



Python Programming



Course Overview

Introduction

- ✓ Overview
- ✓ Why python
- ✓ Versions

Setup

- ✓ Python installation
- ✓ Environment setup
- ✓ Git configuration

Data Structures

- ✓ Basics
- ✓ Collections

Operators

- ✓ Basic operators
- ✓ Chained comparison
- ✓ Mathematical, logical etc

Statements

- ✓ Control statements
- ✓ Loops

Functions

- ✓ Named vs Lambda
- ✓ Nested functions
- ✓ Scopes

Advanced features

- ✓ List comprehension
- ✓ Generators
- ✓ Decorators

OOP

- Class vs Object
- Inheritance
- Special methods



Course Overview

cloud ← AWS →
← GCP →
← Azure →

Modules and packages

- ✓ Module creation
- ✓ Custom packages
- ✓ Package ecosystem

Functional Programming

- Map, filter and reduce
- ✓ Environment setup
- ✓ Git configuration

Web Services

- ✓ Fundamentals of WS
- ✓ WS using Flask

Numpy

- ✓ Requirements
- ✓ Data types
- ✓ Data manipulation

Pandas

- ✓ Series and Data Frames
- ✓ Data manipulation

Data Analysis

- ✓ Fundamentals
- ✓ Requirements
- ✓ Using numpy and Pandas

Data visualization

- ✓ Fundamentals
- Using Matplotlib
- Using Seaborn

Testing

- ✓ Fundamentals
- ✓ BeautifulSoup
- ✓ Selenium



computer
→ compute
→ process

Language Fundamentals



What is a computer language?

- A language is a medium to interact with computer
- A language is used to solve a problem by giving some solution (writing a program)
- Types
 - **Machine languages**: the code written in 0s and 1s and can be directly executed by CPU
 - **Assembly languages**: a language closely related to one or a family of machine languages, and which uses mnemonics to ease writing
LOAD, ADD
 - Programming languages
 - **General purpose languages**: a programming language that is broadly applicable across application domains, and lacks specialized features for a particular domain C, C++, python
 - **Markup languages**: a grammar for annotating a document in a way that is syntactically distinguishable from the text, tags/attributes → HTML, XML, SGML
 - **Stylesheet language**: a computer language that expresses the presentation of structured documents CSS
 - **Configuration language**: a language used to write configuration files YAML, JSON, XML
 - **Query language**: a language used to make queries in databases and information systems -SQL
 - **Scripting languages**: a language used to write simple to complex scripts
↳ python, perl, php, bash



Low Level vs High Level Languages

<u>✓ High Low Level Language</u>	<u>✓ Low High Level Language</u>
It is <u>programmer friendly language</u>	It is a <u>machine friendly language</u>
High level language is <u>less memory efficient</u>	Low level language is <u>high memory efficient</u>
It is <u>easy to understand</u>	It is <u>tough to understand</u>
It is <u>simple to debug</u>	It is <u>complex to debug comparatively</u>
It is <u>simple to maintain</u>	It is <u>complex to maintain comparatively</u>
It is <u>portable</u>	It is <u>non-portable</u>
It can <u>run on any platform</u>	It is <u>machine-dependent</u> ↗
It needs compiler or interpreter for translation	It needs assembler for translation
<u>E.g. Python</u>	<u>E.g. Assembly</u>



Compiled language → code executes using one executable

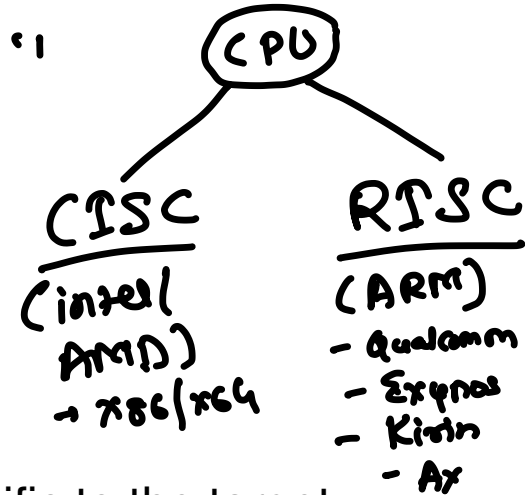
- A programming language which involves an executable to execute the logic instead of executing the source code directly
- E.g. C, C++, Pascal, Objective-C, Swift etc.
- Stages

