



파이썬을 배워야 하는 이유

2013-04-19

발표자 : 김연수

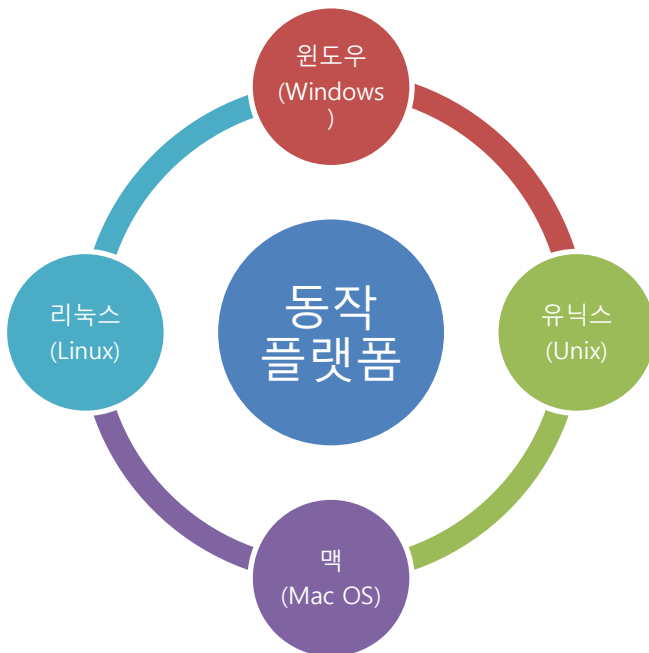
I . 파이썬을 배워야 하는 이유

1. 파이썬이란?
2. 파이썬을 배워야 하는 이유
3. 파이썬 어떤 버전을 배워야 할까?
4. 파이썬 개발 예

파이썬 개요



파이썬(Python)은 1991년 프로그래머인 귀도 반 로섬(Guido van Rossum)이 발표한 고급 프로그래밍 언어로, 플랫폼 독립적이며 인터프리터식, 객체지향적, 동적 타이핑(dynamically typed) 대화형 언어이다. 파이썬이라는 이름은 귀도가 좋아하는 코미디 <Monty Python's Flying Circus>에서 따온 것이다.



인터프리터

- 소스 코드를 바로 실행하는 컴퓨터 프로그램 또는 환경(중간 코드로 변환 후 바로 실행)

객체지향 프로그래밍

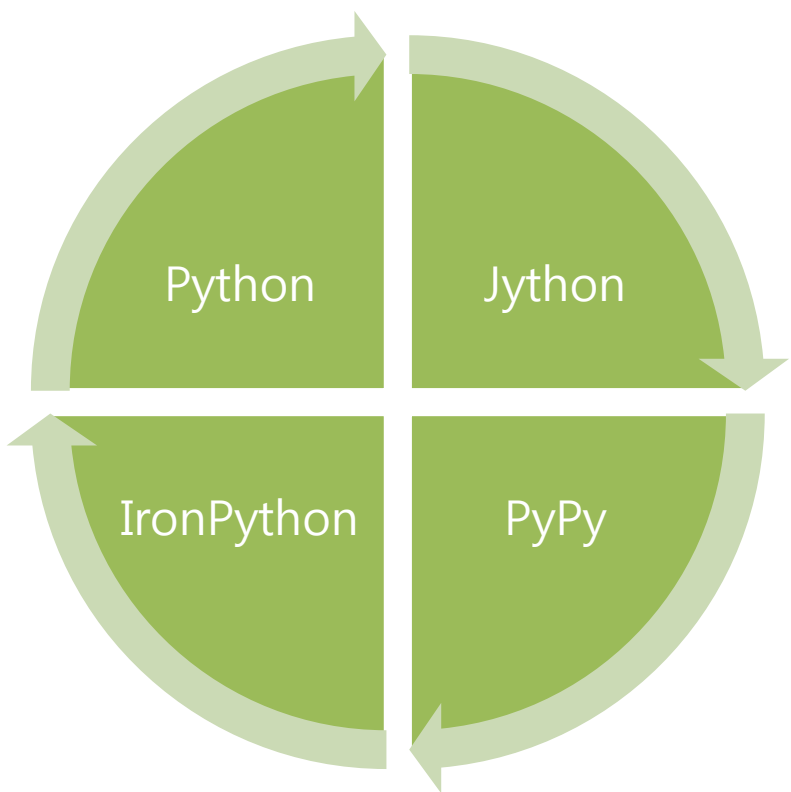
- 프로그래밍을 더 배우기 쉽게 하고 소프트웨어 개발과 보수를 간편하게 하며, 보다 직관적인 코드 분석을 가능하게 하는 장점을 갖고 있다. (C언어의 절차지향적 프로그래밍과 비교)

