Contents

1.1 Python သမိုင်း 2

1.2 Python Goals 4

1.3 ဘာကြောင့် Python က ထူးခြားနေတာလဲ။ 4

1.4 Python အသုံးပြုသည့် နည်းပညာ ထိပ်တန်းကုမ္ပဏီများ 5

1.5 ဘာကြောင့် Python ကို အသုံးပြုကြသလဲ 5

1.6 Features of Python 6

1.7 Python ပြိုင်ဘက် နှစ်ဦး 6

1.8 Python ကို အသုံးနည်းသည့် နေရာများ 7

1.9 Seven Reason Why You Should Learn Python 7

1.10 Python Interpreter 7

1.11 Compilation နှင့် Interpretation တို့၏ ကွာခြားချက် 8

1.11.1 Compilation နည်း 8

1.11.2 Interpretation နည်း 8

1.12 Python သည် scripting language တစ်ခုဖြစ်သည် 8

1.12.1 Scripting language ၏ အင်္ဂါရပ်များ 9

1.12.2 Scripting and Non-Scripting Languages တို့၏ ခြားနားချက်များ 9

1.13 Why Use Built-in Types? 10

1.14 Dynamically typed 11

1.15 Python Remains Most Popular Programming Language 11

1.16 Python Download and Installation Instructions 12

1.17 Python installation 12

1.18 Anaconda Installation 14

Chapter – 1 Introduction

1.1 Python သမိုင်း

Python သည် developer များအကြား ရေပန်းစားသည့် programming language တစ်ခု ဖြစ်လာသည်။ ရှင်းလင်းသည့် syntax နှင့် နားလည်ရန်လွယ်ကူသည့် code များကြောင့် လေ့လာခါစသူများ အထူးကြိုက်နှစ်သက်သည့် programming language ဖြစ်လာသည်။

၁၉၈၀ ပြည့်နှစ်များတွင် Python ဖြစ်စေမည့် အိုင်ဒီယာများ စတင် သန္ဓေတည်ခဲ့သည်။ တီထွင်သူ Guido Van Rossum သည် ၁၉၈၉ ဒီဇင်ဘာလတွင် စတင် အကောင်အထည် ဖော်ခဲ့သည်။ ဘီဘီစီ တီဗွီရှိုးမှ Monty Python's Flying ဇာတ်လမ်းတွဲမှ ယူထားသည့် **Python** စာလုံး ဖြစ်သည်။

၁၉၉၁ ခုနှစ် ဖေဖော်ဝါရီလတွင် version 0.9.0 ကို စတင် ထုတ်ဝေခဲ့သည်။

၁၉၉၄ ခုနှစ် ဇန်ဇဝါရီလတွင် version 1.0 ကို စတင် ထုတ်ဝေခဲ့သည်။

၂၀၀၀ ခုနှစ် အောက်တိုဘာလတွင် version 2.0 ကို စတင် ထုတ်ဝေခဲ့သည်။

၂၀၀၈ ခုနှစ် ဒီဇင်ဘာလတွင် version 3.0 ကို စတင် ထုတ်ဝေခဲ့သည်။

Python သည် high-level programming language ဖြစ်ပြီး နယ်ပယ်မျိုးစုံတွင် အလွယ်တကူ အသုံးချနိုင်သည့်(general-purpose) language တစ်ခု ဖြစ်သည်။ Python code များသည် အလွယ်တကူ ဖတ်ရှု နားလည်နိုင်ရုံသာမက ရေးရန်လည်း လွယ်ကူသည်။ ထိန်းသိမ်းမှုများ ပြုလုပ်ရန်လည်း လွယ်ကူသည်။ အလွယ်တကူ အသုံးပြုနိုင်သည့် standard library များစွာရှိသည်။ နယ်ပယ်မျိုးစုံတွင် သုံးရန်အတွက် third-party library များလည်း ရှိသည်။ တက်ကြွသည့် Python ပညာရှင်များ ကောင်းသထက် ပိုကောင်းအောင် အမြဲ ကြိုးစားပေး နေကြသည်။

Python သည် interpreted programming language ဖြစ်ခြင်း၊ program များကို compile လုပ်ရန် မလိုအပ်ခြင်းတို့ကြောင့် လွယ်ကူ လျှင်မြန်စွာ လုပ်ရမည့် prototyping လုပ်ငန်းများနှင့် လက်တွေ့စမ်းသပ်မှု ကိစ္စ(experimentation)များအတွက် အလွန်ထိရောက်သည့် programming language ဖြစ်လာသည်။

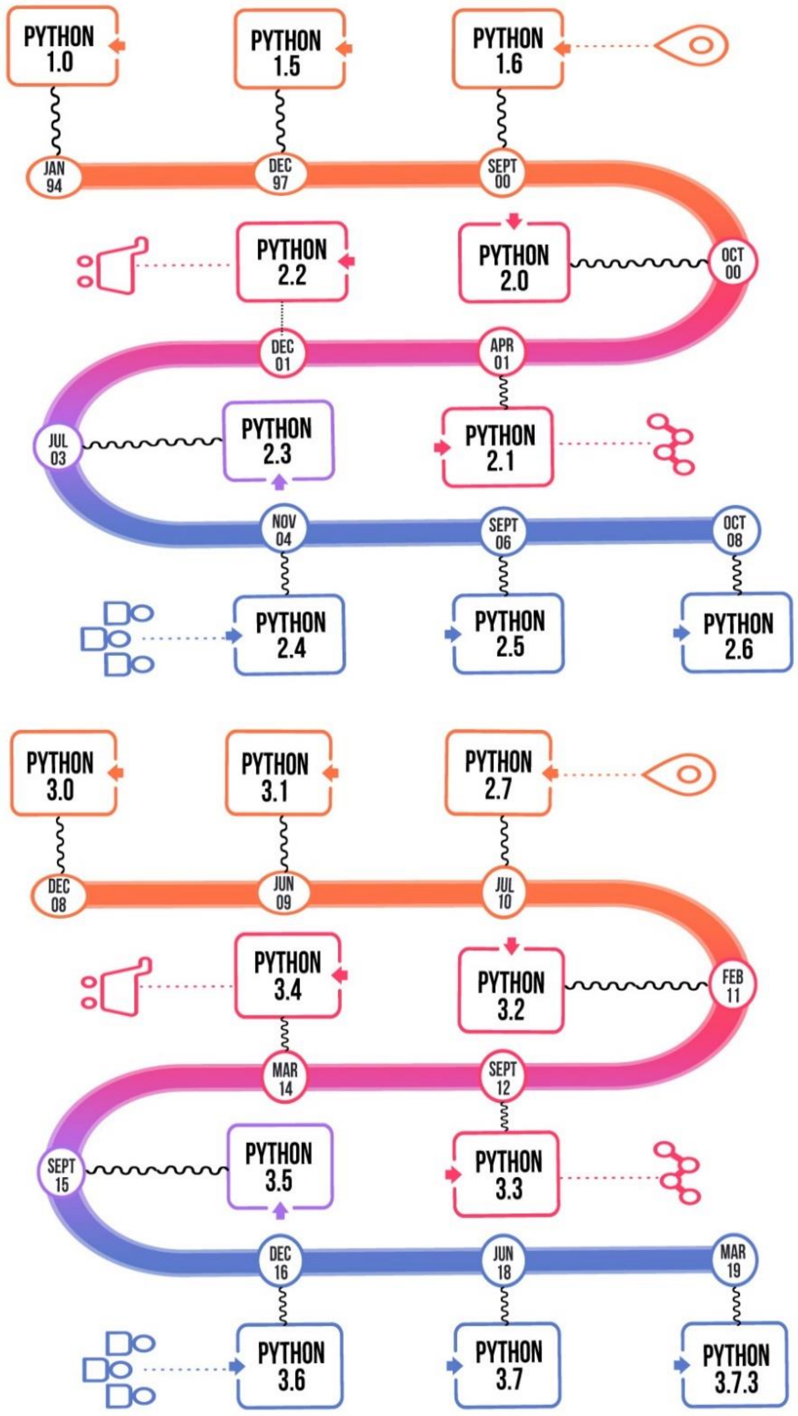
Python သည်

- Object-oriented programming
- Functional programming
- Imperative programming နှင့်
- Procedural programming ဖြစ်သည်။

