

# 파이썬 프로그래밍 언어 소개

이승준 [fb.com/plusjune](https://fb.com/plusjune)

# 같은 이유, 다른 방향성

---

- "Life is Short. Have an Affair" (인생은 짧아요, 바람을 피세요)  
-- 애슐리메디슨 Ashley Madison
- "Life is short, you need Python" (인생은 짧아요, 파이썬을 쓰세요)  
-- 브루스 에켈 Bruce Eckel

# 파이썬 소개

---

- 1989년 암스테르담에서 귀도 반 로섬 Guido van Rossum 개발되기 시작한 언어
- 인터프리터식, 객체지향적, 플랫폼 독립적, 동적인 대화형 스크립트 언어



귀도 반 로섬

- 2013~현재 Dropbox
- 2005~2012 Google
- <http://python.org/~guido>

# 파이썬의 특징

---

- 간편하고, 쉬운 문법
- 객체 지향 프로그래밍, 명령형 프로그래밍, 함수형 프로그래밍
- 방대한 커뮤니티 (필요한 것은 거의 대부분 얻을 수 있다)
- 크로스 플랫폼
- 상호대화적 셸 (인터프리터)
- 임베딩, 확장성
- 오픈소스

# 장점, 단점

---

- 장점: 깔끔하고 간결하다. 적은 양의 코드로 많은 일을 할 수 있다. 즉, 코드 생산성이 높다.
- 단점: C/C++ 같은 강한 타입 언어에 비해 수행속도가 느리다.
- 동일한 내용의 프로그램 C/C++과 비교해 10배 느리다고 하자 (실제로는 생각보다 느리지 않다), 만드는 시간은 C/C++은 10일, 파이썬은 1일 이라면 어떤 선택을 할 것인가?

# 파이썬 FQA

---

- 다른 언어나 환경과 연계 가능성은? (C, JVM, .NET)
- 데이터를 다루는데 적합한가? (어떤 장점이 있을까?)
- R, Excel과 비교?
- 전세계적으로 파이썬을 얼마나 선호(사용)하는가?
- 2.x 와 3.x 두가지 버전 어떤 것을 써야 할까?

# 엑셀

---

- 엑셀은 아주 간편하고 직관적
- 하지만, 대용량 데이터를 다루는데 명확한 한계
- 고수준의 통계와 자동화가 어렵다
- 변수가 매우 적다면 엑셀로도 버텨볼 만 하지만 최소 독립 변수가 대여섯개인 다중회귀분석만 들어가도 답이 없다

# R 과 파이썬 (어떤 것을 선택할 것인가)

---

<https://www.quora.com/Which-is-better-for-data-analysis-R-or-Python>

- 본격적인 데이터 분석 개발이 목적이라면 파이썬을 선택하라
- 머신러닝을 하려면 파이썬을 선택하라

<https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis>

- 웹으로 서비스를 제공하거나 통합하고 싶다면 파이썬을 선택하라

<http://krksap.tistory.com/523> Python을 할 것인가 R을 할 것인가?

- 데이터 탐색(data exploration) 할 때 R보다 파이썬이 낫다

<http://www.kdnuggets.com/2015/05/r-vs-python-data-science.html>

- 데이터 분석외 다른 목적으로도 사용하고 싶다면 파이썬을 선택하라

<http://goo.gl/cA3SZn> Python Displacing R As The Programming Language For Data Science

- 파이썬이 R을 프로그래밍 언어로 대체할 것이다



# 데이터 분석에 파이썬을 사용해야 할 분명한 이유

---

- 파이썬으로 만들어진 소프트웨어들

[https://en.wikipedia.org/wiki/List\\_of\\_Python\\_software](https://en.wikipedia.org/wiki/List_of_Python_software)

# 파이썬 배우기

---

- 문법(언어 자체)보다 소통
- 학습: 學(배우기)과 習(익히기)
- 일부러 예러 만들어 보고 설명해 보기, 고무오리 방법론
- 문제해결에 초점 (필요한 내용을 웹 사이트에서 검색)
- 문법이 아니라 이디엄(idioms)
- 도전적인 과제

# 입문자를 위한 추천 자료

---

- <https://codecademy.com/ko/tracks/python-ko> 코드카데미 파이썬(한국어)
- <https://docs.python.org/3/tutorial> (공식문서) 튜토리얼
- <https://wikidocs.net/1> 점프 투 파이썬 (무료 온라인책)
- <http://goo.gl/1hTohd> 파이썬(열혈강의) Notebook 정리
- <http://pythontutor.com/visualize.html> 파이썬 시각화
- <http://goo.gl/EECP2s> 파이썬을 배우는 최고의 방법

# 파이썬 전체 훑어 보기

---

```
# word counter
import requests
from collections import Counter

r = requests.get('https://www.gnu.org/licenses/gpl.txt')
words = r.content.split()

word_dict = {}
for key in words:
    word_dict[key]=word_dict.get(key,0)+1

count = Counter(word_dict)
for w in count.most_common(10):
    print(w)
```

## 주요 핵심

- comment
- import, from ~ import
- requests
- dictionary
- for in
- print, format

# 파이썬 시각화

- <http://pythontutor.com/visualize.html>

Start shared session Online Python Tutor: Visualize Python, Java, JavaScript, TypeScript, and Ruby code execution  
What are shared sessions?

Python 3.3

```
1 words = "aa bb cc dd cc aa kk".split()
2
3 word_dict = {}
4 for key in words:
5     word_dict[key]=word_dict.get(key,0)+1
6
7 from collections import Counter
8 count = Counter(word_dict)
9 → for w in count.most_common(10):
10 → print(w)
```

Edit code

<< First < Back Step 30 of 30 Forward > Last >>

→ line that has just executed  
→ next line to execute

Program output:

```
('aa', 2)
('cc', 2)
('bb', 1)
('kk', 1)
('dd', 1)
```

Frames

Global frame	
words	
word_dict	
key	"kk"
Counter	
count	
w	

Objects

list

0	1	2	3	4	5	6
"aa"	"bb"	"cc"	"dd"	"cc"	"aa"	"kk"

dict

"bb"	1
"cc"	2
"dd"	1
"kk"	1
"aa"	2

Counter class [extends dict]  
[show attributes](#)

Counter instance

tuple

0	1
"dd"	1