

Python을 활용한 자료구조 & 알고리즘



파이썬(Python)은 1991년 프로그래머인 **귀도 반 로섬(Guido van Rossum)**이 발표한 고급 프로그래밍 언어로, **플랫폼 독립적**이며 **인터프리터식**, **객체지향적**, **동적 타이핑(dynamically typed)** 대화형 언어이다.

명칭 유래 : 코미디〈Monty Python's Flying Circus〉 <http://www.python.org/>

주요 특징

- **동적 타이핑(dynamic typing)**. (실행 시간에 자료형을 검사한다.)
- 객체의 멤버에 **무제한으로 접근**할 수 있다
- 모듈, 클래스, 객체와 같은 언어의 요소가 **내부에서 접근**할 수 있다.
- **리플렉션을 이용한 기술**을 쓸 수 있다.

해석 프로그램의 종류

- **CPython** : C로 작성된 인터프리터.
- **스택리스 파이썬** : C 스택을 사용하지 않는 인터프리터.
- **Jython** : 자바 가상 머신 용 인터프리터. 과거에는 제이파이썬(JPython)이라고 불렸다.
- **IronPython** : .NET 플랫폼 용 인터프리터.
- **PyPy** : 파이썬으로 작성된 파이썬 인터프리터.

정적 타이핑 vs 동적 타이핑

- 정적(Static) 타이핑 - 컴파일 시간에 미리 체크.
- 동적(Dynamic) 타이핑 - 런타임(실행시간)에 체크.

리플렉션이란?

런타임에서 프로그램의 구조를 파악하고 동적 객체 생성 및 함수 호출 등을 수행할 수 있는 기능으로 어떤 클래스가 어디서 파생되었는지, 객체에 어떤 속성이 있는지의 정보를 런타임에서 얻거나 조작할 수 있다.

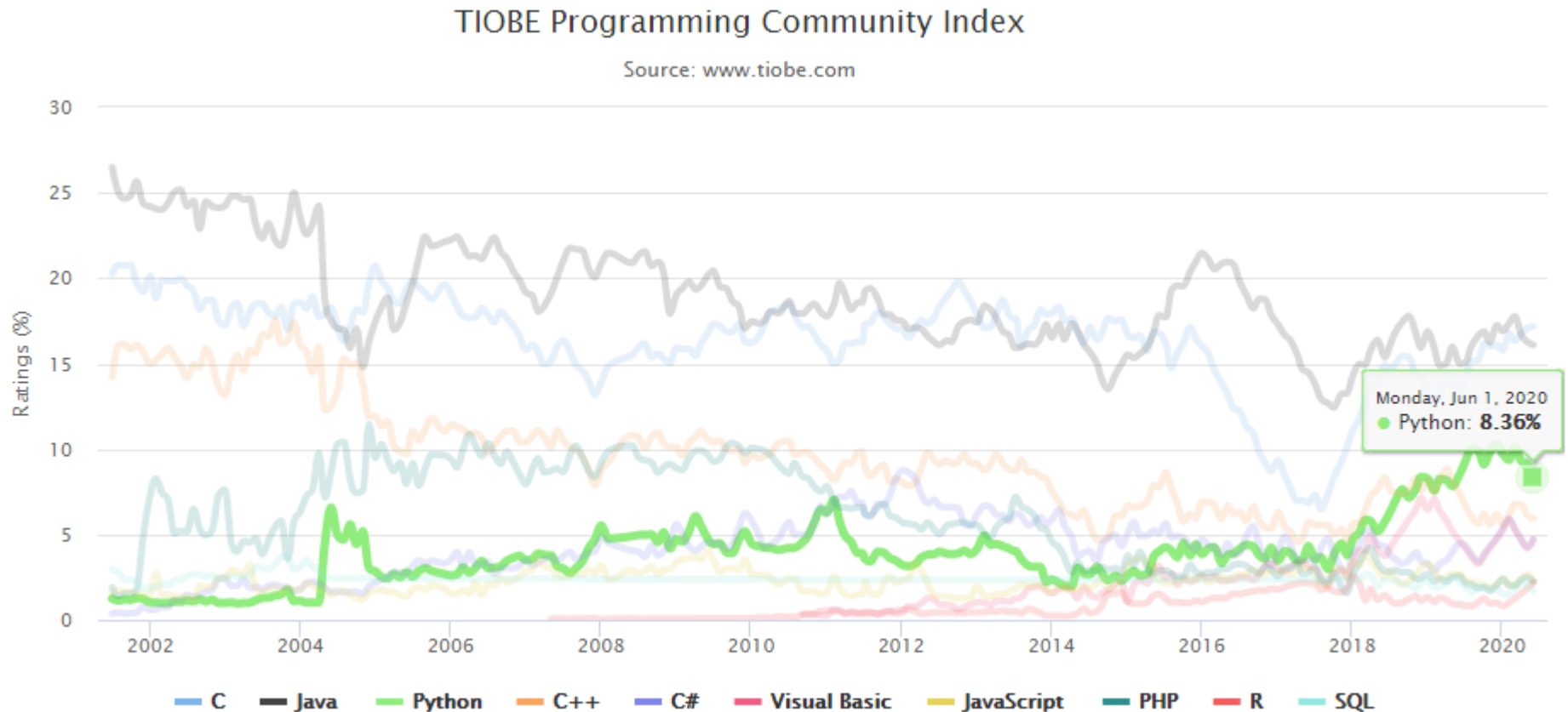
좀 더 쉽게 얘기하면 클래스, 함수, 속성 그 자체에 직접 접근이 가능

```
1 class TestClass:
2     name = 'the value of name'
3     def test_function(self, arg):
4         print('***print word {}'.format(arg))
5
6 # 클래스 및 클래스 변수, 클래스 함수 직접 접근
7 test_class = globals()['TestClass']
8 print(test_class)
9 print(test_class.name)
10 print(test_class.test_function)
11 TestClass.test_function(TestClass, 'test function')
12
13 # 객체 생성
14 test_instance = test_class()
15 print(test_instance)
16
17 # 함수 접근 및 호출, 동적 타이핑
18 method = getattr(test_instance, 'test_function')
19 print(method)
20 method("reflection")
```