တီထွင်သူ Guido Van Rossum

Operating system platform မျိုးစုံပေါ်တွင် Python ကို run နိုင်သည်။ Windows နှင့် GNU/Linux မှ စ၍ macOS အထိ operating system မျိုးစုံတွင် ပေါ်ပြူလာ ဖြစ်လာသည်။ တချို့ operating system များတွင် Python ကို OS အတူ ထည့်သွင်းပေးကြသည်။

Python version နှင့် ထုတ်ဝေခဲ့သည့် ခုနှစ်များ



Python သည် စွယ်စုံရသည့် programming language ဖြစ်ပြီး အလွယ်တကူ တိုးချဲ့နိုင်သည်။ Google(for YouTube), Uber, Facebook, နှင့် Mozilla စသည့် နည်းပညာ ကုမ္ပဏီများက Python ကို အဓိကထား အသုံးပြုကြသည်။

1.2 Python Goals

Python တီထွင်သူ Guido van Rossum က ရည်မှန်းချက်များ(goals)မှာ

၁။ တခြား language များကဲ့သို့ စွမ်အားထက်မြက်ပြီး သင်ယူရန် အလွန်လွယ်ကူရမည်။
၂။ Open source ဖြစ်ရမည်။ မည်သူမဆို ပါဝင် မျှဝေနိုင်ရမည်။
၃။ အင်္ဂလိပ်စာ ဖတ်နေရလို နားလည်နိုင်ရမည်။
၄။ အလုပ်မျိုးစုံအတွက် သင့်လျော်ရမည်။ အချိန်တိုအတွင်း သင်ယူ တတ်မြောက်နိုင်ရမည်။ စသည့် ရည်မှန်းချက်များ အတိုင်း တီထွင်ခဲ့သည်။

1.3 ဘာကြောင့် Python က ထူးခြားနေတာလဲ။

လေ့လာသင်ယူခါစသူများသာမက ကျွမ်းကျင်သူ ပညာရှင်များကပါ ဘာကြောင့် Python ကို အသုံးပြု နေကြတာလဲ? နည်းပညာ ကုမ္ပဏီတွေက သူတို့ရဲ့ အဓိကကျပြီး အရေးကြီးတဲ့ နေရာတွေမှာ Python ကို ဘာကြောင့် အသုံးပြုနေကြတာလဲ? စသည့် အချက်များစွာတို့ အနက်မှ အဓိကကျသည့် အချက်များမှာ

၁။ အလွယ်တကူ သင်ယူလေ့လာနိုင်ခြင်း(**it's easy to learn**)

Python ကို အချိန်တိုအတွင်း သင်ယူ လေ့လာနိုင်သည်။ ထို့ကြောင့် အချိန် အနည်းငယ် အတွင်း စတင် အသုံးပြုနိုင်သည့်သူ တစ်ယောက် ဖြစ်နိုင်သည်။

၂။ အလွယ်တကူ သင်ကြား ပို့ချနိုင်ခြင်း(**it's easy to teach**)

အလွယ်တကူ အသုံးပြုနိုင်ခြင်း၊ အလွယ်တကူနားလည်နိုင်ခြင်းတို့ကြောင့် သင်ကြားပို့ချရာတွင်လည်း လွယ်ကူသည်။ တခြားသူများကို သင်ကြားပေးရန် အလွယ်တကူသည်။

၃။ အလွယ်တကူ အသုံးပြုနိုင်ခြင်း (**it's easy to use**)

Python သည် operating system မျိုးစုံနှင့် အဆင်ပြေပြေ အသုံးပြုနိုင်သည်။ Python ပရိုဂရမ်များကို လျှင်လျှင် မြန်မြန် နားလည်နိုင်သည်။ ရေးရန် အချိန်တိုသည်။

၄။ အလွယ်တကူ နားလည်နိုင်ခြင်း (**it's easy to understand**)

Python ဖြင့် ရေးထားသည့် code များသည် တခြား language များထက်စာလျှင် အလွယ်တကူ လျှင်လျှင် မြန်မြန် နားလည်နိုင်သည်။

၅။ အလွယ်တကူ အခမဲ့ ရနိုင်ခြင်း နှင့် အဆင်ပြေစွာ စတင် အသုံးပြုနိုင်ခြင်း (**it's easy to obtain, install and deploy**)

တခြား language များနှင့် လုံးဝ မတူညီသည့်အချက်မှာ Python ကို အသုံးပြုရန်အတွက် တစ်ပြားမှ ပေးရန် မလိုပါ။ Open source ဖြစ်ပြီး operating system မျိုးစုံတွင် အသုံးပြုနိုင်သည်။

Multiplatform ဖြစ်သည်။ ဤအချက်သည် တခြား language များ လုံးဝ လိုက်မမီသည့်အချက် ဖြစ်သည်။ အားနည်း ချက်မှာ Python အလုပ်လုပ်သည့် မြန်နှုန်း နှေးသည်။ တချို့သော ကိစ္စများတွင် Python code များကို debugging လုပ်သည့်အခါ ခက်ခဲသည်။ သို့သော် Python တွင် အမှား ပြုလုပ် မိနိုင်သည့် အခြေအနေ အလွန် နည်းပါးသည်။

၆။ Simplify complex software development

ရှုပ်ထွေးသည့် software development အလုပ်များကို ရိုးရှင်းသည့်ကိစ္စများ ဖြစ်အောင်(simplify) ပြုလုပ်နိုင်သည်။

1.4 Python အသုံးပြုသည့် နည်းပညာ ထိပ်တန်းကုမ္ပဏီများ

1.5 ဘာကြောင့် Python ကို အသုံးပြုကြသလဲ

Code readability and Maintainable Code

၁။ Python code များကို အလွယ်တကူ ဖတ်ပြီးနားလည် နိုင်သည်။ ပရိုရမ်ကို ထိန်းသိမ်းရန်လည်း လွယ်ကူသည်။

၂။ Functional programming, aspect-oriented programming, object oriented programming နှင့် structured programming စသည့်တို့ကို အပြည့်အစုံ လုပ်နိုင်သည့် programming ဖြစ်သည်။ (Python Multiple Programming Paradigms)