✓ Pre-processing

- Used to preprocess the code
- Comments removal →
- Code expansion → macro #define
- Conditional compilation

headers
modules
package

#include " " " "



✓ Compiling

- Compiler is used to translate pre-processed code into assembly instructions specific to the target processor architecture

object code - asm

✓ Assembling

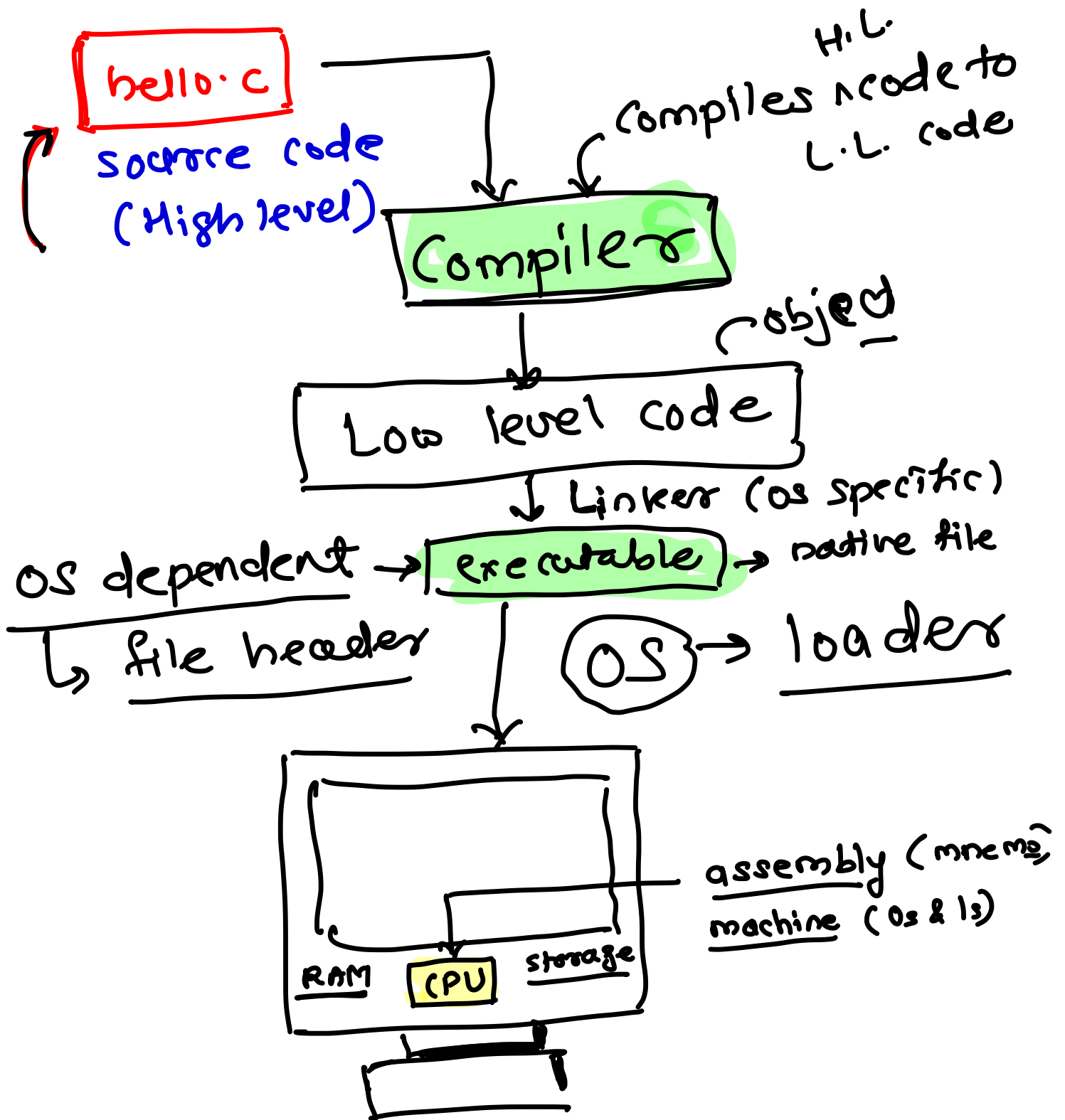
- Assembler is used to translate the assembly instructions to object code

