൈത്തണിൽ input ഫങ്ഷൻ ഉപയോഗിച്ച് ലഭിക്കുന്ന ഡാറ്റ അക്ഷരരൂപത്തിലുള്ളതായിരിക്കും. String എന്നാണതിനെ സൂചിപ്പിക്കുന്നത്.

ഒരു വ്യക്തിയുടെ പേര്, അഡ്രസ് മുതലായ വിവരങ്ങളിൽ അക്ഷരങ്ങൾ (Letters) ആയിരിക്കും ഉണ്ടായിരിക്കുക. എന്നാൽ Year of birth, Height, Weight എന്നീ വിവരങ്ങളിൽ അക്കങ്ങളാ (Digits) യിരിക്കും.

വാക്കുകളിൽ സാധാരണയായി സങ്കലനം, വ്യവകലനം, ഗുണനം, ഹരണം തുടങ്ങിയ ഗണിത പ്രവർത്തനങ്ങൾ (Arithmetic Operations) ചെയ്യാറില്ല. അതിനാൽ, സംഖ്യകൾ ഇത്തരം കാര്യങ്ങൾക്ക് ഉപയോഗിക്കേണ്ടിവരും.

String രൂപത്തിൽ ലഭ്യമായ വിവരങ്ങളെ integer രൂപത്തിലേക്കു മാറ്റുന്നതിന് int() എന്ന ഫങ്ഷൻ ഉപയോഗിക്കാം.

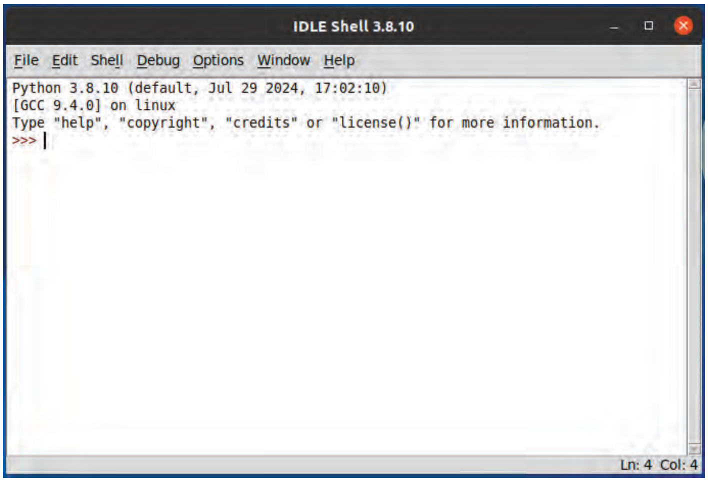
അതിനാൽ, പ്രോഗ്രാം താഴെ നൽകിയ രീതിയിൽ മാറ്റിയെഴുതാം.

```
name = input("What is your name:")

current_year = input("Enter current year:")

birth_year = input("Enter your birth year:")

age=int(current_year) - int(birth_year)

print(name, "  Your age is : " , age)
```

ഗണിതക്രിയ ചെയ്യേണ്ട വരിയിൽ int() എന്ന ഫങ്ഷൻ ഉപയോഗിച്ചിരിക്കുന്നത് പ്രത്യേകം ശ്രദ്ധിക്കുമല്ലോ.

പൈത്തൺ IDE അഥവാ IDLE

Text Editor ഉപയോഗിച്ചാണല്ലോ നാമിപ്പോൾ പൈത്തൺപ്രോഗ്രാം തയ്യാറാക്കിയത്. എന്നാൽ, വലിയ കമ്പ്യൂട്ടർപ്രോഗ്രാമുകൾ തയ്യാറാക്കുമ്പോൾ പ്രോഗ്രാം എളുപ്പത്തിൽ പ്രവർത്തിപ്പിക്കുന്നതിനും പിശകുകൾ തിരുത്തുന്നതിനുമൊക്കെ സഹായിക്കുന്ന IDE (Integrated Development Environment)എന്നറിയപ്പെടുന്ന പ്രത്യേക സോഫ്റ്റ്‌വെയറുകളാണ് ഉപയോഗിക്കാറുള്ളത്. പൈത്തൺഭാഷയിൽ പ്രോഗ്രാം എഴുതുന്നതിനുള്ള ഒരു IDE ആയ IDLE3 ആണ് ചിത്രം 3.5 ൽ നൽകിയിരിക്കുന്നത്.



ചിത്രം 3.5 IDLE 3 ജാലകം

നിങ്ങളുടെ കമ്പ്യൂട്ടറിൽ IDLE3 തുറന്ന് എന്തെല്ലാം സൗകര്യങ്ങളാണ് ഇതിൽ ലഭ്യമായതെന്ന് പരിശോധിക്കൂ.

പുതിയ ഫയൽ നിർമ്മിക്കുന്നതിനും സേവ് ചെയ്യുന്നതിനും പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുന്നതിനുമെല്ലാം സഹായിക്കുന്ന സൗകര്യങ്ങൾ കാണുന്നില്ലേ.

പ്രോഗ്രാം IDLE3 ഉപയോഗിച്ച് പ്രവർത്തിപ്പിക്കാം

തയ്യാറാക്കിയ പ്രോഗ്രാം ടെർമിനൽ ഉപയോഗിച്ചാണല്ലോ നാം പ്രവർത്തിപ്പിച്ചത്. താഴെ നൽകിയിരിക്കുന്ന രീതിയിൽ പ്രോഗ്രാം IDLE3 ൽ തയ്യാറാക്കി പ്രവർത്തിപ്പിച്ചുനോക്കൂ.

പൈത്തൺ ഓപ്പറേറ്ററുകൾ

പൈത്തണിൽ വ്യത്യസ്ത ഡാറ്റകളിലും വേരിയബിളിലും പ്രയോഗിക്കാൻ കഴിയുന്നവ ഓപ്പറേറ്ററുകൾ.