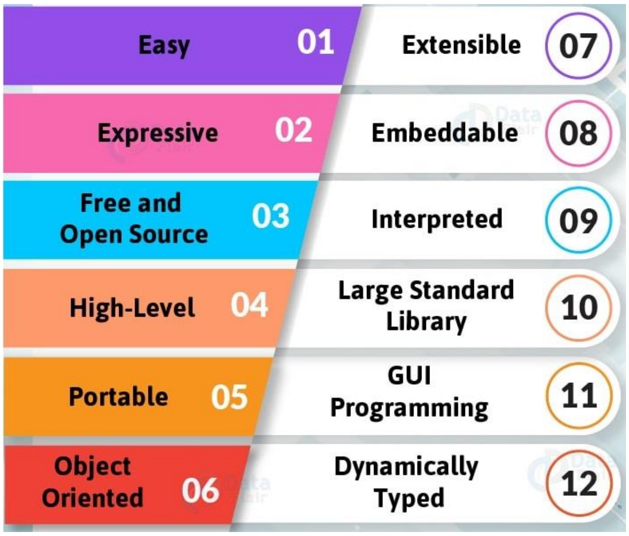
၃။ Python သည် operating system မျိုးစုံနှင့် အဆင်ပြေပြေ အသုံးပြုနိုင်(compatible)သည်။

၄။ အသုံးဝင်သည့် Standard Library မြောက်များစွာ ရှိသောကြောင့် မိမိ လုပ်လိုသည့် အလုပ်တစ်ခု တွက် မိမိ ကိုယ်တိုင် အစမှ စတင်ပြုလုပ်နေရန် မလိုပါ။

၅။ Open source framework နှင့် tool မြောက်များစွာ ရှိသည်။

၆။ ရှုပ်ထွေးသည့် software development အလုပ်များကို ရိုးရှင်းသည့် ကိစ္စများ ဖြစ်အောင် (simplify) ပြုလုပ် နိုင်သည်။

1.6 Features of Python



1.7 Python ပြိုင်ဘက် နှစ်ဦး

Python အား ယှဉ်ပြိုင်နိုင်သည့် ပြိုင်ဘက် နှစ်ဦးမှာ Perl နှင့် Ruby တို့ဖြစ်သည်။

Perl - a scripting language originally authored by Larry Wall;

Ruby - a scripting language originally authored by Yukihiro Matsumoto.

Perl သည် ဂန္ထဝင် C programming language နှင့် ပိုတူသည်။ ရှေးတုန်းက ထင်ရှားခဲ့ language များနှင့် ခပ်ဆင်ဆင် တူညီသည်။ Ruby သည် Python ကဲ့သို့ အိုင်ဒီယာအသစ်များထည့်သွင်းထားသည့်၊ ဆန်းသစ်ထားသည့် language တစ်မျိုး ဖြစ်သည်။ Python သည် Ruby လောက်လည်း မဆန်းသစ်သလို Ruby လောက်လည်း ရှေးမကျပေ။

1.8 Python ကို အသုံးနည်းသည့် နေရာများ

Python သည် တဖြည်းဖြည်း ပေါ်ပြူလာဖြစ်လာပြီး လူသုံးများလာသော်လည်း အသုံးနည်းသည် နယ်ပယ်များ ရှိနေသေးသည်။ Low-level programming အလုပ်များတွင် Python ကို သိပ်မသုံးကြပါ။ စွမ်းအား ကောင်းကောင်း လိုချင်သည့် graphic engine များ၊ driver များတွင် Python ကို မသုံးကြသေးပါဘူး။ ဟန်းဖုန်းတို့ တက်ဘလက်တို့လို mobile device တွေအတွက် application တွေကို ရေးဘို့ Python ကို မသုံးကြသေးပါဘူး။ နောင်တချိန်မှာ သုံးလာကြ လိမ့်မယ်လို့ မျှော်လင့်ထားကြသည်။

အထက်က နယ်ပယ်နှစ်ခုကလွဲရင် ကျန်နေရာတွေအားလုံးမှာ Python ကို တွင်တွင်ကျယ်ကျယ် အသုံးပြု ကြသည်။ ဒီစာအုပ်သည် Python 3 အတွက်သာ ဖြစ်သည်။ Python 2 က လွန်ခဲ့တဲ့ တချိန်က ဗားရှင်း အဟောင်း ဖြစ်သောကြောင့် မသက်ဆိုင်တော့ပါ။ အခု Python 3.8.xx ရောက်နေပြီ ဖြစ်သောကြောင့် Python 2 နှင့် သက်ဆိုင်သည့် အကြောင်း ဘာမှ ထည့်မရေးတော့ပါ။

1.9 Seven Reason Why You Should Learn Python

1.10 Python Interpreter

Python code တွေကို ကွန်ပျူတာက နည်းလည်ပြီး အလုပ်လုပ်နိုင်အောင် machine code တွေအဖြစ် ပြောင်းပေးရသည်။ Python လေ့လာသူ တစ်ယောက်အနေဖြင့် interpreter ဘယ်လို အလုပ် လုပ်သလဲ ဆိုတာ သိထား သင့်သည်။ Python interpreter က မှန်မှန်ကန်ကန် အလုပ်လုပ်ပေးစေရန် အတွက် ပရိုဂရမ် ရေးသည့်အခါ လိုက်နာရမည့် သတ်မှတ်ထားသည့် စည်းကမ်းချက်များ(set of rules)ကို လိုက်နာ ရသည်။ Programming language တိုင်းမှာ code တွေကို ရေးသည့်အခါ လိုက်နာရမည့် စည်းကမ်းတွေ ကိုယ်စီ ရှိကြသည်။ မှန်ကန်အောင် ရေးပါမှ(syntactically correct ဖြစ်ပါမှ) interpreter က အလုပ် လုပ်မှာပါ။

1.11 Compilation နှင့် Interpretation တို့၏ ကွာခြားချက်

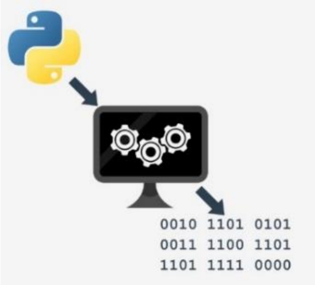
High-level programming language ပရိုဂရပ်များကို machine language အဖြစ် ပြောင်းရန်အတွက် compilation နည်း နှင့် interpretation နည်း ဆိုပြီးတော့ နည်း(၂)မျိုး ရှိသည်။

1.11.1 Compilation နည်း

Source program ကို တစ်ကြိမ် translate လုပ်သည်။ Source code ကို တစ်ခုခု ပြုပြင်ပြောင်းလဲလိုက်သည့် အခါတိုင်း ပရိုဂရမ် တစ်ခုလုံးကို translate လုပ်ရသည်။ (ဥပမာ- MS Windows တွင် run မည့် **.exe** file ဖြစ်လျှင် source code ပြောင်းတိုင်း compiler သို့မဟုတ် translator က translate လုပ်ပေးရသည်။) High-level programming language ပရိုဂရမ်များမှ machine language အဖြစ် translate လုပ်ပေးသောကြောင့် translator ဟုလည်း ခေါ်ဆိုသည်။ အသုံးပြုသူများကို **.exe** file ထုတ်ပေးသည်။ အသုံးပြုသူများ၏ စက်ထဲတွင် compiler ရှိနေရန် မလိုအပ်ပါ။

1.11.2 Interpretation နည်း

Interpretation နည်းကို သုံးလျှင် source code သာမက အသုံးပြုသူများ၏ စက်ထဲတွင် interpreter ရှိရန် လိုအပ်သည်။ အသုံးပြုသည့်အခါတိုင်း interpreter က source program အလုပ် လုပ်စေရန် (run ရန်) interpretation လုပ်ပေးရသည်။