동적타이핑

- 실행시간에 자료형을 검사한다

파이썬의 종류

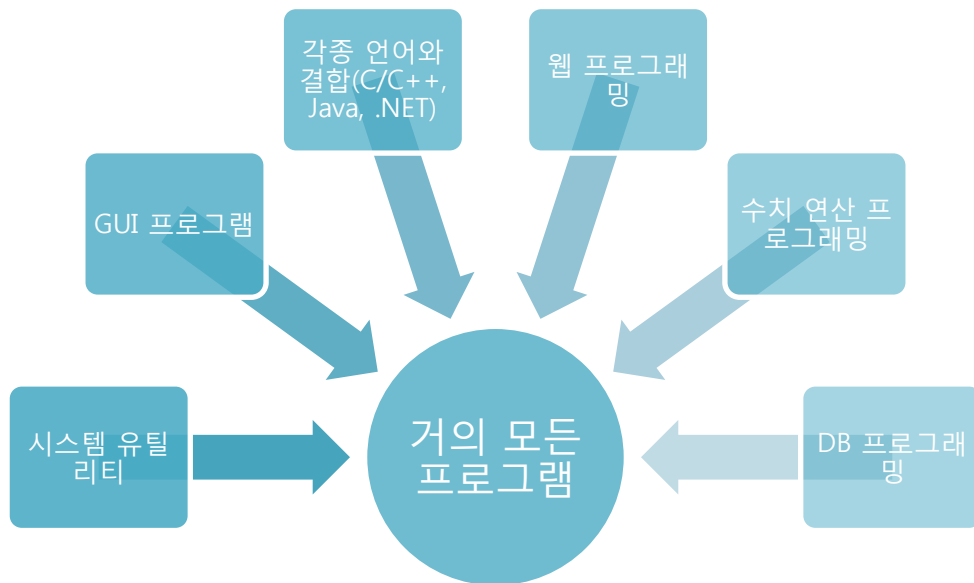
파이썬은 초보자부터 전문가까지 폭넓은 사용자층을 보유하고 있다. 동적 타이핑(dynamic typing) 범용 프로그래밍 언어로, 펄 및 루비와 자주 비교된다. 다양한 플랫폼에서 쓸 수 있고, 라이브러리(모듈)가 풍부하여, 대학을 비롯한 여러 교육 기관, 연구 기관 및 산업계에서 이용이 증가하고 있다. 또 파이썬은 순수한 프로그램 언어로서의 기능 외에도 다른 언어로 쓰인 모듈들을 연결하는 풀언어(glue language)로써 자주 이용된다. 실제 파이썬은 많은 상용 응용 프로그램에서 스크립트 언어로 채용되고 있다. 도움말 문서도 정리가 잘 되어 있으며, 유니코드 문자열을 지원해서 다양한 언어의 문자 처리에도 능하다.