1. Arithmetic operators:-ഗണിതക്രിയകൾ ചെയ്യുന്നതിനുള്ളവ:

Addition	+
Subtraction	-
Multiplication	*
Division	/
Modulus	%

2. Assignment Operator:- വേരിയബിളിലേക്ക് വില ചേർക്കുന്നതിന് = ഉപയോഗിക്കുന്നു.

ഉദാ:- Mark=50

3. Comparison Operators:-

വിലകൾ താരതമ്യം ചെയ്യുന്നതിനുള്ളവ:

Equal to	==
Greater than	>
Less than	<
Greater than or equal	>=
Less than or equal	<=
Not equal	!=

4. Logical Operators:- ഒന്നിലധികം സ്റ്റേറ്റ്മെന്റുകൾ യോജിപ്പിക്കുന്നതിന്:

and, or, not

IDLE3 ൽ പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കാൻ

- IDLE3 തുറന്ന്, File ⟶ New File വഴി പുതിയ ഫയൽ തുറക്കുക.
- പ്രോഗ്രാം തെറ്റുകൂടാതെ ടൈപ്പ് ചെയ്തു ചേർക്കുക (ചിത്രം 3.6).
- File ⟶ Save ക്രമത്തിൽ ക്ലിക്ക് ചെയ്ത്, ഫയൽനാമം നൽകി നിങ്ങളുടെ ഫോൾഡറിൽ സേവ് ചെയ്യുക.
- തുടർന്ന് Run മെനുവിൽനിന്ന് Run Module ക്ലിക്ക് ചെയ്ത് പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുക.

```
name=input("Enter your name:")
birth_year=input("Enter your birth year:")
current_year=input("Enter current year:")
age= int(current_year)-int(birth_year)
print(name, "  Your age is : " , age)
```

ചിത്രം 3.6 IDLE3 ൽ തയ്യാറാക്കിയ പൈത്തൺ പ്രോഗ്രാം

```
File  Edit  Shell  Debug  Options  Window  H
Python 3.8.10 (default, Jul 29 2024, 17
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or
>>>
======================== RESTART: /home/
Enter your name:Sandeep
Enter your birth year:2011
Enter current year:2025
Sandeep   Your age is :  14
>>> |
```

ചിത്രം 3.7 പ്രോഗ്രാം IDLE3 ൽ പ്രവർത്തിപ്പിച്ചപ്പോൾ ലഭിച്ച ഔട്ട്പുട്ട് ജാലകം

പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുമ്പോൾ പേരും ജനന വർഷവും നിലവിലെ വർഷവും input ആയി നൽകുകയും നൽകിയ വിവരങ്ങൾക്കനുസരിച്ചുള്ള output ലഭിക്കുകയും ചെയ്തല്ലോ.

കണ്ടീഷണൽ സ്റ്റേറ്റ്മെന്റ്

സന്ദീപ് സാധാരണദിവസങ്ങളെ പോലെയല്ല അവധി ദിവസങ്ങളിൽ എഴുന്നേൽക്കുന്നതും മറ്റു കാര്യങ്ങൾ നിർവഹിക്കുന്നതുമെന്ന് പാഠഭാഗത്തിന്റെ തുടക്കത്തിൽ നൽകിയ ദിനചര്യയിലൂടെ കണ്ടല്ലോ.

അതായത്, ചില പ്രത്യേക സന്ദർഭങ്ങളിൽ ഒരു രീതിയിലും മറ്റു സന്ദർഭങ്ങളിൽ വേറെ രീതിയിലുമാണ് സന്ദീപ് പ്രവർത്തിക്കുന്നത്. ഇതുപോലെ കമ്പ്യൂട്ടറിലും പ്രോഗ്രാമിന്റെ ഒഴുക്ക് ചില സന്ദർഭങ്ങളിൽ നിബന്ധനകൾക്കനുസരിച്ച് വ്യത്യസ്തമാക്കേണ്ടിവരും.

പ്രോഗ്രാമിങ് രീതിശാസ്ത്രത്തിൽ ***ബ്രാഞ്ചിങ്*** എന്നാണിതിനെ സൂചിപ്പിക്കുന്നത്.

ഉദാഹരണമായി, നമ്മുടെ സ്കൂളിൽ കായികമേള നടക്കുമ്പോൾ കുട്ടികളുടെ പ്രായത്തിനനുസരിച്ചാണല്ലോ അവരെ വ്യത്യസ്ത കാറ്റഗറികളിൽ മത്സരിപ്പിക്കുന്നത്. 14 വയസ്സിനു മുകളിലുള്ളവർ ജൂനിയർ വിഭാഗവും അതിനു താഴെയുള്ളവർ സബ്-ജൂനിയർ വിഭാഗവുമാണെന്നിരിക്കട്ടെ. ലഭിച്ച പ്രായത്തിനനുസരിച്ച് കാറ്റഗറി ഏതാണെന്നു കൂടി പ്രിന്റ് ചെയ്യുന്ന രീതിയിൽ നമ്മുടെ പ്രോഗ്രാമിൽ എങ്ങനെ മാറ്റം വരുത്താമെന്നു നോക്കാം.

ഒരു പ്രത്യേക നിബന്ധന (Condition) പാലിക്കുന്നുണ്ടോ എന്ന് പരിശോധിച്ച ശേഷം പാലിക്കുന്നുണ്ടെങ്കിൽ ഒരു രീതിയിലും പാലിക്കുന്നില്ലെങ്കിൽ മറ്റൊരു രീതിയിലും പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുന്നതിന് if ... else സ്റ്റേറ്റ്മെന്റ് ഉപയോഗിക്കാം.