Interpretation ၏ အားသာချက်

Code တွေ အပြည့်အစုံဖြစ်တာနဲ့ run ရန် အဆင်သင့်ဖြစ်နေသည်။ ပရိုဂရမ် တစ်ခုလုံး ပြီးသည့်အထိ စောင့်ရန် မလိုပါ။ Code တစ်ကြောင်းသည် ပြည့်စုံသည့် statement တစ်ခု ဖြစ်လျှင် run လို့ရပါသည်။ တချို့ code များသည် (၂)ကြောင်း၊ (၃)ကြောင်း ဖြစ်မှ ပြည့်စုံသည့် statement တစ်ခု ဖြစ်သည်။

Interpreter က code များကို run ပေးသောကြောင့် interpreter ရှိနေသည့် operating system များအားလုံးပေါ်တွင် run နိုင်သည်။ မတူညီသည့် operating system system အတွက် code များကို သီးခြားတစ်မျိုးချင်းစီ compile လုပ်ပေးနေရန် မလိုအပ်ပါ။

အားနည်းချက်

အလုပ်လုပ်သည့်နှုန်း နှေးသည်။ Interpretation နည်းသည် လျှင်မြန်စွာ အလုပ် မလုပ်နိုင်ပါ။ Code များကို run ရန် အတွက် အသုံးပြုသူ၏ ကွန်ပျူတာထဲတွင် interpreter ကို install လုပ်ထားရန် လိုသည်။

1.12 Python သည် scripting language တစ်ခုဖြစ်သည်

Scripting language သည် compile မလုပ်ဘဲ interpret လုပ်သည့် programming language တစ်မျိုး ဖြစ်သည်။ Scripting language များမှ code များကို execute လုပ်ရန် သီးခြား runtime execution လိုအပ်သည်။ အသုံးတွင်ကျယ်သည့် scripting language များမှာ Python, JavaScript, Perl, Visual Basic, ECMAScript, Bash and Unix Shell Scripts စသည်တို့ ဖြစ်သည်။ Java, C++, C တို့သည် Non-scripting language များ ဖြစ်ကြသည်။

1.12.1 Scripting language ၏ အင်္ဂါရပ်များ

- Scripting language များသည် အသုံးပြုရန် လွယ်ကူသည်။
- အခြေခံ ပရိုဂရမ်မင်း ဗဟုသုတရှိပါက code ကို လေ့လာရန်နှင့် မိမိကိုယ်တိုင်ရေးရန် လွယ်ကူသည်။
- သီးခြား IDE များ မလိုအပ်သောကြောင့် ကုဒ်ကို အလွယ်တကူ ရေးနိုင်သည်။
- ကုဒ်များကို ရေးသားနိုင်သည့် တည်းဖြတ်သည့် code editor များစွာ ရှိသည်။
- Object များသည် JavaScript ကဲ့သို့ object-oriented language များ နှင့် ဆင်တူသည်။
- Traditional programming language များ နှင့် နှိုင်းယှဉ်လျှင် ရေးရသည့်ကုဒ် လျော့နည်းသည်။
- Scripting language ကို အသုံးပြုရန်အတွက် memory allocation လုပ်ရန် မလိုအပ်ပါ။

1.12.2 Scripting and Non-Scripting Languages တို့၏ ခြားနားချက်များ

- လုပ်ငန်းသုံး ပရိုဂရမ်များတည်ဆောက်ရန် scripting language များတွင် အားကောင်း သောစွမ်းဆောင်ချက်များ ရှိသည်။
- Scripting language များသည် compile မလုပ်ဘဲ interpret လုပ်သည်။ Scripting language များသည် compile လုပ်ရန် မလိုပါ။ Non-scripting language များမှ ကုဒ်များကို machine code အဖြစ် ပြောင်းရန် compile လုပ်ပေးသောကြောင့် compiler လိုအပ်သည်။
- Scripting language များအတွက် interpreter လိုအပ်သည်။ Non-scripting language များအတွက် compiler လိုအပ်သည်။
- Scripting Language များသည် Non-scripting language များနှင့် နှိုင်းယှဉ်လျှင် နှေးကွေး(run slower)သည်။
- Scripting language များသည် တခြား program အတွင်း၌သာ ရှိနေနိုင်သည်။ Scripting language များကို တခြား program တစ်ခုဖြင့် wrap လုပ်သည် သို့မဟုတ် encapsulate ထားသည်။
- Scripting language များသည် code တစ်ကြောင်းချင်းစီကို အပေါ်ဘက်မှာ အောက်ဘက်သို့(top down) တစ်ခုပြီးမှ တစ်ခု(interpreted line by line sequentially) run သည်။ Non-scripting language များတွင် code အားလုံးကို တစ်ပြိုင်နက် compile လုပ်ပြီး machine code ထုတ်ပေးသည်။
- Scripting language များကို ရှိပြီးသား system များ သို့မဟုတ် application များနှင့် ပေါင်းစပ်၍ အသုံးပြုနိုင်သော်လည်း၊ Non-scripting language များကို အစမှ(scratch မှ) တည်ဆောက်ရသည်။
- Scripting language များသည် အခြားပရိုဂရမ်တစ်ခုအတွင်း၌ အလုပ် လုပ်သည်။ Non-scripting language များသည် ပြင်ပမှ ပရိုဂရမ်များ(parent or external programs)နှင့် မသက်ဆိုင်ဘဲ သီးခြား လွတ်လပ်စွာ run သည်။

Scripting language များသည် script ကို run ရန် လက်ခံထားမည့် host လိုအပ်သည်။ Non scripting language များအတွက် host မလိုအပ်သော်လည်း application သို့မဟုတ် web server ကို အသုံးပြုရန်အတွက် application ကို run ထားရန် လိုအပ်သည်။ Standalone application အနည်းငယ်သာ လွတ်လပ်စွာ run နိုင်သည်။

Python scripting language ကို web application လုပ်ရန်အတွက် Django ဟုခေါ်သော framework ကို အသုံးပြုနိုင်သည်။ Python တွင် functional, procedural, reflective, imperative စသည့် Object-Oriented feature များ ရှိသည်။ OOP အပြင် Python သည် Networking, Web frameworks, Graphical User Interfaces(GUI), Databases, Automation စသည့် အခြားသော လုပ်ဆောင်မှုများကို ကောင်းစွာ ထောက်ပံ့ ပေးသည်။ (Support လုပ်သည်။)

Python Conceptual Hierarchy

Python program များတွင် statement များ၊ expression များ နှင့် object များပါဝင်သည်။

၁။ ပရိုဂရမ်များကို module များဖြင့် ဖွဲ့စည်းတည်ဆောက်ထားသည်။ (**Programs are composed of modules.**)

၂။ Module များကို statement များဖြင့် ဖွဲ့စည်းတည်ဆောက်ထားသည်။ (**Modules contain statements.**)

၃။ Statement များ ထဲတွင် expression များ ပါဝင်သည်။ (**Statements contain expressions.**)

၄။ Expression များက object များကို ဖန်တီးနိုင်ပြီး၊ ပိုင်ဆိုင်ထားနိုင်သည်။ (**Expressions create and process objects.**)

1.13 Why Use Built-in Types?

Lower-level language များဖြစ်ကြသည့် C သို့မဟုတ် C++ တို့တွင် မိမိဖာသာ data structure များကို တည်ဆောက် ယူရသည်။ Memory structure များကို lay out ချခြင်း၊ memory များကို နေရာ ချခြင်း စသည့် ကိစ္စရပ်များကို ကိုယ်တိုင် ပြုလုပ်ရသည်။ အချိန်နှင့် လုပ်အား ပေးဆပ်ရသည်။ တစ်ခါ တစ်လေ အမှားတစ်ချို့ ပြုလုပ်မိကောင်း ပြုလုပ်မိ နိုင်သည်။

