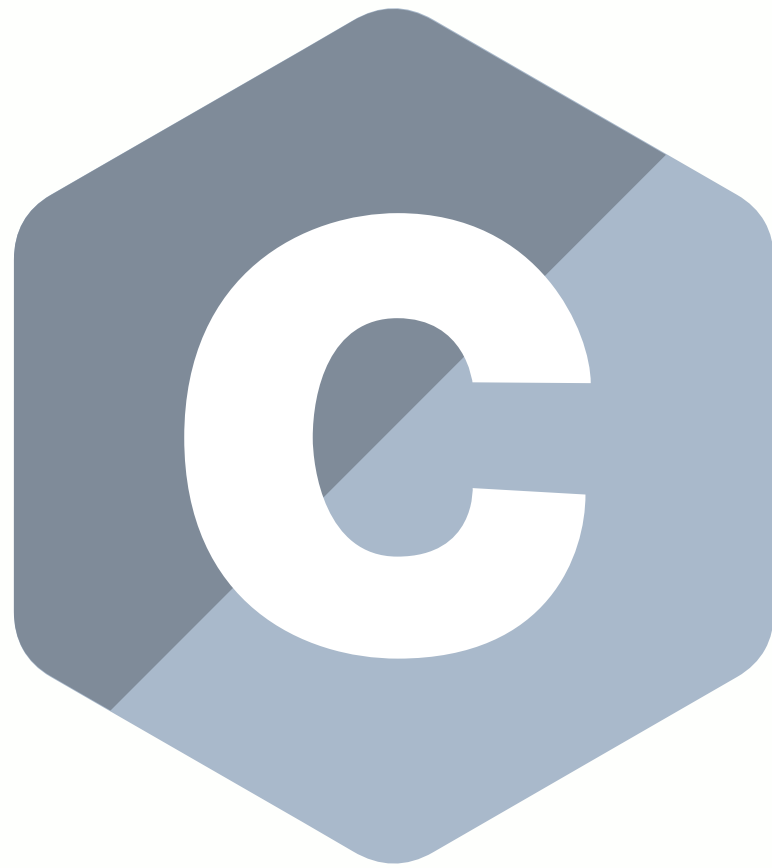


# SWIG

SIMPLIFIED WRAPPER AND INTERFACE GENERATOR

# NumPy



# WHAT?

Спрощений генератор обгортки та інтерфейсів

- Open Source програмний інструмент
- Створення **wraper-ів програм або бібліотек**, написаних на C або C ++, із скриптовими мовами, такими як Lua, Perl, PHP, Python, R, Ruby, Tcl та іншими мовами як-от C #, Java, JavaScript, Go, D, OCaml, Octave, Scilab та Scheme
- SWIG може аналізувати header файли і, використовуючи лише **code prototypes**, створювати **інтерфейс для певної мови** програмування
- **Initial release** - February 1996; 25 years ago.  
**Stable release** - 4.0.2 / June 8, 2020
- **Written in** -- C, C++

# Thoughts about different programming languages?

- Простіше використовувати різні мови програмування для різних завдань
- Наприклад, **написання графічного інтерфейсу** користувача на скриптових мовах
- Мови програмування мають **різні сильні та слабкі сторони**
- Як використовувати найкращі функції кожної мови
- SWIG дозволяє **зосередитись на основній програмі C** та використовувати мовний інтерфейс на високому рівні



# КІЛЬКА СКЛАДНОЩЕЙ З ПРОГРАМУВАННЯМ НА C / C ++

1

Написання **користувальницького інтерфейсу** досить болісно (тобто врахуйте програмування з MFC, X11, GTK або будь-якою кількістю інших бібліотек).

2

**Тестування** займає багато часу (цикл компіляції / налагодження).

3

**Налаштування**: не легко налаштувати або налаштувати без перекомпіляції.

4

**Модуляризація** може бути складною.

5

Побоювання щодо **безпеки** (наприклад, переповнення буфера).

# ЧОМУ ВАРТО ВИКОРИСТОВУВАТИ SWIG?

1

Побудова **інтерпретованого інтерфейсу** для існуючих програм C/C++

2

Побудова **високоєфективних** модулів C/C++ для скриптових мов

3

**Написання обгортки** на деяких скриптових мовах, таких як Python, де дуже просто написати тест тощо

4

Програмування систем та **інтеграція** систем.

5

Відмінна **підтримка** у написанні бібліотек програмування.

6

Висока **продуктивність** (подрібнення чисел, обробка даних, графіка тощо).

# How does SWIG work internally?



# АЛЬТЕРНАТИВИ

Подібну функціональність для Python пропонують SIP та бібліотека Boost's Boost.python:



## SIP

Програмний інструмент з відкритим кодом, який використовується **для з'єднання комп'ютерних програм** або бібліотек, написаних на C або C ++, з скриптовою мовою Python



## Boost

Набір бібліотек для мови програмування C ++, яка забезпечує підтримку завдань та структур, таких як **лінійна алгебра, генерація псевдовипадкових чисел**, багатопотоковість, обробка зображень, регулярні вирази та тестування одиниць.



# ПРОЕКТИ, ЩО ВИКОРИСТОВУЮТЬ SWIG

1

TensorFlow

(для машинного навчання, таких як нейронні мережі)

2

Apache SINGA

(для масштабованого розподіленого навчання,  
розширюється для роботи в широкому спектрі  
апаратних засобів)

3

NumPy

4

QuantLib

5

GNU Radio

# ДЖЕРЕЛА ДЛЯ ПОЧАТКУ ВИКОРИСТАННЯ SWIG

- <http://www.swig.org/Doc3.0/Introduction.html> - official documentation of SWIG
- <https://www.geeksforgeeks.org/wrapping-cc-python-using-swig-set-1/> - good example how to start SWIG usage with its installation
- <https://numpy.org/doc/stable/reference/swig.html> -- NumPy and SWIG
- <https://github.com/Mizux/cmake-swig> - how to add SWIG in Cmake. Examples for different programming languages
- <http://www.swig.org/Doc4.0/SWIGDocumentation.html> - all documentation for building for different languages
- <https://github.com/swig/swig/tree/master/Examples/javascript/simple> , <https://github.com/paularmstrong/swig> - JS та SWIG
- <https://github.com/swig/swig> - official Github repository