ഇത് ടൈപ്പ് ചെയ്യേണ്ട രീതി താഴെ നൽകിയിരിക്കുന്നു.

```
if <condition> :
    statements for condition true
else:
    statements for condition false
```

നമ്മുടെ പ്രോഗ്രാമിൽ, വയസ്സ് 14ഉം അതിൽ കൂടുതലുമാണെങ്കിൽ ജൂനിയറാണെന്നും അല്ലെങ്കിൽ സബ്ജൂനിയറാണെന്നും പ്രിന്റ് ചെയ്യണമെന്നിരിക്കട്ടെ. അപ്പോൾ കണ്ടീഷണൽ സ്റ്റേറ്റ്മെന്റ് ചേർത്ത്, താഴെ നൽകിയ രീതിയിൽ പ്രോഗ്രാം എഴുതാം.

```
if age>=14:
    print("You are in Junior Category")
else:
    print("You are in Sub-Junior Category")
```

പ്രോഗ്രാമിങ്ങിലെ പദവിന്യാസക്രമം

ഓരോ പ്രോഗ്രാമിങ് ഭാഷയ്ക്കും പ്രത്യേക പദങ്ങളും പദവിന്യാസക്രമവുമുണ്ട്. ഇവയാണ് Syntax അല്ലെങ്കിൽ ഉപയോഗക്രമം.

if <condition>: എന്ന വരിയിലെ : കാരക്ടർ, നിബന്ധന പാലിക്കുന്നുണ്ടെങ്കിൽ പ്രവർത്തിക്കേണ്ട രീതികൾക്കു മുൻപിലെ Indentation, നിബന്ധന പാലിക്കുന്നില്ലെങ്കിൽ പ്രവർത്തിക്കേണ്ട വരികൾക്ക് മുൻപിലെ Indentation എന്നിവ പ്രത്യേകം ശ്രദ്ധിക്കുക.

പൈത്തൺപദങ്ങൾ ഇംഗ്ലീഷ് അക്ഷരമാലയിലെ ചെറിയ അക്ഷരങ്ങൾ (Lowercase) ഉപയോഗിച്ചാണ് എഴുതേണ്ടത്. ഇത് പൈത്തൺ ഭാഷയുടെ Syntax ന്റെ ഭാഗമാണ്.

നേരത്തേ തയ്യാറാക്കിയ പ്രോഗ്രാമിൽ ഈ വരികൾ കൂടി ചേർത്തു പ്രവർത്തിപ്പിച്ചുനോക്കൂ.

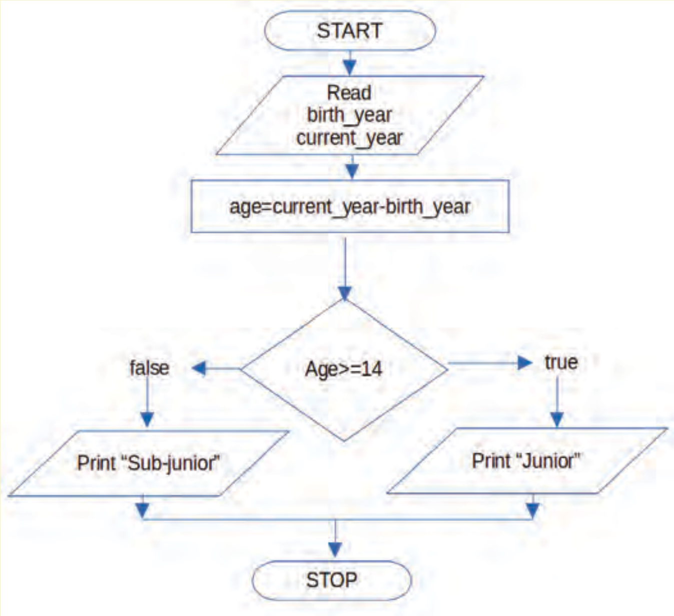
```
name=input("Enter your name:")
birth_year=input("Enter your birth year:")
current_year=input("Enter current year:")
age= int(current_year)-int(birth_year)
print(name, "  Your age is : " , age)
if age>=14:
    print("You are in Junior Category")
else:
    print("You are in Sub-Junior Category")
```

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 1
Type "help", "copyright", "credits" or "license()"
>>>
================================== RESTART: /home/r
Enter your name:Anu
Enter your birth year:2013
Enter current year:2025
Anu  Your Age is : 12
You are in Sub-Junior Category
>>>
```

ചിത്രം 3.8 if...else ചേർത്ത പ്രോഗ്രാമിന്റെ ഔട്ട്പുട്ട്

Flow Chart

നാം തയ്യാറാക്കിയ പ്രോഗ്രാമിന്റെ ചിത്രരൂപത്തിലുള്ള അവതരണം (Flowchart) നോക്കൂ. വലിയ സങ്കീർണ്ണമായ പ്രശ്നങ്ങൾ എളുപ്പത്തിൽ വിശകലനം ചെയ്ത് പ്രശ്നപരിഹാരം കണ്ടെത്താൻ Flowchart കൾ വഴിയുള്ള ചിത്രീകരണം സഹായകമാണ്.



ഇവിടെ നാം ഒരു നിബന്ധന മാത്രമാണ് പരിശോധിച്ചത്. ചില സന്ദർഭങ്ങളിൽ ആദ്യത്തെ നിബന്ധന പാലിച്ചില്ലെങ്കിൽ മറ്റു നിബന്ധനകൾ പാലിക്കുന്നുണ്ടോ എന്നുകൂടി പരിശോധിക്കേണ്ടിവരും.

ഉദാഹരണമായി, 16 വയസ്സിനുമുകളിലുള്ളവർ സീനിയർ കാറ്റഗറിയിലും 14 നു മുകളിലുള്ളവർ ജൂനിയർ, 12നു മുകളിലുള്ളവർ സബ് ജൂനിയർ, അതിനു താഴെയുള്ളവർ Kiddies കാറ്റഗറികളിലായി വേർതിരിക്കണമെന്നിരിക്കട്ടെ. ഇങ്ങനെ ഒന്നിലധികം നിബന്ധനകൾ പരിശോധിക്കുന്നതിനായി if ....elif ...else എന്ന Format താഴെ നൽകിയ രീതിയിൽ ഉപയോഗിക്കാം.