구분	설명
Python	기본 파이썬, C로 구현된 파이썬 인터프리터 C언어 모듈과 파이썬 모듈 연결 가능
Jython	Java로 구현된 파이썬 인터프리터 Java언어 모듈과 파이썬 모듈 연결 가능 UI 프로그램을 구현할때, AWT나 SWING을 사용
IronPython	.NET 파이썬 인터프리터 .NET언어 모듈과 파이썬 모듈 연결 가능
PyPy	파이썬으로 구현된 파이썬 인터프리터

파이썬으로 할 수 있는 일과 없는 일

할 수 있는 일



할 수 없는 일

파이썬으로 도스나 리눅스 같은 운영체제, 엄청난 횟수의 반복과 연산을 필요로 하는 프로그램 또는 데이터 압축 알고리즘 개발 프로그램등을 만들기는 어렵다. 즉, 대단히 빠른 속도를 요구하거나 하드웨어를 직접 건드려야 하는 프로그램에는 어울리지 않는다.

파이썬의 특징점

파이썬의 특징

가독성

풍부한 라이브러리

접착성

무료

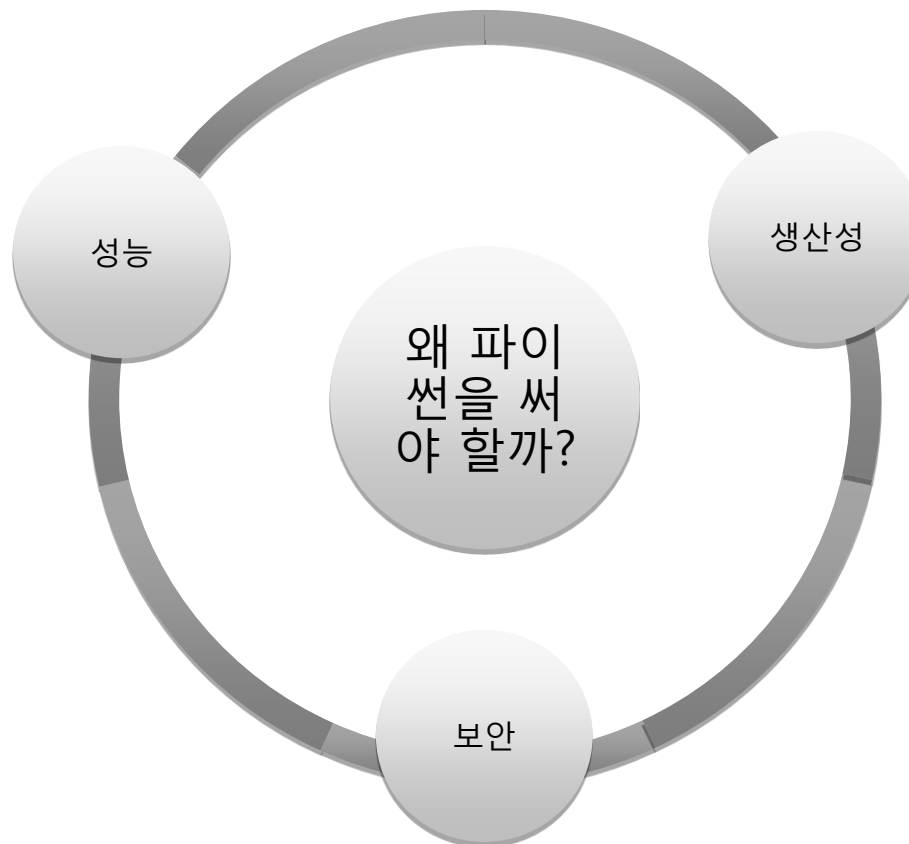
유니코드

동적타이핑

왜 파이썬을 써야 할까?

왜 파이썬을 써야 할까? 아래와 같은 주제로 파이썬을 왜 써야 하는지 알아보자.

