

Python in Internet Business – Presentation Note

장혜식

<알림> 이 자료는 Google을 대표하여 Greg Stein이 PyCon 2005와 SDForum 2006에서 발표한 내용과 Pythonology 등의 사이트에서 각 회사들이 공개한 정보를 토대로 작성되었지만 의도치않게 일부 사실과 다른 내용이 포함 되어 있을 수도 있습니다.

Python has been an important part of Google since the beginning, and remains so as the system grows and evolved. - Peter Norvig, Director of Search Quality at Google

기업에서 파이썬이 사용되는 곳

- Google
 - C++, Java, Python을 공식 언어로 사용
 - 웹 프론트엔드보다는 백그라운드 작업이나 관리 작업에 많이 사용함
- Microsoft
- NASA
- NYSE
- 국내 ISP들..
- 여러 게임 회사들..

I had lamented that it was a long time since I had a practical hacking environment, and Dan Connolly suggested Python as something you could start quickly but which would scale to a large system. One day, 15 minutes before I had to leave for the airport, I got my laptop back out of my bag, and sucked off the web the python 1.6 system and the python tutorial, and a copy of a small notation3 parser Dan had hacked together. I was happy to find that Python is a language you can get into on one battery! I have been happily hacking ever since. - Tim Berners-Lee, the inventor of the world wide web

오픈소스 프로젝트에서의 파이썬의 위치

- GNOME
- KDE
- Debian
- Gentoo
- Trac
- Zope / Twisted
- Bittorrent

파이썬의 특징

- 동적 언어로써의 특징
- 멀티플랫폼 언어로써의 특징
- 지배적 스크립트 언어
- 접착제 언어 (Glue language)

(Python is) the most efficient language I've ever used. It's 10 times better than any of the other tools I have used. It's free, it's object-oriented, it adapts to everything, it runs on everything. There is almost an indescribable, 'quality without a name' attraction on my part. - Bruce Eckel, the author of "Thinking in Java"

파이썬의 장점

- 프로토타입을 매우 빨리 만들 수 있음
- 결과적으로 제품을 빨리 출시할 수 있음
- 요구사항 변경에 대한 대처
- 환경 변화에 대응하기 쉬움
- 유지보수가 쉽고 코드를 읽기 쉬움

It did provide a hard measurement on the benefits of using Python instead of C++: the lines of Python code was 10% of the equivalent C++ code. - Greg Stein, Microsoft eShop

인터넷 기업의 업무 적용 분야와 사례

- 자동 빌드 / Unit test / Acceptance test
 - 파이썬 빌드봇
 - 구글의 perforce 연계 빌드/관리 시스템
 - 테스트
 - 브랜치 통합 처리
 - 코딩 스타일 / 가이드라인
- 패키징 시스템
 - 포팅이 간단해짐
 - 설치 스크립트에 유연성을 줄 수 있음
 - 자동화가 간단
- 크롤러와 binary pusher 등의 백그라운드 프로세스
 - 예외사항이 너무 많고 유연한 대처가 필요
 - 똑똑한 프로그램을 만들어도 오래 가지 못함
 - 죽으면 난감하다
- 모니터링과 로깅
 - 매일 매일 바뀌는 요구사항
 - 다양하고 급한 연동 요청
 - QA 부서에서도 직접 코딩이 가능
 - 포팅이 간단해진다
 - 기존에 나온 틀들을 쉽게 쓸 수 있다
- 웹 프론트엔드
 - 진짜 부하가 집중 되는 곳은 몇 군데 안 된다
 - Scalability 보장과 파이썬 사용의 관계
 - Google Groups, Help Center 등..

- Microsoft eShop

We will perhaps eventually be writing only small modules which are identified by name as they are used to build larger ones, so that devices like indentation, rather than delimiters, might become feasible for expressing local structure in the source language. - Donald E. Knuth, "Structured Programming with goto Statements", Computing Surveys, Vol 6 No 4, Dec. 1974

여러 언어의 통합적인 사용

- 파이썬이 정말로 느린가
- 프로젝트에 정말로 얼마나 빨라야 하는가
- 병목은 어디인가
- 여러가지 도구를 적당한 곳에 쓰기
- 집중적으로 사용되는 곳은 전체의 일부에 지나지 않는다
- 잡종강세 (heterosis): 다양한 언어의 장점들을 수용
- 다른 언어를 쓰는 팀들끼리의 다름이 없어짐

We make it easy to use any of the languages, and don't really force people to use a different language. - Greg Stein, Google

파이썬과 다른 언어들의 연계

- SWIG
- RPC
- ctypes, boost::python
- IronPython
- Jython

Beautiful is better than ugly. - Tim Peters

파이썬의 내부 개발 구조와 미래

- Python-Dev
- PEP Process
- trunk / branch 구조
- Python Software Foundation
- Python 3000
- IronPython (CLR & Mono)
- PyPy (cbackend, llvmbackend)