```
if age>=16:
    print("You are in Senior Category")
elif age>=14:
    print("You are in Junior Category")
elif age>=12:
    print("You are in Sub-Junior Category")
else:
    print("You are in Kiddies Category")
```

ഇത്തരത്തിൽ പ്രോഗ്രാം പരിഷ്കരിച്ചു പ്രവർത്തിപ്പിച്ചുനോക്കുമല്ലോ.

ആവർത്തനനിർദേശങ്ങൾ

ഇപ്പോൾ നമ്മുടെ പ്രോഗ്രാം ഒരു വ്യക്തിയുടെ പേര്, ജനനവർഷം, നിലവിലെ വർഷം എന്നീ Input കൾ സ്വീകരിച്ച് പ്രായം കണക്കാക്കുകയും പ്രായത്തിനനുസരിച്ച് കാറ്റഗറി ഏതാണെന്നു പ്രിന്റ് ചെയ്യുന്ന വിധത്തിലും ആയിക്കഴിഞ്ഞു.

സ്കൂൾ കായികമേളയിൽ നമ്മുടെ ക്ലാസിൽനിന്ന് ഒന്നിലധികം കുട്ടികൾ പങ്കെടുക്കുന്നുവെന്ന് കരുതുക. അവരുടെയെല്ലാം പ്രായവും കാറ്റഗറിയും കണ്ടെത്തുന്ന രീതിയിലേക്ക് ഈ പ്രോഗ്രാമിനെ മാറ്റാൻ എന്താണു ചെയ്യേണ്ടത്?

എത്ര കുട്ടികളാണ് മത്സരത്തിൽ പങ്കെടുക്കുന്നത്, അത്രയും തവണ നമ്മുടെ പ്രോഗ്രാം വരികൾ ആവർത്തിച്ചുകൊണ്ടിരിക്കണം, അല്ലേ.

കമ്പ്യൂട്ടർപ്രോഗ്രാം തയ്യാറാക്കുമ്പോൾ, ഇത്തരത്തിലുള്ള ആവർത്തനനിർദേശങ്ങൾ നൽകുന്നതിന് Loop സ്റ്റേറ്റ്മെന്റുകൾ ഉപയോഗിക്കാം.

പ്രധാനമായും രണ്ടു തരത്തിലുള്ള Loop സ്റ്റേറ്റ്മെന്റു കളാണ് പൈത്തൺ ഭാഷയിലുള്ളത്.

1. while loop
2. for loop

while loop

while ലൂപ്പുകളുടെ syntax താഴെ പറയുന്ന വിധത്തിലാണ്.

while <condition is true> :

    statements to repeat

നമ്മുടെ ക്ലാസിൽനിന്ന് മൂന്നു കുട്ടികളാണ് കായികമേളയിൽ പങ്കെടുക്കുന്നത് എന്നിരിക്കട്ടെ. while loop ഉപയോഗിച്ച് അതെങ്ങനെ ചെയ്യാമെന്ന് നോക്കാം.

അതിനായി പ്രോഗ്രാമിൽ count എന്ന ഒരു ചരം കൂടി ചേർക്കാം. തുടക്കത്തിൽ അതിന്റെ വില പൂജ്യം ആക്കി വയ്ക്കാം.

count = 0

തുടർന്ന്, ഓരോ തവണ ആവർത്തിക്കുമ്പോഴും count എന്ന ചരത്തിന്റെ വില 1 കൂട്ടി കൊണ്ടുവരണം. അതിനായി,

count = count + 1   എന്ന കോഡ് ഉപയോഗിക്കാം.

ആകെ മൂന്നു കുട്ടികളാണല്ലോ പങ്കെടുക്കുന്നത്. അപ്പോൾ count മൂന്നിൽ താഴെയാവുമ്പോൾ മാത്രമേ ആവർത്തിക്കേണ്ടതുള്ളൂ. മൂന്നോ അതിൽ കൂടുതലോ ആയാൽ പ്രോഗ്രാം അവസാനിക്കണം.

അപ്പോൾ while ലൂപ്പിലെ നിർദേശങ്ങൾ പ്രവർത്തിക്കുന്നതിനുള്ള നിബന്ധന എന്നത് count < 3 എന്നതാണ്.

അതായത്,

while count < 3 :

ഇനി ആവർത്തിക്കേണ്ട കോഡുകൾകൂടി ചേർത്ത് പ്രോഗ്രാം ചിത്രം 3.9 ൽ കാണുന്ന വിധത്തിൽ മാറ്റിയെഴുതാം.

ആവർത്തിച്ച് വരേണ്ട നിർദേശങ്ങൾ while ബ്ലോക്കിൽനിന്ന് ഒരു Indentation അകലത്തിലാണ് എഴുതിയിരിക്കുന്നത് എന്നത് ശ്രദ്ധിക്കുമല്ലോ.

File Edit Format Run Options Window Help