내용 출처 : <http://www.itworld.co.kr/news/78349/%EC%9E%90%EB%B0%94+%C2%B7+%EB%8B%B7%EB%84%B7+%EB%8C%80%EC%8B%A0+%ED%8C%8C%EC%9D%B4%EC%8D%AC+%C2%B7%C2%B7+%EB%8B%A4%EC%9D%B4%EB%82%B4%EB%AF%B9+%EC%96%B8%EC%96%B4+%EC%A0%84%EC%84%B1%EC%8B%9C%EB%8C%80>



생산성 : 파이썬 코드는 깔끔하다

- 소프트웨어 업체 액티브 스테이트(ActiveState)의 CTO 제프 홉스는 “파이썬 등 다이내믹 언어를 사용하면 속도를 10배까지 끌어올릴 수 있다고 하는데 어느정도 일리가 있다고 생각한다”고 말했다. 이어 그는 “많은 기업들이 프로토 타입을 개발하는데 다이내믹 언어를 사용한다”
- 오픈소스 분석 업체 레드몽크(RedMonk)의 애널리스트 제임스 거버너는 많은 기업들이 단지 쉽다는 이유로 파이썬을 사용한다고 분석한다. 그는 “다이내믹 언어는 생산성 측면에서 분명 장점이 있다”
- 그러나 자바와 닷넷용 라이브러리가 더 많고 이들 모듈이 모두 파이썬으로 이식되지 않는 것도 사실이다 그래서 필요하다면 자이썬 또는 아이언 파이썬 등을 통해 기존 개발환경에 통합 하는 방법을 사용 할 수도 있다.

성능 : 기업들은 점차 파이썬을 신뢰하고 있다

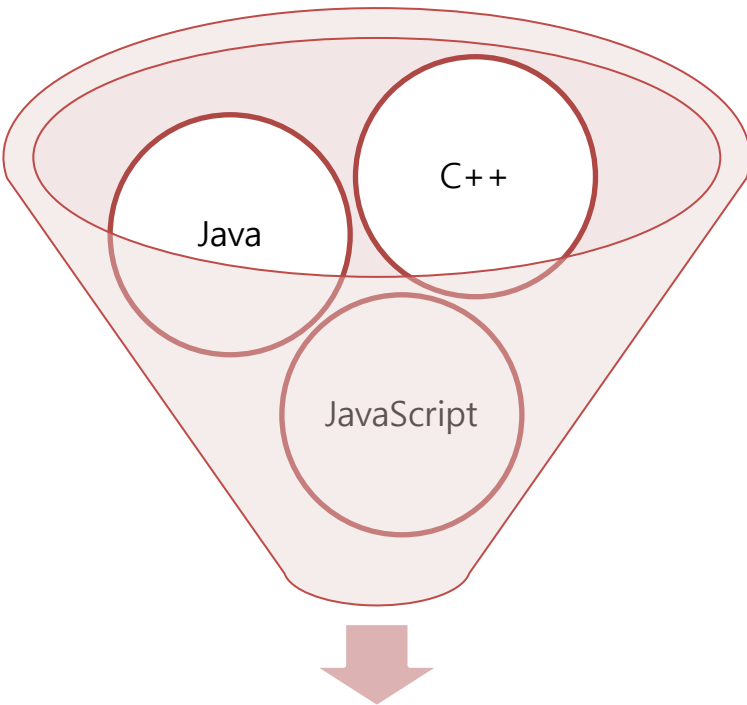
- 거버너는 다이내믹 언어가 인기를 얻고 있지만 이미 기업 시장에서 자리를 잡은 다른 개발 언어 들을 대체하지 못하고 있고 오히려 그 반대 현상이 일어나고 있다고 지적한다. 많은 대형 웹 비즈니스 기업들이 다이내믹 언어에서 자바로 전향하고 있다는 것이다.
- 그는 “자바는 지금 재기를 꿈꾸고 있다”며 “성장하는 웹 기업들이 대부분 자바를 중점적으로 사용하고 있다” 예를 들어 트위터는 본래 루비 온 레일즈로 개발됐지만 확장성 문제로 현재 자바 플랫폼 기반으로 전면 재개발되고 있다.
- 다이내믹 언어는 자바 또는 C++ 처럼 프로세서에 근접해 구동하지 않기 때문에 적절한 수준의 속도를 제공할 수 없다는 지적이 있다. 그러나 현재의 멀티코어 프로세서는 이전보다 훨씬 강력해졌으며 그 결과 다이내믹 언어의 성능은 구글, 유튜브, 메릴 린치, 시스코, VM웨어, 필립스 등이 사용하기에 충분한 것으로 증명됐다.(물론 속도가 중요한 고성능 트레이딩 플랫폼 애플리케이션이라면 파이썬 보다는 C나 자바가 더 적절한 선택이다)