프로그래밍 언어 순위(2020.06)

Jun 2020	Jun 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	17.19%	+3.89%
2	1	▼	Java	16.10%	+1.10%
3	3		Python	8.36%	-0.16%
4	4		C++	5.95%	-1.43%
5	6	▲	C#	4.73%	+0.24%
6	5	▼	Visual Basic	4.69%	+0.07%
7	7		JavaScript	2.27%	-0.44%
8	8		PHP	2.26%	-0.30%
9	22	▲▲	R	2.19%	+1.27%
10	9	▼	SQL	1.73%	-0.50%
11	11		Swift	1.46%	+0.04%
12	15	▲	Go	1.02%	-0.24%
13	13		Ruby	0.98%	-0.41%
14	10	▼▼	Assembly language	0.97%	-0.51%
15	18	▲	MATLAB	0.90%	-0.18%
16	16		Perl	0.82%	-0.36%
17	20	▲	PL/SQL	0.74%	-0.19%
18	26	▲▲	Scratch	0.73%	+0.20%
19	19		Classic Visual Basic	0.65%	-0.42%
20	38	▲▲	Rust	0.64%	+0.38%

프로그래밍 언어 순위(2020.06)



파이썬의 장·단점

장점

- 파이썬은 쉽다.
 - 문법이 간결
 - 인간의 언어체계와도 유사
- 무료이지만 강력하다.
 - 꾸준한 업데이트
 - 다른 언어와 쉬운 연동
 - 풍부한 라이브러리 보유
- 빠른 개발속도
 - 공동작업이 수월
 - 유지보수 용이

단점

- 속도가 느리다
 - 인터프리터 언어
- 모바일 컴퓨팅과 브라우저 부재
 - 파이썬으로 개발된 스마트폰 애플리케이션이 없음.
- 설계 제약
 - 동적 입력 형태를 띠기 때문에, 더 많은 테스트가 필요

파이썬으로 무엇을 할 수 있을까?