```
count=0
while count<3:
    name=input("Enter your name:")
    birth_year=input("Enter your birth year:")
    current_year=input("Enter current year:")
    age= int(current_year)-int(birth_year)
    print(name, "  Your age is : " , age)
    if age>=14:
        print("You are in Junior Category")
    else:
        print("You are in Sub-Junior Category")
    count = count + 1
```

ചിത്രം 3.9 while നിർദേശം ഉപയോഗിച്ച പൈത്തൺ പ്രോഗ്രാം

ഇനി പ്രോഗ്രാം പ്രവർത്തിപ്പിച്ചുനോക്കൂ.

മൂന്നു കുട്ടികളുടെ വിവരങ്ങൾ നൽകിക്കഴിഞ്ഞാൽ പ്രോഗ്രാം അവസാനിക്കുന്നില്ലേ. ഈ പ്രോഗ്രാമിൽ,

- പ്രോഗ്രാം ആരംഭിക്കുമ്പോൾ count എന്ന വേരിയബിളിന്റെ വില പൂജ്യമായിരുന്നു.
- ഓരോ തവണ while ബ്ലോക്കിലെ കോഡുകൾ പ്രവർത്തിക്കുമ്പോഴും count ന്റെ വില 1 കൂടുന്നു.
- മൂന്നു തവണ ആവർത്തിക്കുമ്പോൾ countന്റെ വില 3 ആവുന്നു. അപ്പോൾ count < 3 എന്ന നിബന്ധന പാലിക്കാതിരിക്കുകയും പ്രവർത്തനം അവസാനിക്കുകയും ചെയ്യുന്നു.

ഈ പ്രോഗ്രാം പ്രവർത്തിക്കുമ്പോൾ കുട്ടികളുടെ എണ്ണം input ആയി സ്വീകരിക്കുന്ന തരത്തിൽ പ്രോഗ്രാം മാറ്റിയെഴുതിരിക്കുന്നതു നോക്കൂ (ചിത്രം 3.10). no_of_students എന്ന ചരമാണ് കുട്ടികളുടെ എണ്ണം സ്വീകരിക്കാൻ ഇവിടെ ഉപയോഗിച്ചിരിക്കുന്നത്.

```
File  Edit  Format  Run  Options  Window  Help
no_of_students=input("Enter the number of students:")
count=0
while count < int(no_of_students):
    name=input("Enter your name:")
    birth_year=input("Enter your birth year:")
    current_year=input("Enter current year:")
    age= int(current_year)-int(birth_year)
    print(name, "  Your age is : " , age)
    if age>=14:
        print("You are in Junior Category")
    else:
        print("You are in Sub-Junior Category")
    count = count + 1
```

ചിത്രം 3.10 കുട്ടികളുടെ എണ്ണം ഇൻപുട്ടായി സ്വീകരിക്കുന്ന പ്രോഗ്രാം

for loop

ആവർത്തനനിർദേശങ്ങൾ നൽകാനുള്ള മറ്റൊരു സങ്കേതമാണ് for ലൂപ്പ്.

1 മുതൽ 100 വരെയുള്ള എണ്ണൽസംഖ്യകൾ സ്ക്രീനിൽ പ്രിന്റ് ചെയ്യുന്നതിനുള്ള പൈത്തൺ പ്രോഗ്രാം while ലൂപ്പ് ഉപയോഗിച്ച് തയ്യാറാക്കിയതു നോക്കൂ.

```
count=1
while count <=100:
    print(count)
    count = count +1
```

=, = = എന്നിവ തമ്മിലുള്ള വ്യത്യാസം

= എന്ന ഓപ്പറേറ്റർ ചരത്തിലേക്ക് ഒരു വില assign ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്നതാണ്. ഉദാ:- price=200 എന്നത് price എന്ന ചരത്തിലേക്ക് 200 എന്ന വില ചേർക്കുന്നതിനാണ്.

== എന്ന ഓപ്പറേറ്റർ തുല്യത പരിശോധിക്കുന്നതിനാണ് ഉപയോഗിക്കുന്നത്. ഉദാ:- price == 200 എന്നത് price ന്റെ വില 200 ആണോ എന്നു പരിശോധിക്കുന്നു.

ഈ പ്രോഗ്രാം IDLE3ൽ എഴുതി പ്രവർത്തിപ്പിച്ചു നോക്കുമല്ലോ.

ഇനി, ഇതേ പ്രോഗ്രാം for ലൂപ്പ് ഉപയോഗിച്ച് എഴുതിയതു കാണുക.