보안

- 지난 수년 동안 자바에서만 수백 개의 취약성이 발견됐고 여기에는 지난 9월 10억명 이상의 자바 SE 5, 6, 7 사용자들이 영향을 받은 치명적인 보안 취약성도 포함돼 있다. 반면에 파이썬에서 발견된 보안 취약성은 소수에 불과하다.
- 브루스 슈나이더 등의 보안 전문가들은 모든 소프트웨어를 단일 개발 언어를 이용해 개발하는 소프트웨어 단일화가 오히려 해커로 인한 피해를 더 키울 수 있다고 지적한다.(한쪽의 취약성이 모두에 영향을 미칠수 있기 때문에)

결론

- 다이내믹 언어와 자바, 닷넷의 장점을 비교하는 것은 너무 이론적인 접근이다.
- 기업에서 사용하는 언어와 툴이 대부분의 경우 CIO가 아니라 개발자들의 의사에 따라 결정된다.
- 그리고 개발자들이 기업환경에서 파이썬을 사용하는 것은 작업을 더 빨리 끝낼 수 있기 때문이다. 파이썬은 이미 기업 내에서 자리를 잡고 있고 개발자들이 이를 선호하면서 닷넷 또는 자바를 대체하기 시작했다.



파이썬과 다른 언어의 비교

비교 언어	설명
Java	일반적으로 Python 프로그램은 Java 프로그램 보다는 느리게 수행된다. 하지만 Python 프로그램은 개발하는 시간이 훨씬 적게 걸린다. Python 프로그램은 Java 프로그램보다도 3-5배 정도 코드가 짧다. 이 차이는 Python의 내장 고수준 데이터 형과 동적인 형결정 기능에서 기인한다고 생각한다. 실행시간 형결정으로 인해서 Python의 실행시간에 Java가 하는 것보다 좀더 많은 일을 한다. 이러한 이유들로, Python은 '접착' 언어로서 아주 적당한 반면, Java는 저수준 구현 언어로 특성화 지을 수 있다. Python이 Java로 구현되기 전에 그 프로토타입을 정하는데 활용된다.
JavaScript	Python의 '객체기반' 부분 집합이 대략 JavaScript와 동일하다. JavaScript와 같이 (그러나 Java와는 다르게), Python은 클래스안에 정의하지 않아도 되는, 단순한 함수와 변수를 사용하는 프로그래밍 스타일을 지원한다. 하지만 JavaScript는 이것이 지원하는 전부이다. Python은, 반면에, 훨씬 큰 프로그램을 클래스와 상속이 중요한 역할을 하는 진정한 객체 지향 프로그래밍 스타일을 통하여 더 좋은 코드 재사용을 하도록 지원한다.
C++	Java에 대해서 이야기 한 대부분이 C++에도 적용된다. Python코드가 Java 코드보다 3-5배 짧으며, C++코드에 비해 5-10배 짧다!! 일 예로 한명의 Python 프로그래머는 C++프로그래머 두 명이 1년에 끝낼 수 없는 일을 두달만에 끝낼 수 있다. Python은 C++로 쓰여진 코드를 사용하는 접착 언어로 빛을 발한다.