할 수 있는 일

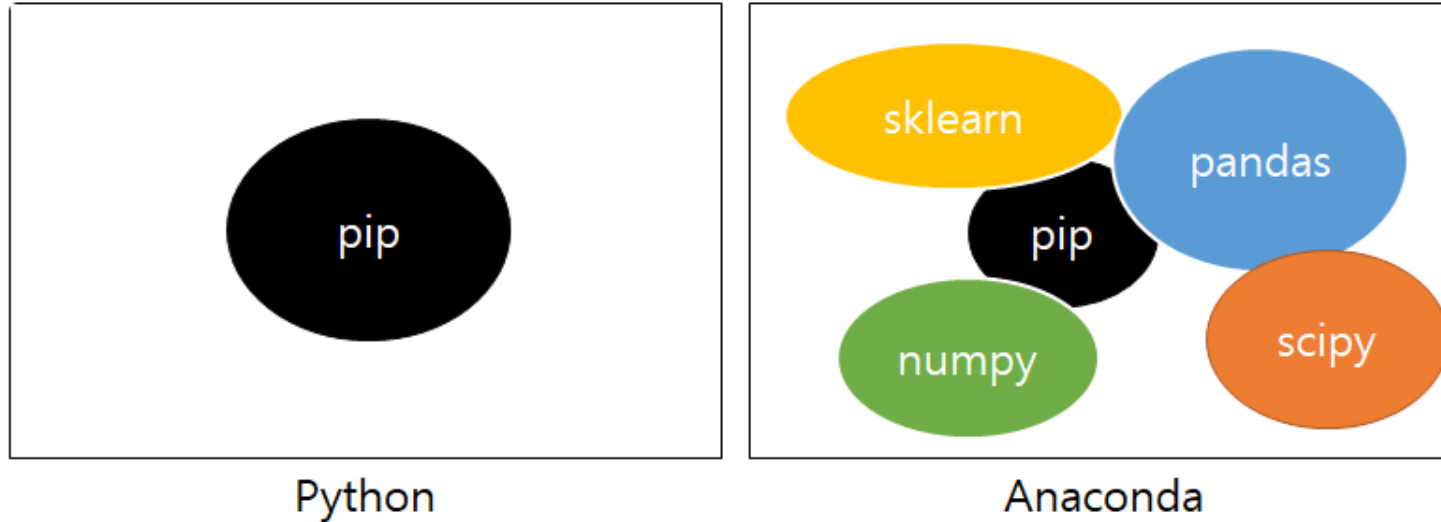
- 시스템 유틸리티 제작
- GUI 프로그래밍
- C/C++와의 결합
- 웹 프로그래밍
- 수치 연산 프로그래밍
- 데이터베이스 프로그래밍
- 사물 인터넷
- 데이터 분석

할 수 없는 일

- 시스템과 밀접한 프로그래밍 언어
- 모바일 프로그래밍

Python vs 아나콘다

Python과 아나콘다(Anaconda)의 차이점은 다음과 같습니다.



Python은 [파이썬 공식 홈페이지](#)에서 받을 수 있으며, `pip` 툴만을 포함하고 있습니다. 필요한 패키지나 라이브러리 등을 설치할 때 모두 수동으로 해줘야 합니다.

아나콘다는 Python 기본 패키지에 각종 수학/과학 라이브러리들을 같이 패키징해서 배포하는 버전입니다. [여기](#)에서 다운로드 할 수 있습니다. 아나콘다에 포함된 툴들로는 대표적으로 `panda`, `numpy`, `scipy`, `sklearn`, `matplotlib`, `Jupyter Notebook` 등이 있습니다.

어떤 것을 선택할 것인가

아나콘다에 포함되어 있는 라이브러리들이 불필요한 경우에는 기본 Python만 설치해도 무관합니다. 하지만, 요즘 유행하는 인공지능이나 빅데이터 관련 개발을 할 경우에는 결국 아나콘다에 포함된 라이브러리들을 설치할 가능성이 높기 때문에 애초에 아나콘다를 설치하는 것이 더 유리합니다. 일일이 라이브러리들을 설치하다보면 의존성 문제 등이 발생할 수도 있기 때문입니다.