```
for count in range(1,101):
    print(count)
```

പ്രോഗ്രാം ടൈപ്പ് ചെയ്തു പ്രവർത്തിപ്പിച്ചല്ലോ.

എന്തെല്ലാമാണ് while, for ലൂപ്പുകൾ ഉപയോഗിച്ചപ്പോഴുള്ള വ്യത്യാസങ്ങൾ? പട്ടിക 3.1 നോക്കൂ.

while	for
count എന്ന വേരിയബിളിന് തുടക്ക വില നൽകി. (count=1)	ഈ കാര്യങ്ങളെല്ലാംതന്നെ ഒറ്റവരിയിൽ സൂചിപ്പിക്കുന്നു.
വില വർധിക്കുന്നതിനുള്ള കോഡ് ഉപയോഗിച്ചു. (count = count + 1)	range നോടൊപ്പമാണ് for ഉപയോഗിക്കുന്നത്. for count in range(1,101):
നിബന്ധന പരിശോധിക്കുന്നതിനുള്ള കോഡ് ചേർത്തു. (count<=100)	

പട്ടിക 3.1 while, for ലൂപ്പുകൾ ഉപയോഗിക്കുമ്പോഴുള്ള വ്യത്യാസം

range നിർദേശം

പൈത്തൺഭാഷയിൽ സംഖ്യകളെ ആവശ്യാനുസരണം ക്രമീകരിക്കുന്നതിന് range എന്ന സങ്കേതം ഉപയോഗിക്കുന്നു.

- range(10) എന്ന് നൽകിയാൽ 0 മുതൽ 9 വരെയുള്ള പത്തു സംഖ്യകൾ (0,1,2,3,4,5,6,7,8,9) ലഭ്യമാകുന്നു.
- range(1,10) എന്നു നൽകിയാൽ 1 മുതൽ 9 വരെയുള്ള 9 സംഖ്യകൾ (1,2,3,4,5,6,7,8,9) ആണ് ലഭിക്കുക.
- range(1,20,2) എന്നു നൽകിയാൽ 1 മുതൽ 20 ൽ താഴെയുള്ള ഒറ്റസംഖ്യകൾ (1, 3, 5, 7, 9, 11, 13, 15, 17, 19) ലഭ്യമാവും. അതായത് സംഖ്യകളുടെ ക്രമം രണ്ടുവീതം വർധിക്കും.

പൈത്തൺഭാഷ ഉപയോഗിച്ച് തയ്യാറാക്കിയ ചില പ്രോഗ്രാമുകളാണ് നാമിപ്പോൾ പരിചയപ്പെട്ടത്.

നാമിപ്പോൾ ചെയ്തതുപോലെ, ടെക്സ്റ്റ് രൂപത്തിലുള്ള Output മാത്രമാണോ പൈത്തൺ ഉപയോഗിച്ച് തയ്യാറാക്കാൻ സാധിക്കുക?

ഉദാഹരണമായി, നിങ്ങൾക്കൊരു ചിത്രം കമ്പ്യൂട്ടറിൽ വരയ്ക്കണമെങ്കിൽ സാധാരണയായി ഏതു സോഫ്റ്റ്‌വെയറാണ് ഉപയോഗിക്കുക?

- GIMP
- Inkscape
- ....

ഇതിനുപകരം പൈത്തൺ പ്രോഗ്രാം എഴുതി കമ്പ്യൂട്ടറിൽ ചിത്രം വരയ്ക്കാൻ സാധിക്കുമോ? നമുക്ക് പരിശോധിച്ചാലോ.

പൈത്തൺ ഗ്രാഫിക്സ്

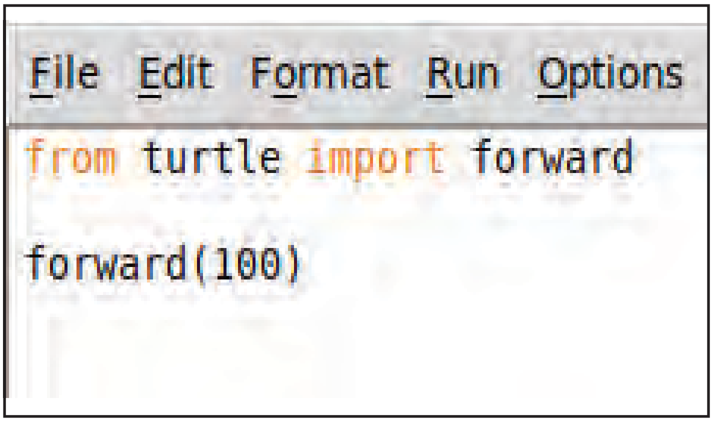
ടെക്സ്റ്റ് രൂപത്തിലുള്ളവ മാത്രമല്ല, ചിത്രങ്ങളും ജ്യാമിതീയരൂപങ്ങളുമൊക്കെ വരയ്ക്കാനും ഗ്രാഫിക്കൽ രൂപത്തിലുള്ള Output നൽകാനും പൈത്തൺഭാഷയ്ക്ക് കഴിയും. ഇത്തരം വ്യത്യസ്തമായ ആവശ്യങ്ങൾക്ക് പ്രത്യേകം ലൈബ്രറി മൊഡ്യൂളുകൾ പൈത്തൺഭാഷയിലുണ്ട്. turtle എന്ന മൊഡ്യൂൾ ഉപയോഗിച്ചുകൊണ്ട് പൈത്തൺ ഭാഷയിൽ Graphical രൂപത്തിലുള്ള Output നൽകുന്ന പ്രോഗ്രാമുകൾ തയ്യാറാക്കാൻ കഴിയും.

turtle മൊഡ്യൂൾ ഉപയോഗിച്ച്, വര വരയ്ക്കുന്നതിനുള്ള ഫങ്ഷനാണ് forward. ഇതുമാത്രമാണ് നാം ഉപയോഗിക്കുന്നതെങ്കിൽ,

from turtle import forward