파이썬의 인기

Position Apr 2013	Position Apr 2012	Delta in Position	Programming Language	Ratings Apr 2013	Delta Apr 2012	Status
1	1	=	C	17.862%	+0.31%	A
2	2	=	Java	17.681%	+0.65%	A
3	3	=	C++	9.714%	+0.82%	A
4	4	=	Objective-C	9.598%	+1.36%	A
5	5	=	C#	6.150%	-1.20%	A
6	6	=	PHP	5.428%	+0.14%	A
7	7	=	(Visual) Basic	4.699%	-0.26%	A
8	8	=	Python	4.442%	+0.78%	A
9	10	↑	Perl	2.335%	-0.05%	A
10	11	↑	Ruby	1.972%	+0.46%	A
11	9	↓↓	JavaScript	1.509%	-1.37%	A
12	14	↑↑	Visual Basic .NET	1.095%	+0.12%	A
13	15	↑↑	Lisp	0.905%	-0.05%	A
14	16	↑↑	Pascal	0.887%	+0.07%	A
15	13	↓↓	Delphi/Object Pascal	0.840%	-0.53%	A
16	32	↑↑↑↑↑↑↑↑	Bash	0.840%	+0.47%	A
17	18	↑	Transact-SQL	0.723%	-0.04%	A
18	12	↓↓↓↓↓	PL/SQL	0.715%	-0.66%	A
19	24	↑↑↑↑↑	Assembly	0.710%	+0.24%	A--
20	21	↑	Lua	0.650%	+0.08%	B

* 2013년 4월 티오베 순위

왜 파이썬 공식 홈페이지에는 두가지 버전의 다운로드를 제공할까?

The screenshot shows the Python.org download page. The browser address bar displays 'python.org/download/'. The page header includes the Python logo and a 'Download' link. A sidebar on the left contains links to 'ABOUT', 'NEWS', 'DOCUMENTATION', 'DOWNLOAD', 'License', 'Releases', 'Windows', 'Macintosh', 'Other', 'Source', '下载', 'COMMUNITY', 'FOUNDATION', and 'CORE DEVELOPMENT'. The main content area is titled 'Download Python' and states: 'The current production versions are Python 2.7.4 and Python 3.3.1. Start with one of these versions for learning Python or if you want the most stability, they're both considered stable products. If you don't know which version to use, try Python 3.3. Some existing third-party software is not yet compatible with Python 2.7.4 instead.' A red box highlights the download links for both versions. For Python 3.3.1, the links are: 'Python 3.3.1 Windows x86 MSI Installer (Windows binary -- does not include source)', 'Python 3.3.1 Windows X86-64 MSI Installer (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)', 'Python 3.3.1 Mac OS X 64-bit/32-bit x86-64/i386 Installer (for Mac OS X 10.6 and later [2])', 'Python 3.3.1 Mac OS X 32-bit i386/PPC Installer (for Mac OS X 10.5 and later [2])', 'Python 3.3.1 bzipipped source tarball (for Linux, Unix or Mac OS X)', and 'Python 3.3.1 xzipped source tarball (for Linux, Unix or Mac OS X, better compression)'. For Python 2.7.4, the links are: 'Python 2.7.4 Windows Installer (Windows binary -- does not include source)', 'Python 2.7.4 Windows X86-64 Installer (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)', 'Python 2.7.4 Mac OS X 64-bit/32-bit x86-64/i386 Installer (for Mac OS X 10.6 and later [2])', 'Python 2.7.4 Mac OS X 32-bit i386/PPC Installer (for Mac OS X 10.3 and later [2])', 'Python 2.7.4 compressed source tarball (for Linux, Unix or Mac OS X)', and 'Python 2.7.4 bzipipped source tarball (for Linux, Unix or Mac OS X, more compressed)'.

python.org/download/

이 페이지는 영어로 되어 있습니다. 번역하시겠습니까? [번역](#) [번역 안함](#)

python

» Download

ABOUT »
NEWS »
DOCUMENTATION »
DOWNLOAD »
License
Releases
Windows
Macintosh
Other
Source
下载 »
COMMUNITY »
FOUNDATION »
CORE DEVELOPMENT »

Python Wiki
Python Insider Blog
Python 2 or 3?
Help Fund Python

PayPal
DONATE

Python 3.3.1

Download Python

The current production versions are Python 2.7.4 and Python 3.3.1.

Start with one of these versions for learning Python or if you want the most stability, they're both considered stable products.

If you don't know which version to use, try Python 3.3. Some existing third-party software is not yet compatible with Python 2.7.4 instead.