လေ့လာခါစသူများအတွက် ခက်ခဲသည့် ထိုကိစ္စရပ်များကို Python က ပြုလုပ်ပေးသည်။ Built-in object type များက ပြုလုပ်ပေးသည်။

၁။ Built-in object များက လွယ်ကူစွာ ပရိုဂရမ် ရေးနိုင်အောင် ပြုလုပ်ပေးသည်။ Collections (lists) ၊ search tables (dictionaries) စသည့် အလွန် အသုံးဝင်သည့် built-in object များရှိခြင်းကြောင့် ချက်ချင်း အသုံးပြုနိုင်သည်။ Builtin object type များကို အသုံးပြု၍ အလုပ်အားလုံးကို လွယ်ကူ လျှင်မြန်စွာ ပြီးမြောက် နိုင်သည်။

၂။ Built-in object များသည် တစ်ခါတစ်ရံ မိမိဖာသာ ပြုလုပ်ရသည့် data structure များထက် ပို၍ ထိထိရောက်ရောက် အလုပ် လုပ်ပေး နိုင်သည်။

၃။ Python’s built-in type များသည် optimize လုပ်ထားသည့် data structure algorithm ဖြစ်သောကြောင့် ထိရောက်စွာ အလုပ် လုပ်ပေးနိုင်ခြင်း ဖြစ်သည်။ Built-in type များကို မသုံးချင်ပါက သို့မဟုတ် အသုံးမတည့်ပါက မိမိဖာသာ ရေးသားနိုင်သည်။ Built-in object များ သည် Python language ၏ မရှိမဖြစ် အစိတ်အပိုင်းများ ဖြစ်ကြသည်။

Built-in object type များသည် လွယ်ကူစွာ ပရိုဂရမ် ရေးနိုင်အောင် ပြုလုပ်ပေးရုံသာမက အလွန် အစွမ်းထက်(more powerful)သည်။ သင့်ကိုယ်တိုင် ရေးထားသည့် ပရိုဂရမ်များထက် ထိထိရောက်ရောက် အလုပ် လုပ်ပေးနိုင်(more efficient ဖြစ်)သည်။ Python တွင် object တစ်ခု ပြုလုပ်ပြီးသည်နှင့် တစ်ပြိုင်နက် ၄င်းနှင့် သက်ဆိုင်သည့် operation များကို ချက်ချင်း ပြုလုပ်နိုင်သည်။ ဥပမာ - string တစ်ခု ပြုလုပ်ပြီသည်နှင့် တစ်ပြိုင်နက် ထို string နှင့် သက်ဆိုင်သည့် operation များကို ချက်ချင်း အသုံးချနိုင်သည်။

1.14 Dynamically typed

Python သည် dynamically typed ဖြစ်သည်။ Dynamically typed ဆိုသည်မှာ ဒေတာအမျိုးအစား (data type)များကို ပရိုဂရမ်ရေးသူက မသတ်မှတ်ပေးလျှင် Python က အလိုအလျောက် သတ်မှတ်ပေးသည်။ Python သည် dynamically typed ဖြစ်သောကြောင့် အက္ခရာများ(characters) ထည့်ပေးလိုက်လျှင် string data type ဟု အလိုအလျောက် သတ်မှတ်ပေးသည်။ ကိန်းပြည့်များ(integers) ထည်ပေးလိုက်လျှင် integers data type ဟု အလိုအလျောက် သတ်မှတ်ပေးသည်။ ဒသမကိန်းများ(decimal number)ထည့်ပေးလိုက်လျှင် float data type ဟု အလိုအလျောက် သတ်မှတ်ပေးသည်။

၁။ Variable တန်ဖိုးများ(values)ကို လိုက်၍ variable type ကို အလိုအလျောက် လိုက်ပြောင်း ပေးသည်။

၂။ Variable များကို အသုံးမပြုခင် variable အမျိုးအစား(type)ကို ကြိုတင်သိထားရန် မလိုအပ်ပါ။ တစ်နည်းအားဖြင့် variable type များအတွက် ကြိုတင်စီမံနေရန် စိတ်ပူနေရန် မလိုအပ်ပေ။

Variable(x) ထဲတွင် ရှိသည့် တန်ဖိုးများ(values) ပြောင်းလျှင် variable(x)၏ အမျိုးအစား လိုက်ပြောင်း နေသည်။

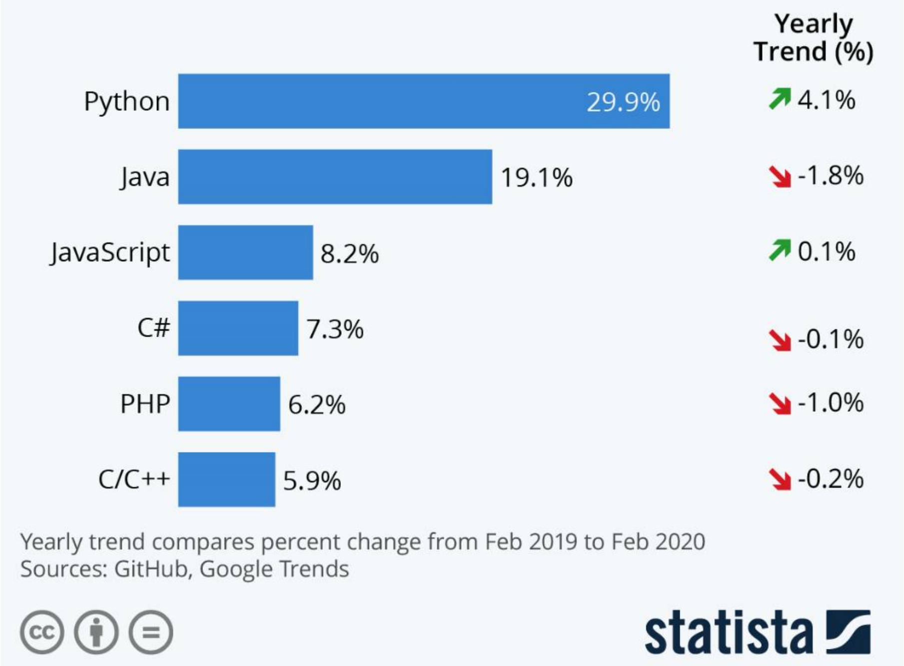
C, C++, Fortran နှင့် Java စသည့် static language များတွင် variable များကို အသုံးမပြုခင် variable အမျိုးအစား(type)ကို ကြိုတင်ကြေငြာ(declare) ပေးရသည်။

Special Variables

Python တွင် အရေးပါသည့် special variable များ ရှိသည်။ **True**, **False**, **None** နှင့် **NotImplemented** စသည်တို့ ဖြစ်သည်။ Python interpreter စတင်လိုက်သည်နှင့် တစ်ပြိုင်နက် ၄င်းတို့ တည်ရှိနေပြီ ဖြစ်သည်။

1.15 Python Remains Most Popular Programming Language

Google searche တွင် လူကြိုက်များသည့် programming language များကို အောက်တွင်ဖော်ထားသည်။