എന്നു തുടക്കത്തിൽ ചേർത്ത് പ്രോഗ്രാം എഴുതാം.

IDLE3 തുറന്ന് താഴെ കാണുന്ന പൈത്തൺ പ്രോഗ്രാം (ചിത്രം 3.11) എഴുതി പ്രവർത്തിപ്പിച്ചുനോക്കൂ.



ചിത്രം 3.11 വര വരയ്ക്കുന്നതിനുള്ള പൈത്തൺ പ്രോഗ്രാം

ഗ്രാഫിക്സ് ജാലകത്തിൽ 100 pixel നീളമുള്ള ഒരു വര പ്രത്യക്ഷപ്പെട്ടല്ലോ. (ചിത്രം 3.12)

പൈത്തൺ മൊഡ്യൂളുകൾ

പ്രത്യേകമായ ആവശ്യങ്ങൾക്കുള്ള ഫങ്ഷനുകളും വേരിയബിളുകളും classകളും രേഖപ്പെടുത്തിവച്ചിട്ടുള്ള പൈത്തൺ ഫയലുകൾതന്നെയാണ് മൊഡ്യൂളുകൾ.

import എന്ന കീവേഡ് ഉപയോഗിച്ച് മൊഡ്യൂളുകളെ ആവശ്യാനുസരണം പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്താം.

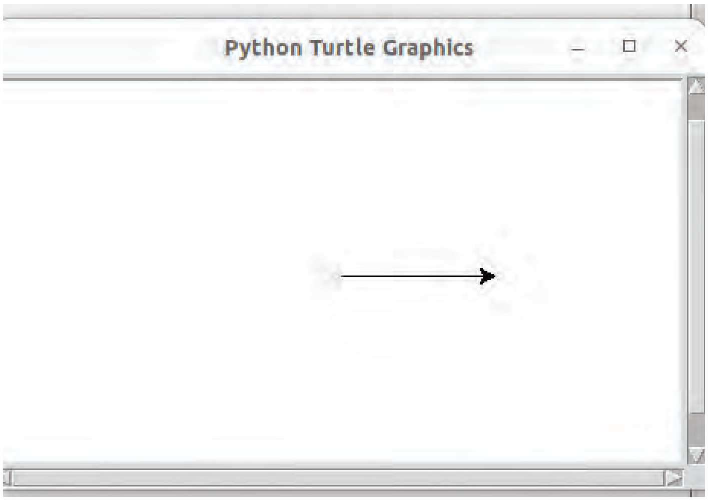
ചില പൈത്തൺ മൊഡ്യൂളുകൾ

os - ഓപ്പറേറ്റിങ് സിസ്റ്റവുമായി ബന്ധപ്പെട്ട ഫങ്ഷനുകൾക്ക്:

math - ഗണിതക്രിയകൾക്ക്.
ഉദാ:- sqrt, sin, cos etc

turtle - ഗ്രാഫിക്സ്, ഡ്രോയിങ്

datetime – സമയം, തിയ്യതി എന്നിവയുമായി ബന്ധപ്പെട്ട കാര്യങ്ങൾക്ക്.



ചിത്രം 3.12 പൈത്തൺ ഗ്രാഫിക്സ് ജാലകം

turtle മൊഡ്യൂളിൽ ലഭ്യമായ എല്ലാ ഫങ്ഷനുകളും ക്ലാസുകളും നമ്മുടെ പ്രോഗ്രാമിൽ ചേർക്കുന്നതിനായി from turtle import * എന്നാണ് പ്രോഗ്രാമിന്റെ തുടക്കത്തിൽ ചേർക്കേണ്ടത്.

വര വരയ്ക്കുന്നതിന് forward നിർദേശം ഉപയോഗിക്കാം എന്നു കണ്ടല്ലോ. 90 ഡിഗ്രി വലത്തോട്ടു തിരിയുന്നതിനു right(90) എന്ന നിർദേശം ഉപയോഗിക്കാം.

അങ്ങനെയെങ്കിൽ 100 യൂണിറ്റ് നീളവും 100 യൂണിറ്റ് വീതിയുമുള്ള ചതുരാകൃതിയിലുള്ള ഒരു കളിസ്ഥലത്തിന്റെ രൂപം പൈത്തൺ ഉപയോഗിച്ച് വരയ്ക്കണമെങ്കിൽ എന്തെല്ലാം നിർദേശങ്ങൾ നൽകേണ്ടി വരും? (ചിത്രം 3.13).

File Edit Format Run Options

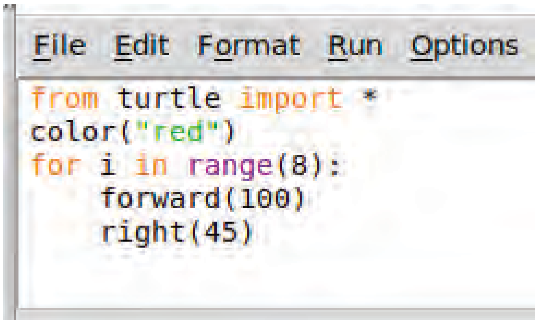
```
from turtle import *
for i in range(4):
    forward(100)
    right(90)
```

ചിത്രം 3.13 100 യൂണിറ്റ് നീളവും വീതിയുമുള്ള കളിസ്ഥലത്തിന്റെ രൂപം വരയ്ക്കുന്നതിനുള്ള പ്രോഗ്രാം

ഈ പ്രോഗ്രാമിൽ മുന്നോട്ട് 100 യൂണിറ്റ് നീങ്ങുന്നതിനും 90 ഡിഗ്രി വലത്തോട്ടു തിരിയുന്നതിനുമുള്ള നിർദേശങ്ങൾ നാലു തവണ പ്രവർത്തിക്കുമ്പോൾ ചതുരാകൃതിയിലുള്ള കളിസ്ഥലത്തിന്റെ രൂപം തയ്യാറാവുന്നു.

ഇപ്പോൾ നമുക്കു ലഭിച്ച രൂപങ്ങൾ ഏതു നിറത്തിലുള്ളതാണ്?

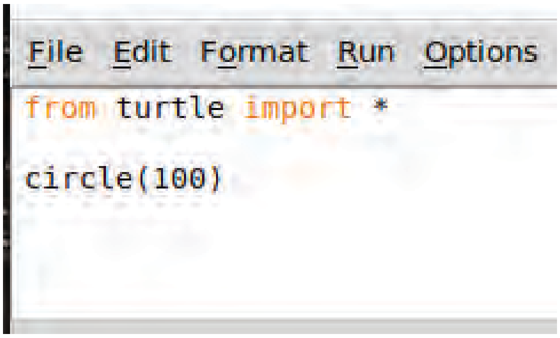
IDLE3 തുറന്ന് ചിത്രം 3.14, 3.15 എന്നിവയിൽ നൽകിയിരിക്കുന്ന പ്രോഗ്രാമുകൾ ഓരോന്നായി ടൈപ്പ് ചെയ്തു പ്രവർത്തിപ്പിച്ചുനോക്കൂ.



