대표적인 프로그래밍 언어

* C, C++, Java → 그러나 복잡 & 어려움
* Python의 장점?
  + 빠른 학습속도
  + 명령의 수행 결과를 빠르게 확인 가능
  + 확장 기능 지원
  + 다양한 플랫폼 (윈도우즈, 맥, 리눅스 등) 사용 가능
* 발전 가능성
  + 성능에 영향을 미치는 부분은 C언어 등으로 모듈화하여 단점 극복
* 빅데이터 분석, 머신러닝, 딥러닝, IoT 분야에서 가파른 활용률 증가

Python에 대해

* 독립적
* 인터프리터 방식
* 객체지향
* 동적인 & 대화형 성격을 가진 프로그래밍 언어

Python의 종류

* Cpython: C로 작성된 파이썬 (표준 파이썬)
* IronPython: .Net과 Mono용으로, C#으로 구현되어 있음
* Jython: 자바로 구현된 파이썬; 자바가상머신에서 동작 (자바 클래스, 자바 표준 라이브러리 사용 가능)
* PyPy: 파이썬으로 구현되어 Cpython보다 빠르게 구현 가능함

Python 개발 철학

* 아름다운 것이 추한 것보다 낫다
* 명시적인 것이 묵시적인 것보다 낫다 explicit is better than implicit
* 단순한 것이 복잡한 것보다 낫다
* 복잡한 것이 난해한 것보다 낫다 complex is better than complicated
* 가독성이 중요하다

특징

* 인터프리팅 방식: 명령의 실행결과를 대화형으로 바로 확인 가능하여 비전문가들에게 프로그래밍의 허들을 낮춰줌
* 동적 타이핑 (dynamic typing): 실행 시간 값에 의해 자료형 결정
* Garbage collector: 생성된 객체에 대한 메모리 관리는 garbage collector 이용
* 가독성: e.g.) 코드 블록의 들여쓰기를 강제하여 가독성 상승
* 풍부한 라이브러리: 다양한 분야를 쉽게 추가 및 확장 가능; 표준 라이브러리와 통합환경이 배포판과 함께 제공 e.g.) 정규 표현식, 운영 체제의 시스템 호출, XML 처리, 직렬화, 각종 통신 프로토콜, 전자 메일이나 CSV 파일 처리
* 유니코드 문자열
* 오픈소스: 사실상 무료로 파이썬 소프트웨어 재단에서 관리
* 학습 용이성: 프로그램의 문서화를 중시해 언어의 기본 기능에 포함; 도움말 문서와 API도 체계적으로 정리
* 내장 스크립트 언어: 다른 언어로 쓰인 모듈을 연결하려는 목적으로 자주 이용

파이썬의 인기 분야

* 광고
* 데이터 수집 (웹 크롤링)
* 데이터 과학 및 인공지능
  + Numpy
  + Pandas: 데이터 분석 시 사용하는 표준 라이브러리
  + SciPy: 수학 과학 엔지니어링 분야에서 활용
  + Scikit-learn: 데이터 마이닝과 데이터 분석을 위한 도구; 머신러닝 지원