1.16 Python Download and Installation Instructions

လေ့လာခါစ သူတစ်ယောက်အတွက် Python ကို စတင်အသုံးပြုရန် စမ်းကြည့်ရန်အတွက် ဘာတွေ install လုပ်ရမလဲ? Anaconda တစ်ခုထဲ install လုပ်စရာလိုပါသည်။ တခြားဘာမှ ထပ်ပြီး တော့ install လုပ်စရာမလိုပါ။ Anaconda ထဲမှာ Jupyter notebook နှင့် PyCharm တို့ကို အဓိက ထားသုံးပါသည်။ Jupyter notebook မှာ ကောင်းတဲ့အချက်တွေ အများကြီး ရှိသည်။ ကုဒ်တစ်ကြောင်းတည်းကိုလည်း run လို့ရသည့် code editor မျိုး ဖြစ်သည်။ python file (**.py** file)များကို ဖွင့်ကြည့်ရန်အတွက်, run ရန်အတွက် PyCharm ကို သုံးပါ။

Anaconda ကို install လုပ်ထားရင် Python ကို သီးသန့် install လုပ်စရာမလိုပါဘူး။ Anaconda ထဲမှာ Python ကို ထည့်ပြီး install လုပ်ပေး လိုက်လို့ပါ။ Anaconda ကို install မလုပ်ဘဲ Python တစ်ခုတည်းကိုသာ install လုပ်ချင်ရင် အောက်ကအတိုင်း install လုပ်ပါ။

1.17 Python installation

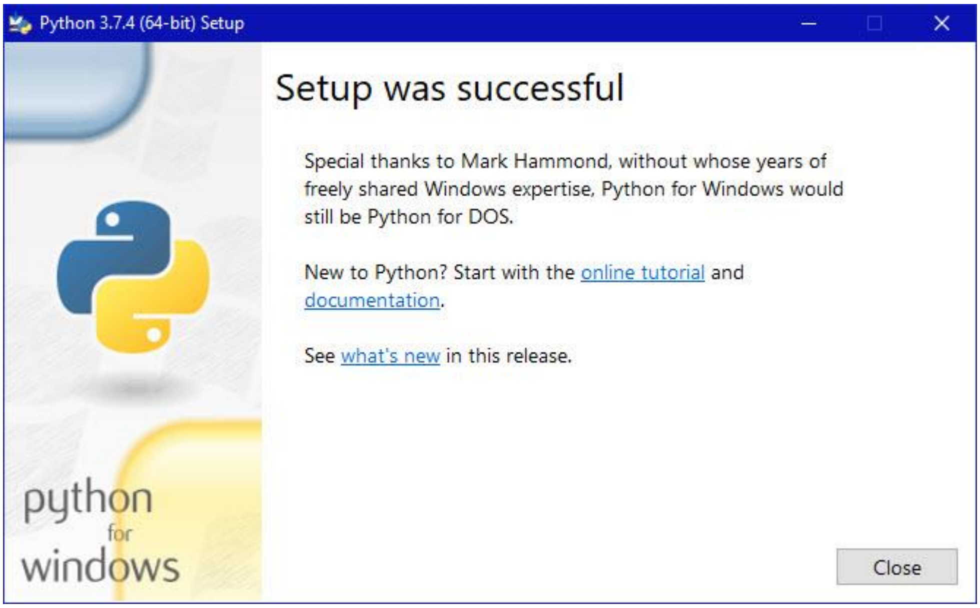
Windows 10 တွင် Python 3.7.4 installation လုပ်ပုံကို ဖော်ပြထားသည်။ Python: Version 3.7.4 ကို download လုပ်ရန် disk space 25 Mb လိုအပ်ပြီး install လုပ်ရန် disk space 90 Mb ခန့် လိုအပ်သည်။

https://www.python.org မှ download ရန် Download Python 3.7.4 button ကို ကလစ်ပါ။ ထို့နောက် installer ဖိုင်ကို မိမိနှစ်သက်သည့် နေရာတွင် save လုပ်ပါ။

Installing လုပ်ခြင်း

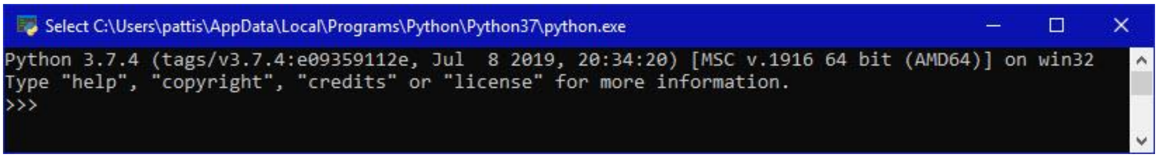
python-3.7.4-amd64.exe ကို Double-click လုပ်၍ installation လုပ်ခြင်းကို စတင်ပါ။ Python 3.7.4 (64-bit) Setup pop-up window ပေါ်လာပါလိမ့်မည်။

Install launcher for all users (recommended) နှင့် Add Python 3.7 to PATH checkboxes နှစ်ခု စလုံးကို check လုပ်ပါ။ Python version အဟောင်းတစ်ခုခုကို install လုပ်ထားခဲ့ဘူးလျှင် Upgrade Now ဆိုပြီး ပေါ်လာပါလိမ့်မည်။ A new Python 3.7.4 (64-bit) Setup pop-up window will appear.



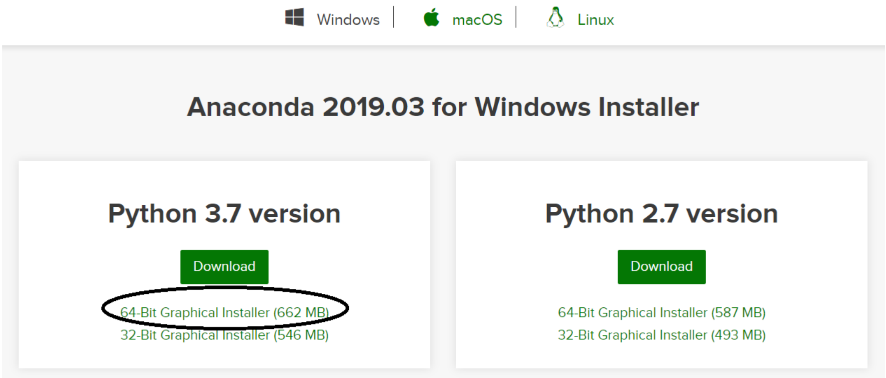
Installation ကို စစ်ဆေး(verify လုပ်)ရန်

directory C:\Users\Pattis\AppData\Local\Programs\Python\Python37 သို့မဟုတ် မိမိ install လုပ်ခဲ့သည့် directory ကို သွားပါ။ ထိုနောက် python.exe ဖိုင်အိုင်ကွန်(icon/file)ကို ကလစ်ပါ။ အောက်တွင် ဖော်ပြထားသည့် pop-up window ပေါ်လာပါလိမ့်မည်။