```
File  Edit  Format  Run  Options
from turtle import *
color("red")
for i in range(8):
    forward(100)
    right(45)
```

ചിത്രം 3.14

ടർട്ടിൽ ഗ്രാഫിക്സ് ഉപയോഗിച്ച് രൂപം വരയ്ക്കുന്നതിനുള്ള പ്രോഗ്രാം 1



```
File  Edit  Format  Run  Options
from turtle import *

circle(100)
```

ചിത്രം 3.15

ടർട്ടിൽ ഗ്രാഫിക്സ് ഉപയോഗിച്ച് രൂപം വരയ്ക്കുന്നതിനുള്ള പ്രോഗ്രാം 2

പ്രോഗ്രാമുകളുടെ ഔട്ട്പുട്ട് കണ്ടല്ലോ. ഇവയിൽ ഉപയോഗിച്ച forward, left, right, circle, range എന്നീ നിർദേശങ്ങളുടെ ഉപയോഗം തിരിച്ചറിഞ്ഞില്ലേ.

ഇനി, ഇവ ഉപയോഗപ്പെടുത്തി വ്യത്യസ്ത രൂപങ്ങൾ വരച്ച് ഔട്ട്പുട്ട് നിരീക്ഷിച്ച് പട്ടിക 3.2 പൂർത്തിയാക്കൂ.

വരയുടെ നിറം, നീളം, വശങ്ങളുടെ എണ്ണം, കോണളവ്, വൃത്തത്തിന്റെ ആരം എന്നിവ വ്യത്യാസപ്പെടുത്തുമല്ലോ.

പൈത്തൺ നിർദേശം	ഉപയോഗം
color("red")	
circle(100)	100 യൂണിറ്റ് ആരമുള്ള വൃത്തം ലഭിക്കാൻ.
left(45)	
right(45)	
range(8)	

പട്ടിക 3.2 പൈത്തണിലെ ചില ഗ്രാഫിക്സ് നിർദേശങ്ങളും ഉപയോഗങ്ങളും

പൈത്തൺ ലിസ്റ്റുകൾ

പൈത്തണിലെ *ലിസ്റ്റ്* സങ്കേതം ഉപയോഗിച്ച് വിവിധ ഡാറ്റകൾ ഒറ്റ ചരത്തിൽ സൂക്ഷിക്കാൻ സാധിക്കും.

വിവിധ നിറങ്ങളുടെ പേരുകൾ colors എന്ന ചരത്തിൽ ഉൾക്കൊള്ളിച്ചിരിക്കുന്നതു നോക്കൂ.

colors=[“red”,“yellow”,“blue”,“green”,“orange”,“violet”]

ഒരു ലിസ്റ്റിലുള്ള വിവരങ്ങളെ, നാം പരിചയപ്പെട്ട for ലൂപ്പ് സങ്കേതം ഉപയോഗിച്ച് എളുപ്പത്തിൽ പ്രോഗ്രാമിൽ പ്രയോജനപ്പെടുത്താൻ സാധിക്കും. ചുവടെ നൽകിയ പ്രോഗ്രാം ടൈപ്പ് ചെയ്തു പ്രവർത്തിപ്പിച്ചുനോക്കൂ.

```
for clr in colors:
    print(clr)
```

വ്യത്യസ്ത നിറവും ആരവുമുള്ള വൃത്തങ്ങൾ നിർമ്മിക്കുന്നതിനുള്ള ഒരു പൈത്തൺ പ്രോഗ്രാം ചിത്രം 3.16 ൽ നൽകിയിരിക്കുന്നു. ഈ പ്രോഗ്രാം ടൈപ്പ് ചെയ്ത് പ്രവർത്തിപ്പിച്ച് ഔട്ട്പുട്ട് നിരീക്ഷിക്കൂ.

File Edit Format Run Options Window Help

```
from turtle import *

colors=["red", "yellow", "blue", "green", "orange" , "violet"]

radius=20

for c in colors:
        color(c)
        circle(radius)
        radius=radius + 10
```

ചിത്രം 3.16 വ്യത്യസ്ത നിറങ്ങളും ആരവുമുള്ള വൃത്തങ്ങൾ തയ്യാറാക്കുന്നതിനുള്ള പൈത്തൺ പ്രോഗ്രാം

വിലയിരുത്താം

- a=100
  a=a+25
  print(a)

  ഈ പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുമ്പോൾ ലഭിക്കുന്ന Output എന്തായിരിക്കും?

- 20 മുതൽ 40 വരെയുള്ള ഇരട്ടസംഖ്യകൾ print ചെയ്യുന്നതിനുള്ള പൈത്തൺനിർദേശം താഴെ തന്നിരിക്കുന്നവയിൽ ഏതാണ്?

  a) for i in range(20,42,2):
        print(i)
  b) for i in range(20,2,42):
        print(i)
  c) for i in range(20,42):
        print(i)
  d) for i in range(1,2,20):
        print(i)

തുടർപ്രവർത്തനങ്ങൾ

1. പൈത്തണിൽ വിവിധ ലിസ്റ്റുകൾ തയ്യാറാക്കി പ്രിന്റ് ചെയ്യുക.
   - ➢ കൂട്ടുകാരുടെ പേരുകൾ
   - ➢ പൂക്കളുടെ പേരുകൾ
   - ➢ സൗരയൂഥത്തിലെ ഗ്രഹങ്ങളുടെ പേരുകൾ

2. ഒരു മൈതാനത്തിന്റെ നീളം 120 മീറ്ററും വീതി 50 മീറ്ററുമായാൽ അതിന്റെ പരപ്പളവ് കാണുന്നതിനുള്ള പൈത്തൺ പ്രോഗ്രാം തയ്യാറാക്കുക.

3. മൂന്നു വ്യത്യസ്ത സംഖ്യകൾ input ആയി സ്വീകരിച്ച് അവയിലെ വലിയ സംഖ്യ പ്രിന്റ് ചെയ്യുന്നതിനുള്ള പൈത്തൺ പ്രോഗ്രാം തയ്യാറാക്കുക.

4. ഒരു വിദ്യാർഥിക്ക് ഒരു വിഷയത്തിന് 100 മാർക്കിൽ ലഭിച്ച സ്കോർ input ആയി സ്വീകരിച്ച് താഴെ കാണുന്ന നിബന്ധനകൾക്കനുസരിച്ച് ഗ്രേഡ് കണക്കാക്കുന്നതിനുള്ള പൈത്തൺ പ്രോഗ്രാം തയ്യാറാക്കുക.

   Score>=90 Grade= A+
   Score>=80 Grade=A
   Score>=70 Grade=B+
   Score>=60 Grade=B
   Score>=50 Grade=C+
   Otherwise print "Not Eligible"

5. വ്യത്യസ്ത നിറങ്ങൾ (നീല, പച്ച, ചുവപ്പ്, മഞ്ഞ) വശങ്ങളായുള്ള 200 യൂണിറ്റ് നീളവും 100 യൂണിറ്റ് വീതിയുമുള്ള ഒരു ചതുരം വരയ്ക്കുക.