65 13

✓ Linking

- Linker used in this stage re-arranges the code and insert missing files to emit an executable





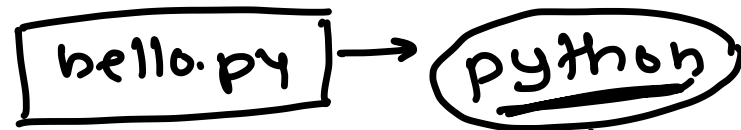
Executable file format

- ① macOS : Mach-O3 ←
- ② Linux : ELF
- ③ Windows : PE / COFF ←

magic number
↳ OS specific

Interpreted Language

- A language mostly executes source code directly and freely, without previously compiling a program into an executable
- E.g. Python, JavaScript, Perl, BASIC etc.
- Interpretation
 - Interpreter used in the language executes the high level source code directly



Interpreter

↳ interpretes the code

↳ reads the code line by line , from
executes top to bottom

→ slower than compiled languages

→ OS independent
(write once & run anywhere)

→ source code gets executed



Python



Introduction

- Python is a general purpose, high level and interpreted language
- It is dynamically typed and garbage collected language
- It supports programming paradigms like
 - ✓ Procedural programming
 - ✓ Object oriented programming
 - ✓ Functional programming
 - ✓ Aspect oriented programming (metaprogramming and magic metaobjects)
- Heavily dependent on indentation to create blocks which makes it very easy to read the code
- It has more than 130,000 packages included with wide range of functionality
 - ✓ Graphical user interfaces
 - ✓ Web frameworks
 - ✓ Networking
 - ✓ Automation
 - ✓ Web scraping



History

- Python was conceived in the late 1980s by Guido Rossum in Netherlands
- It is considered as a successor to the ABC language (itself inspired by SETL)
- Its implementation began in December 1989 ← 1.0 → 1991
- Van Rossum continued as Python's lead developer until July 12, 2018
- When he announced his permanent vacation from his responsibilities, a team of five members was developed in Jan 2019 to lead the project



Versions

- Version 0.9.0 [Feb 1991] ←
 - Having features like classes with inheritance, exception handling, functions etc.
 - One of the major versions of python
- Version 1 [Jan 1994]
 - The major new features included in this release were the functional programming tools like lambda, map, filter, reduce.
 - The last version released was 1.6 in 2000
- Version 2 [Oct 2000]
 - Introduced features like list comprehension, garbage collection, generators etc.
 - Introduced its own license known as Python Software Foundation License (PSF)
 - The last version released was 2.7.16 in Mar 2019
- Version 3 [Dec 2008]
 - Python 3.0 is also called "Python 3000" or "Py3K"
 - It was designed to rectify fundamental design flaws in the language
 - Python 3.0 had an emphasis on removing duplicative constructs and modules
 - The last version released was 3.7.4 in Jul 2019

DS

print "hello"
MIT/GPL
print ("hello")



Environment Setup

- Windows:
 - Download python installer
 - Download pycharm community edition
- Ubuntu
 - Install python3 on using apt installer
 - Download pycharm community edition
- macOS
 - Download python 64 bit installer
 - Download pycharm community edition



