MeltingCon 2018

'C++로 Python API 만들기'

옥찬호

C++ Korea, Microsoft MVP

utilForever@gmail.com

소개

- 옥찬호 (Chris Ohk)
  - Microsoft VSDT MVP
    (Visual Studio and Development Technologies)
  - 페이스북 그룹 C++ Korea 대표
  - IT 전문서 집필 및 번역 다수
    - 게임샐러드로 코드 한 줄 없이 게임 만들기 (2013)
    - 유니티 Shader와 Effect 제작 (2014)
    - 2D 게임 프로그래밍 (2014)
    - 러스트 핵심 노트 (2017)
    - 모던 C++ 입문 (2017)


WORLD OF WARCRAFT
BATTLE FOR AZEROTH

시작하기 전에…

- C++ 코드로 Python API를 만드는 방법을 설명합니다.
- C++ 언어에 대한 기본 지식이 필요합니다.

  (클래스, 상속, 다형성, 가상 함수 등)
- 예제 코드는 Github 저장소에서 다운로드 받을 수 있습니다.

  (https://github.com/utilForever/MeltingCon-CppPython)

C++ 프로젝트

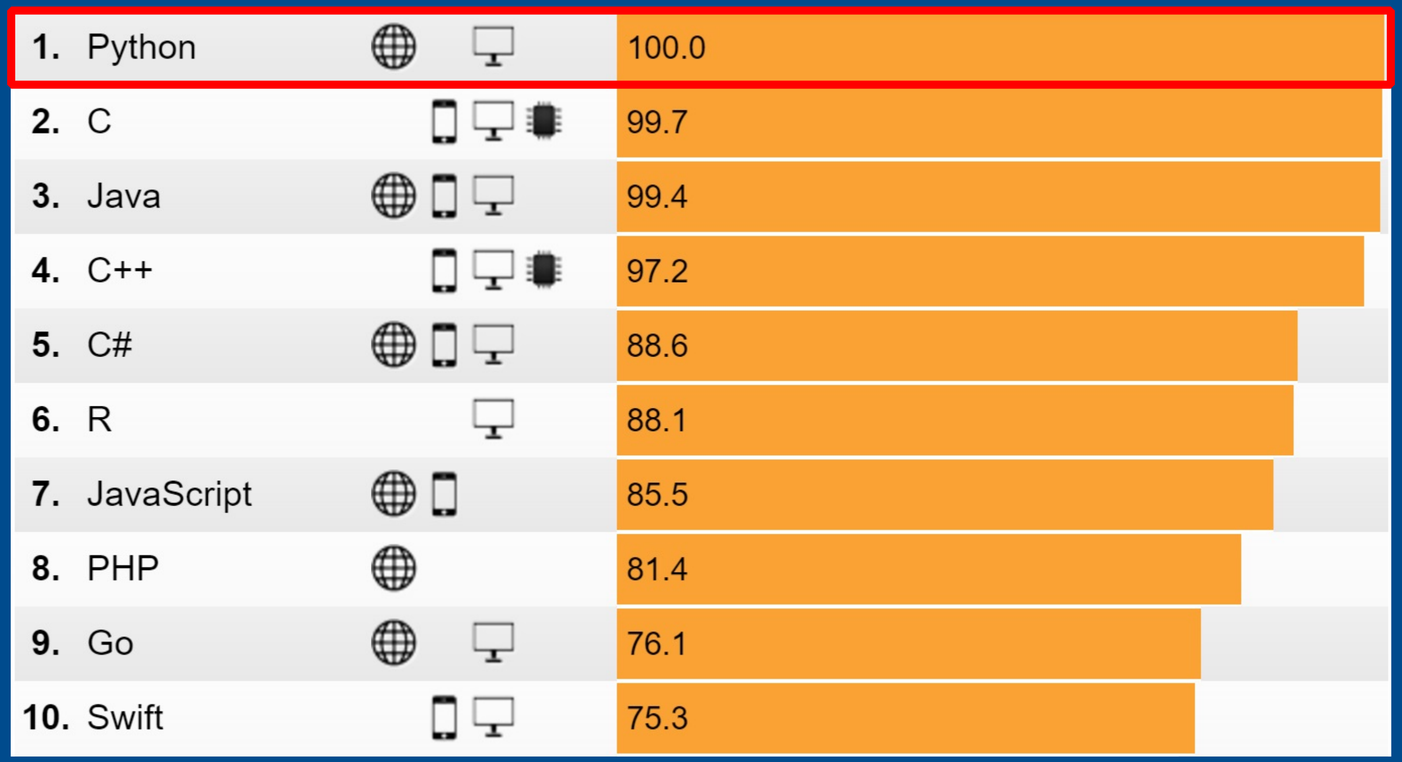
- 현재 다양한 C++ 프로젝트를 진행하고 있습니다.
  - fluid-engine-dev : https://github.com/doyubkim/fluid-engine-dev
  - CubbyFlow : https://github.com/utilForever/CubbyFlow-v0
  - Hearthstone++ : https://github.com/utilForever/Hearthstonepp
  - Civilization++ : https://github.com/utilForever/Civilizationpp

API 지원 문제

- C++ 언어로 만든 라이브러리는 C++에서만 사용 가능합니다.
- 고생해서 만든 라이브러리를 C++에서만 사용하기엔 너무 아깝습니다.
- 다른 언어에서도 라이브러리를 사용할 수 있으면 좋겠습니다.
- 사람들이 많이 사용하는 언어를 먼저 지원하면 어떨까요?

최고존엄 Python

MeltingCon 2018
Python API using C++



Programming language rankings and image by IEEE Spectrum

API 설계 방침

- C++ 코드를 최대한 재사용할 수 있으면 좋겠습니다.

  (프로그래밍 언어마다 다시 만드는 작업은 너무 무겁습니다.)

- 확장 가능성을 고려해야 합니다.

  (새로운 클래스가 추가될 때마다 바인딩하기 어렵습니다.)

- API 설계 방침을 만족하면서 Python API를 만들 수는 없을까요?

pybind11

MeltingCon 2018
Python API using C++

- https://github.com/pybind/pybind11
- C++ 코드를 기반으로 Python 모듈을 생성해주는 라이브러리
- 열거체, 단일/다중 상속, 순수 가상 함수 등 C++ 핵심 기능들을 지원
- STL 자료 구조, 반복자, 스마트 포인터 등 표준 라이브러리도 지원
- Python 2, 3 및 PyPy 지원

pybind11 사용법

- git에 새로운 저장소를 만든 뒤 clone하세요.

```
git clone <repo>
```

- pybind11을 submodule로 추가하세요.

```
git submodule add https://github.com/pybind/pybind11 <path>
```


!선거짜잔!

pybind11 사용법

- 이제 사용하면 됩니다. 어떻게 사용하냐구요? 코드로 직접 보겠습니다.
  - 함수
  - 클래스
    - `public` 멤버 접근, `private` 멤버 접근
    - 순수 가상 함수, 오버라이딩
  - STL
    - 컨테이너 (`std::vector`, `std::map`, `std::set`, `std::array`, `std::valarray`)
    - 함수 포인터 (`std::functional`)
    - 시간 (`std::chrono`)

감사합니다

http://github.com/utilForever

질문 환영합니다!