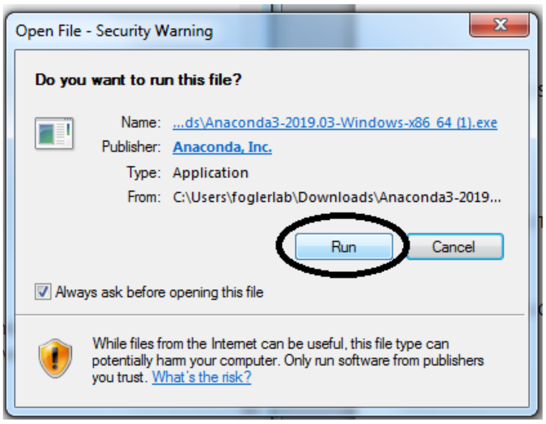


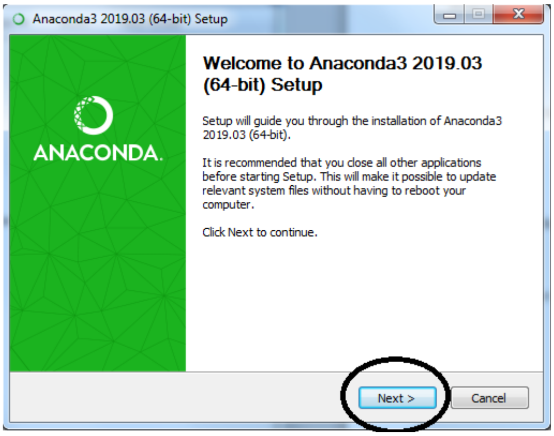
1.18 Anaconda Installation

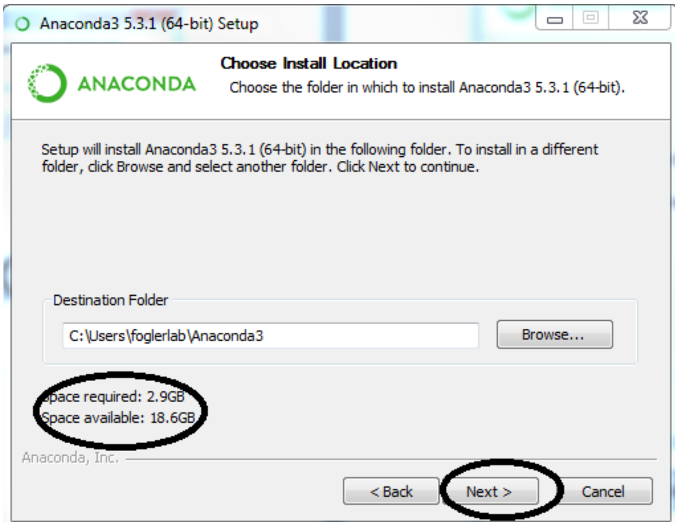
အောက်က address **https://www.anaconda.com** တွင် download လုပ်နိုင်ပါသည်။ **Anaconda ကို install လုပ်မယ်ဆိုရင် Python ကို သီးသန့် install လုပ်စရာမလိုပါဘူး။**



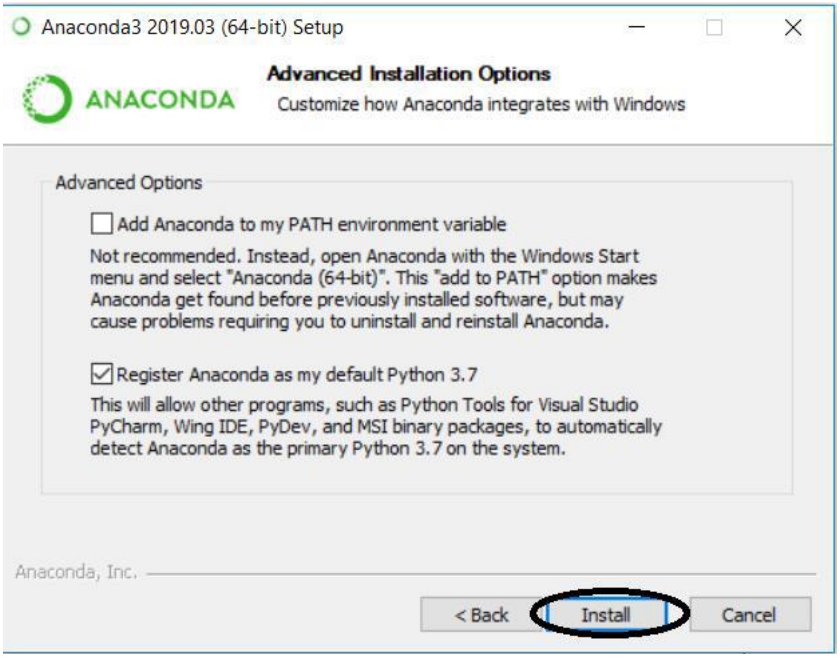
Download လုပ်ပြီးပါက click ၍ install လုပ်ပါ။