What have you downloaded

- Python has many versions like

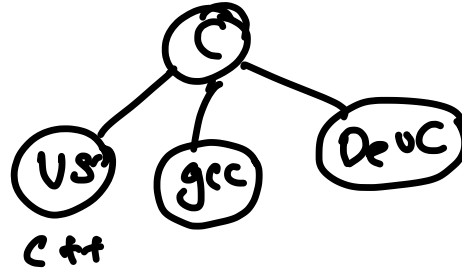
✓ CPython

✓ Pypy

✓ Jython

✓ IronPython

python



- The standard version is called as CPython which is what most of us get when we download from the official site
- Henceforth when we say we are using python, it will be CPython



Hello world program explained

- Python is not a compiled language: misconception
- CPython has a compile base
- Python compiles into bytecodes
- CPU can not understand bytecodes
- We need an interpreter to understand and execute the bytecodes
- This interpreter is nothing but the python virtual machine



Python Virtual Machine (PVM)

- Written in C
- Compiles the bytecode into machine language
- It emulates the machine or CPU
- Executes bytecodes similar to the way a CPU executes the machine instructions



hello.py → python3

① Lexical analyzer -

- Syntax checking
- tokens (identifiers)



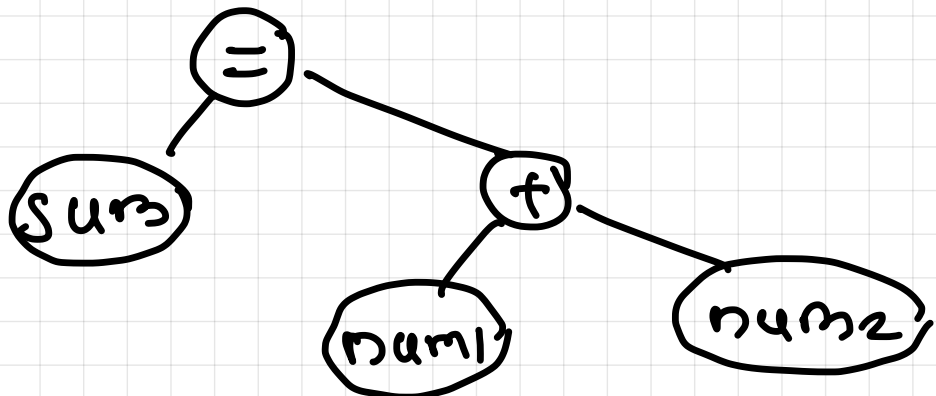
- symbol table



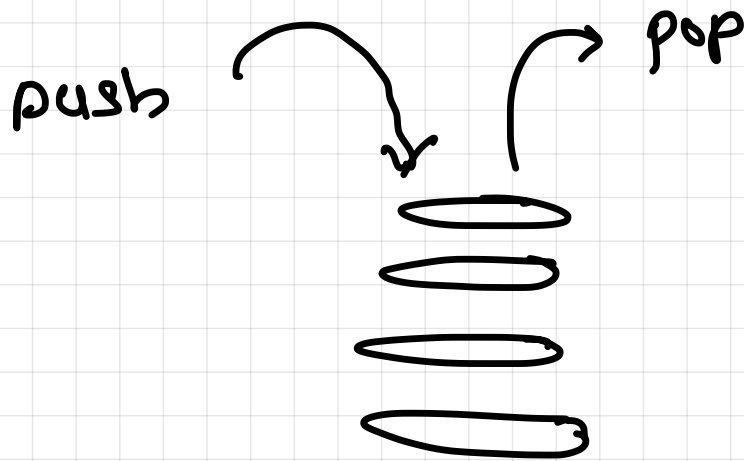
Symbol	Def.	Add

② AST - Abstract Syntax Tree

sum = num1 + num2



③ Stack



④ byte codes → similar to machine code

- ① CPU does not understand byte code
- ② It is not meant for CPU
- ③ It is meant for PVM