참고 Site : <https://snowdeer.github.io/python/2017/11/07/python-vs-anaconda/>

아나콘다 설치



Individual Edition

<https://www.anaconda.com/products/individual>

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download

Anaconda Installers

Windows

Python 3.7
64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (423 MB)

Python 2.7
64-Bit Graphical Installer (413 MB)
32-Bit Graphical Installer (356 MB)

MacOS

Python 3.7
64-Bit Graphical Installer (442 MB)
64-Bit Command Line Installer (430 MB)

Python 2.7
64-Bit Graphical Installer (637 MB)
64-Bit Command Line Installer (409 MB)

Linux

Python 3.7
64-Bit (x86) Installer (522 MB)
64-Bit (Power8 and Power9) Installer (276 MB)

Python 2.7
64-Bit (x86) Installer (477 MB)
64-Bit (Power8 and Power9) Installer (295 MB)

Python Style Guide

스타일이 없는 코드는 일관성이 없는 코드

가독성이 떨어지는 코드로 이어지고 흔히 얘기하는 코드 스멜이 된다

1. PEP8

- 모듈의 이름은 only 소문자
- 클래스는 CamelCase
- 함수의 이름은 snake_case
- 들여쓰기는 스페이스 4칸

2. Google Python Style Guide

- PEP8과 거의 유사하다.
- 구글 프로젝트(Tensorflow) 때문에 가장 요즘 많이 접하게 된다.
- PEP8와 다른 점은 “Docstring”을 작성하는 법에 대해 좀더 깊이 다루고 있다.

3. Numpy/Scipy Style Guide

- PEP8을 베이스로 한다
- 구글과 마찬가지로 Docstring만 거의 다르다

참고 Site : <https://medium.com/@kkweon/파이썬-doc-스타일-가이드에-대한-정리-b6d27cd0a27c>

코드 스멜

- 컴퓨터 프로그래밍 코드에서 더 심오한 문제를 일으킬 가능성이 있는 프로그램 소스 코드

참고 Site : https://ko.wikipedia.org/wiki/코드_스멜

Python 기본 구문

1. 식별자

- 문자 A~Z로 시작하고 밑줄(_), 숫자(0~9) 를 사용
- @, \$, %는 식별자로 사용할 수 없다.
- 대소문자를 구분

2. 예약어

- 총 30개
- 예약어는 상수 또는 변수나 다른 식별자로 사용 불가
- 예약어는 모두 소문자로 구성됨.

Keyword	Keyword	Keyword
and	exec	Not
assert	finally	Or
break	for	pass
class	from	print
continue	Global	raise
def	If	return
del	Import	try
elif	In	while
else	Is	with
except	Lambda	yield

3. 줄 들여쓰기

- 코드 블록을 표시하거나 흐름 제어 할 중괄호가 없다.
- 코드블록은 줄 들여쓰기에 의해 엄격히 적용
- 들여쓰기는 공백(Space), 탭(Tab) 모두 가능
단, 들여쓰기 공백 수는 블록내에 동일 수로 해야 함

4. 멀티 라인 문

- 줄 연속 문자(w)의 사용
- [], {}, () 안에서는 는 줄 연속 문자를 사용할 필요 없음

5. 주석

- 기호 (#)를 사용해 주석을 시작
- 기호 (#) 후 물리적 라인 끝까지 모든 문자를 주석으로 간주

Python 자료형(data type)

1. 수치자료형

int - 1, 1024, 24

float - 3.14, 314e-2 (지수형 표현)

complex - 3 + 4j (실수부와 허수부를 구분)

2. 불자료형

bool - True, False

3. 군집자료형

str - 'Hi Py', "Hi Py", '''Hi Py''',
- 여러 문장을 입력하기 위해서는 ''' ''', """ """

list - 순서 O, 중복 O, 변경 O
- L = [1, 2, 3, 4, 5]
- 순서가 있어서 인덱스, 슬라이싱 가능
- 삽입(append), 삭제(remove), 정렬(sort) 가능
- 더하기(+), 곱하기(*) 가능

tuple - 순서 O, 중복 O, 변경 X
- t = (1, 2, 3, 4, 5)
- 속도 빨라서 파이썬 내부적으로 자주 사용된다.
- 변경이 불가능하여 사용할 수 있는 메소드 수도 적다.

set - 순서 X, 중복 X, 변경 O
- 수학에서 말하는 집합과 유사
- 순서가 없어서 인덱스, 슬라이싱을 사용할 수 없다.
- 값 중복 불가, 변경 가능함

dict - 순서 X, 중복 X, 변경 O
- d = {1:10, 2:20, 3:30, 4:40}
- Key:Value 구성(Key 중복 불가)

Python 제어문

1. Boolean

2. 논리연산자

`and` , `or` , `not`

3. if 문

4. for 문

5. while 문