For the MD5 checksums and OpenPGP signatures, look at the detailed Python 3.3.1 page:

- Python 3.3.1 Windows x86 MSI Installer (Windows binary -- does not include source)
- Python 3.3.1 Windows X86-64 MSI Installer (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)
- Python 3.3.1 Mac OS X 64-bit/32-bit x86-64/i386 Installer (for Mac OS X 10.6 and later [2])
- Python 3.3.1 Mac OS X 32-bit i386/PPC Installer (for Mac OS X 10.5 and later [2])
- Python 3.3.1 bzipipped source tarball (for Linux, Unix or Mac OS X)
- Python 3.3.1 xzipped source tarball (for Linux, Unix or Mac OS X, better compression)

For the MD5 checksums and OpenPGP signatures, look at the detailed Python 2.7.4 page:

- Python 2.7.4 Windows Installer (Windows binary -- does not include source)
- Python 2.7.4 Windows X86-64 Installer (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)
- Python 2.7.4 Mac OS X 64-bit/32-bit x86-64/i386 Installer (for Mac OS X 10.6 and later [2])
- Python 2.7.4 Mac OS X 32-bit i386/PPC Installer (for Mac OS X 10.3 and later [2])
- Python 2.7.4 compressed source tarball (for Linux, Unix or Mac OS X)
- Python 2.7.4 bzipipped source tarball (for Linux, Unix or Mac OS X, more compressed)

왜 파이썬 공식 홈페이지에는 두가지 버전의 다운로드를 제공할까?

2008년 12월에 릴리즈된 파이썬3는(당시에는 파이썬 3000이라는 이름으로 발표) 언어 자체의 완성도를 높인다는 명목 하에 하위 호환성을 포기하고 말았다. 많은 것이 바뀌었다. 단순히 문법만이 바뀐 것이 아니었다.

이로 인해서 생겨난 문제는 자명했다. 첫째는 파이썬2.x를 이용해서 짠 코드가 돌아가지 않는다는 것이었다. 물론 파이썬2.x용 코드가 파이썬3.x에서도 돌아갈 수 있도록 자동으로 컨버전을 해주는 툴이 있다고는 하지만 항상 100% 컨버전이 성공한다는 보장도 없을뿐더러 정성들여 짠 코드를 컴퓨터에게 맡긴다는 것은 찜찜한 일이 아닐 수 없다. 두번째 문제는 더 심각한 것이었다. 파이썬이 애초에 인기를 끌게 된건 생산성이 높다는 명성을 얻었기 때문이기도 했지만 파이썬의 장점을 살린 Django 같은 오픈소스 프로젝트나 써드파티 라이브러리들이 생겨났기 때문이기도 했다. 그런데 많은 것이 바뀐 3.x에서는 이것들을 이용할 수가 없었다.

그럼 파이썬3.x의 개선이 하위 호환성 포기로 인해 빚어지는 불편함을 감수할 만큼 필수 불가결한 것이었냐는 의문이 들기 마련인데, 의견이 갈리기는 하지만 그 정도의 댓가를 치를만한 개선은 아니었다는게 대체적인 평가인 것 같다.

아무튼 이런 문제로 3.x는 2.x를 완전히 대체할 수 없었고 결국 2.x와 3.x는 공존하게 되었다.

출처 : <http://hongjiho.tistory.com/14>

자신이 사용하는 써드파티 라이브러리들이 모두 3.X가 지원하지 않는 이상 2.X를 사용하라

아래와 같은 짧은 슬라이드 한장에 들어갈 짧은 소스로, 네이버 웹툰 이미지 저장이 가능하다.