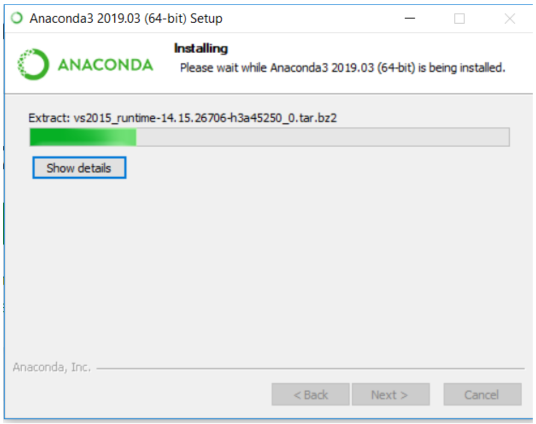


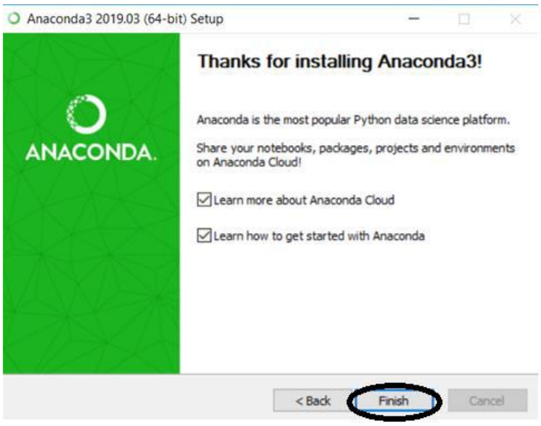


3GB ခန့်လိုအပ်သည်ကို သတိပြုပါ။



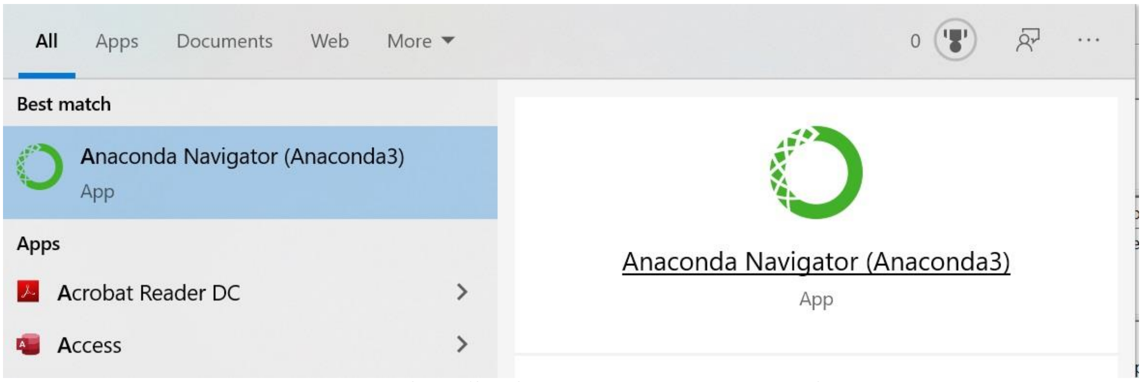
Showing the progress of installation





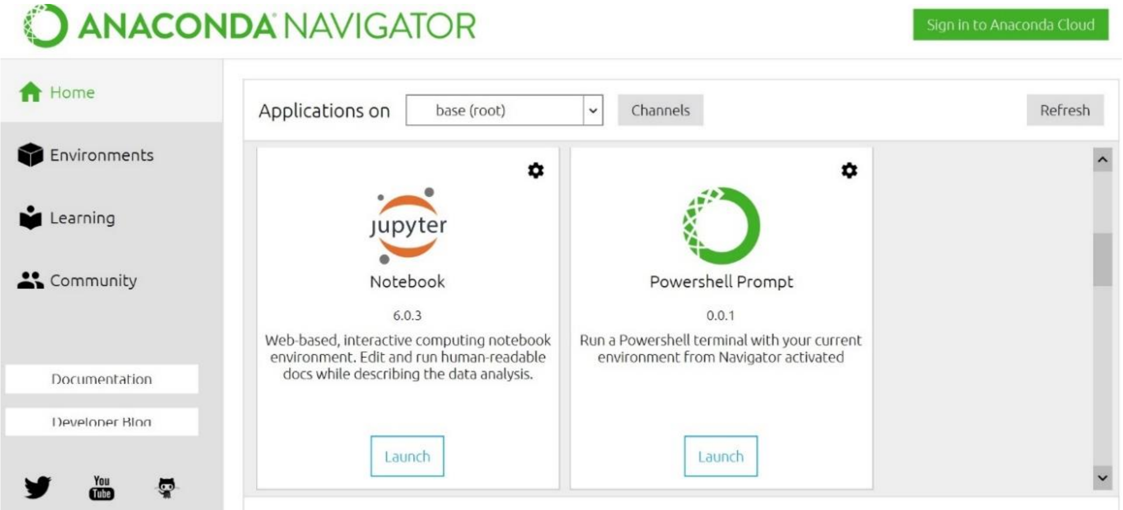
Click on Finish button to complete the installation . စတင် အသုံးပြုလို့ ရပါပြီ။

Window 10 မှာ Anaconda Navigator 1.9.12 ကို စသုံးဘို့ ဖွင့်မှာပါ။

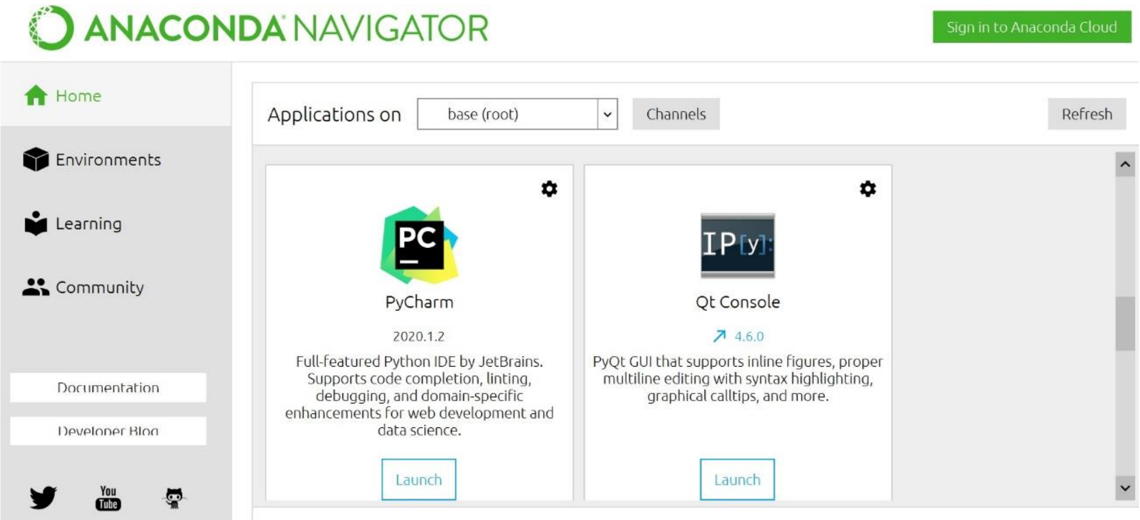


Anaconda Navigator တက်လာပြီးရင် Jupyter Notebook သုံးရန်

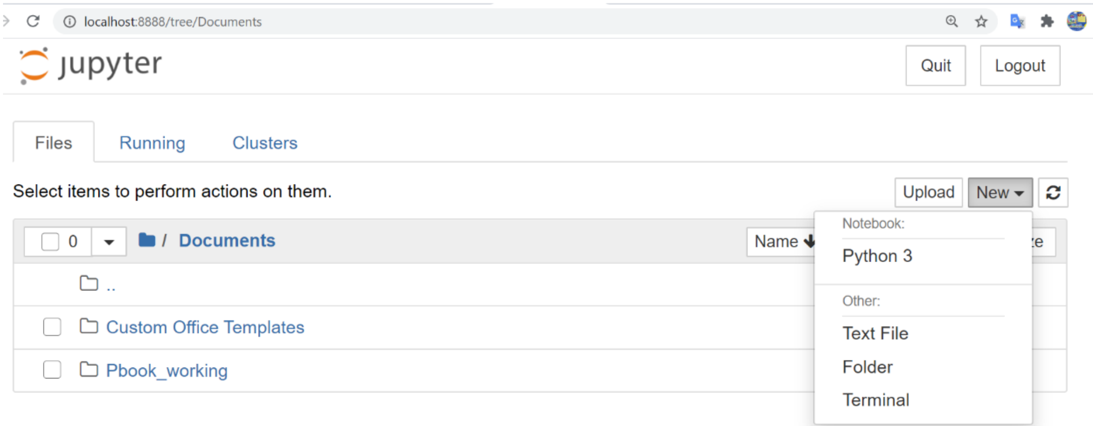
Jupyter Notebook သုံးရန်



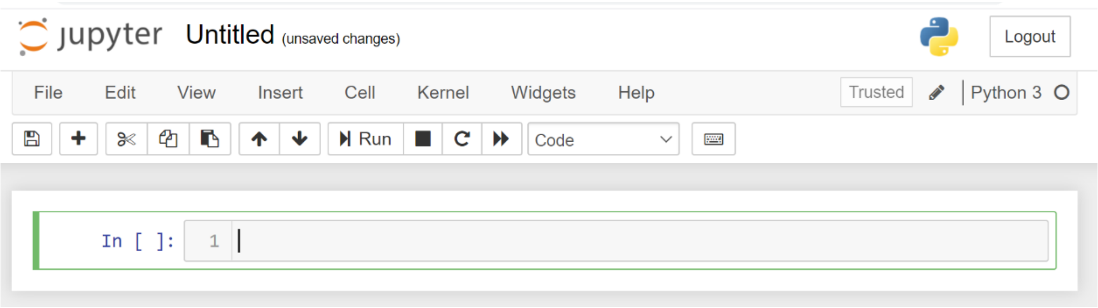
PyCharm သုံးရန်



အခု Jupter စသုံးလို့ ရပါပြီ။ ဖိုင် အသစ်ဖွင့်ဘို့ New ကို ကလစ်ပြီး Python 3 ကို ကလစ်ပါ။



ဖိုင်အသစ်ကို ဖိုင်နာမည်ပေးပြီး စသုံးလို့ ရပါပြီ။



-End-