

3. 파이썬의 이해

1. 파이썬의 개요

파이썬의 탄생 1

- Guido van Rossum 1989년 개발
- 네덜란드 암스테르담 대학에서 컴퓨터 전공
- Monty Python's Flying Circus
- 물론 동물 세계(동물학)에 지대한 관심을 가진 불 스크립팅 언어로 개발
- 불 스크립팅 언어 : 철학적 기술

파이썬의 탄생 2

- 다양한 프로그래밍 패러다임 수용
- 프로그래밍을 용이하게 접근 방식
- "통합형 프로그래밍", "절차적 프로그래밍", "객체지향 프로그래밍, 함수형 프로그래밍 지원
- 다목적 활용
- 응용프로그래밍, 웹, 백엔드 개발, 게임, 시스템 분야 뿐만 아니라 교육적인 목적으로 활용

파이썬의 발전 과정

- 1991년 ABC의 후속 프로그래밍 언어로 시작
- 1999년 DARPA의 "Computer Programming for Everybody" 작
- 2000년 파이썬 2.0 출시
- 기후나이를 통한 개발 체계 시작
- 2008년 파이썬 3.0 출시
- 비 해킹 호환성을 갖는 파이썬 인터프리터

파이썬의 인기

Pythonic

- 독일적, 일본식, 인종적
- 신속성, 직관적
- 복합적
- 특정 회사에 종속X, OS(종류) 독립적
- 오픈소스
- 간단적
- 친숙성
- 확장적

오픈소스

- 파이썬 관련 개발을 위한 제안(PEP)
- 많은 개발자의 의견을 수용하고 토론하여 발전한 언어
- 새로운 파이썬의 기능, 파이썬 프로젝트, 환경에 대해 커뮤니티에 공개 문서나 정보를 제공
- 파이썬 관련 다양한 기술적 사항과 기능을 위한 근거를 제공
- 커뮤니티의 의견을 수렴, 합의 도출, 반대파도 수용
- PEP 8(스타일 가이드) 대표적

간단적 & 직관적

- 실행할 수 있는 의사 코드(Executable pseudocode) 수준의 문법
- if 3 in [1,3,5,7]:
print("3이 들어있습니다")
- 리스트 [1, 3, 5, 7]에 3이 포함되어 있으면 "3이 들어있습니다"를 출력합니다.

명세형 & 친숙함

```
# C
int sum = 0;
for(i=0; i<n; i++) {
    sum += i;
}
printf("합은 %d", sum);
```

C

```
# Python
sum = 0
for i in range(n):
    sum += i
print("합은 {}".format(sum))
```

Python

대형 개발자 커뮤니티

라이브러리 및 프레임워크

파이썬의 안정

- C나 제어 용으로 작성된 프로그래밍 언어인 속도
- 유명한 개발 커뮤니티로 인해 개발이 용이함
- 불 스크립팅 언어 용으로 개발
- 모바일 및 등 응용 애플리케이션 개발 용이함
- Rust 또는 Go 고려

2. 파이썬 프로그램의 실행

파이썬 실행 환경

- 1) 운영체제 독립적이며 2) 인터프리터 3) 객체지향적, 4) 동적 타입(dynamically typed) 5) 다목적 언어
- 1. 플랫폼 독립적
- 윈도우, 리눅스, 유닉스, macOS 등 다양한 운영체제(환경)에서 별도의 컴파일 없이 실행 가능
- 2. 인터프리터적
- CPython, PyPy, Cython, Jython 등 다양한 인터프리터 환경 사용 가능
- 3. 객체지향적
- 프로그래밍 객체로 모델링
- 4. 동적 타이핑
- 변수의 자료형을 지정하지 않음
- 5. 대용량 언어
- 작업은 코드에 대한 수동 결과를 바로 확인하고 디버깅해서 코드 작성 가능

CPython

- C 언어로 개발된 파이썬 인터프리터
- C 기반 라이브러리의 지원을 통한 확장성 극대화
- 컴파일러의 도움
- 1. 실행 코드 실행할 때 : 부트스트랩을 필요로 하며 실행을 통해 자신의 언어로 작성된 컴파일러
- 2. 소스 코드 실행할 때 : C 언어로 작성된 컴파일러
- 오픈소스로 커뮤니티의 기여로 자체적 발전

파이썬 프로그램 실행과정

파이썬 애플리케이션은 소스 코드 형태로 제공

- 1 CPython의 인터프리터 후 바이트코드(.pyc) 파일 생성
- 2 파이썬 가상머신은 바이트코드를 실행 중인 VM 실행
- 3 변형없이 재실행 시 바이트코드로 빠르게 실행

3. 파이썬 프로그래밍 환경

IDLE

- 기본으로 포함된 파이썬의 통합 개발 환경
- 파이썬의 Tkinter GUI 용으로 개발
- 구문 강조, 자동 완성, 스코프 추적 등 용이 포함된 인터프리터 IDE 지원
- stopping, breakpoint, call stack을 확인할 수 있는 통합 디버깅 환경 제공
- 파이썬 공식 홈페이지에서 다운로드 가능
- <https://www.python.org/>

주제어 노트북

- 오픈소스 기반의 웹 애플리케이션
- 파이썬을 비롯한 40여개의 프로그래밍 언어 지원
- 협동적인 소스코드-문서-데이터 환경에서 벗어나 할 작업 "데이터" 개발 및 실행 환경
- 문서화하여 다른 사람과 공유하기가 편리
- 마크다운(Markdown)을 사용하여 코드, 문장, 이미지, 그래프 등 다양 가능

구글 Colab

- 2017년 과학 연구와 교육을 목적으로 개발
- 클라우드 기반 주제어 노트북 개발 환경
- 주제어 노트북 + 구글 드라이브를 결합한 서비스
- 데이터 분석 및 다양한 연산 중 고성능 컴퓨팅 리소스 활용 가능

4. 실행 - Colab

Google Colab

<https://colab.research.google.com/>

Colab 사용 방법

- 로그인
- Google Drive
- Chapter3test.ipynb
- 3 + 5
- 성공 버튼을 누르면 GCP로부터 자원 할당
- 이동 버튼
- 파이썬 3.6.0을 다운로드
- 데이터 분석 및 다양한 연산 중 고성능 컴퓨팅 리소스 활용 가능
- 리소스 확인 가능
- [2] a = 10
print(a)
- a = 10
print(a)
- 엑스포트 버튼 가능
4. 실행(Run)
- 행 번호 표시
5. 박스
- 구글 드라이브의 리소스