gethtml.py – 네이버 웹툰 페이지 HTML 저장, extractimages.py – HTML 중에서 이미지 URL 경로만 추출하여 해당 URL의 이미지를 파일로 저장

gethtml.py

extractimages.py

```
# -*- coding: cp949 -*-
# 코드페이지를 지정합니다. 파이썬 스크립트 시작에 반드시 삽입해주세요.
# 한글 윈도우면 'cp949', 리눅스면 보통 'utf-8'을 씁니다.
import urllib2, sys
# filename으로 contents를 기록합니다.
def savefile(contents, filename):
    f = open(filename, 'w')
    f.write(contents)
    f.close()
# url의 문서(HTML)를 읽어 리턴합니다.
def gethtml(url):
    response = urllib2.urlopen(url)
    return response.read()
# 메인함수
def main(argv):
    if len(argv) != 3:
        print 'Usage: gethtml.py <url> <savefile>'
        return 1
    url = argv[1]
    savefile = argv[2]
    # url에 접근해서 파일 내용(HTML)을 얻어옴
    html = gethtml(url)
    # 출력해서 확인
    # print html
    # HTML을 파일로 저장
    savefile(html, savefile)
    # 성공적으로 종료
    return 0
# 파이썬은 스크립트 언어이므로 C/C++와 같이 main함수를 먼저 실행하지 않습니다.
# 본 파일을 실행했을 때 main() 함수가 실행되기 위한 처리를 직접 해야 합니다.
if __name__ == '__main__':
    sys.exit(main(sys.argv))
```

```
# -*- coding: cp949 -*-
# 정규식을 사용하기 위한 모듈입니다.
import re, sys
# html에서 만화 이미지 주소를 얻기 위한 함수
def extractimgs(html):
    exp =
re.compile(r'<img.+?src="(http://imgcomicW.naverW.com/webtoon/[0-9]+/[0-9]+/?.+?W.(jpg|png|gif))".*?>')
    imgs = exp.findall(html)
    return imgs
# 메인함수
def main(argv):
    if len(argv) != 2:
        print 'Usage: extractimgs.py <filename>'
        return 1
    f = open(argv[1], 'r')
    html = f.read()
    f.close()
    imgs = extractimgs(html)
    if len(imgs) == 0:
        print >> sys.stderr, "No images!"
        return 1
    for img in imgs:
        print img[0] # full link
        print img[1] # file name
        print img[2] # extension
    return 0
# 파이썬은 스크립트 언어이므로 C/C++와 같이 main함수를 먼저 실행하지 않습니다.
#
# 본 파일을 실행했을 때 main() 함수가 실행되기 위한 처리를 직접 해야 합니다.
if __name__ == '__main__':
    sys.exit(main(sys.argv))
```

출처 : <http://python.kr/viewtopic.php?t=29494>

충격 고로케는 어떻게 만들게 되셨어요?

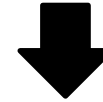
만들게 된 배경이요?^^ 너무 솔직하게 이야기해도 되는지 모르겠지만, 정말 별 생각 없이 만들었어요. 알려지는 게 목적으로 만든 서비스라기 보다는 개인적으로 궁금해서요~. 솔직히 말씀 드리면 코드도 몇 줄 안되고 짧습니다.

앗 몇줄 안 되는 코드! 그럼 만드는데 얼마나 걸리셨어요? 기술적인 이야기도 궁금해요

코드 짜는 데는 정말 1시간 정도 걸린것 같아요. Python으로 기사 집계하는 로직을 간단히 만들고 Django에 짜둔 간단한 모델에 집계 데이터를 넣은 뒤 웹페이지에 키워드 별로 간단히 결과를 보여주는 것이 전부입니다.

집계 로직은요~ 모든 신문사들이 공개적으로 제공하는 실시간 최신 기사제목들 중 문제의 '충격' '경악' '결국' '알고보니' '폭소' '헉' '무슨일이' '살아있네' '이것' '몸매' '미모' '숨막히는' 'ㅇㅇ녀' 등의 키워드가 있으면 잡아내는 것이지요.

개발자가 파이썬으로 한시간만에 만들었다는 "충격 고로케" 사이트 (각종 언론사 기사 중 해당 키워드가 들어간 기사 만을 보여준다)



출처 : <http://readme.skplanet.com/?p=4338>

감사합니다.