

IBK 직업훈련 과정

파이썬 기초

이일주/백재원 강사

강사 소개

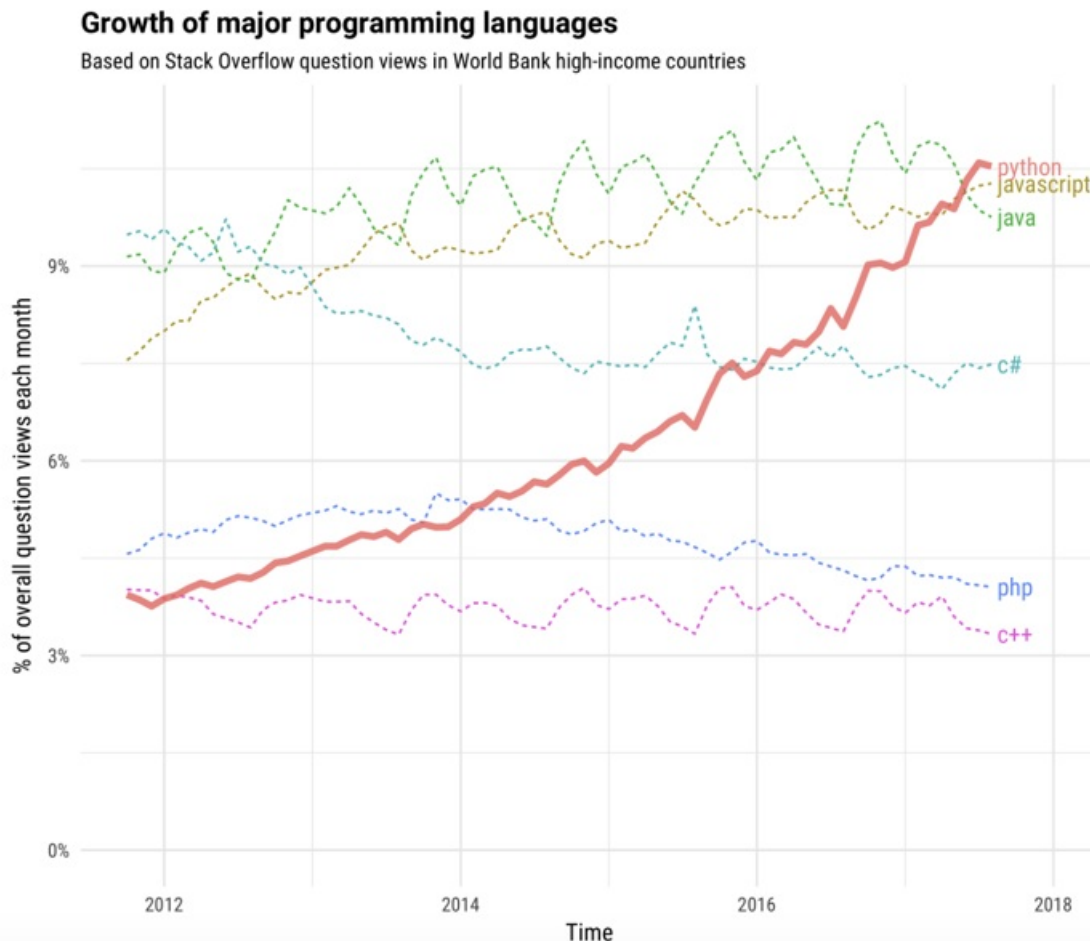
이일주 / 백재원

- 서울대학교 전기정보공학부 연구원
- 서울대학교 경영학부 전임강사

왜 파이썬인가?

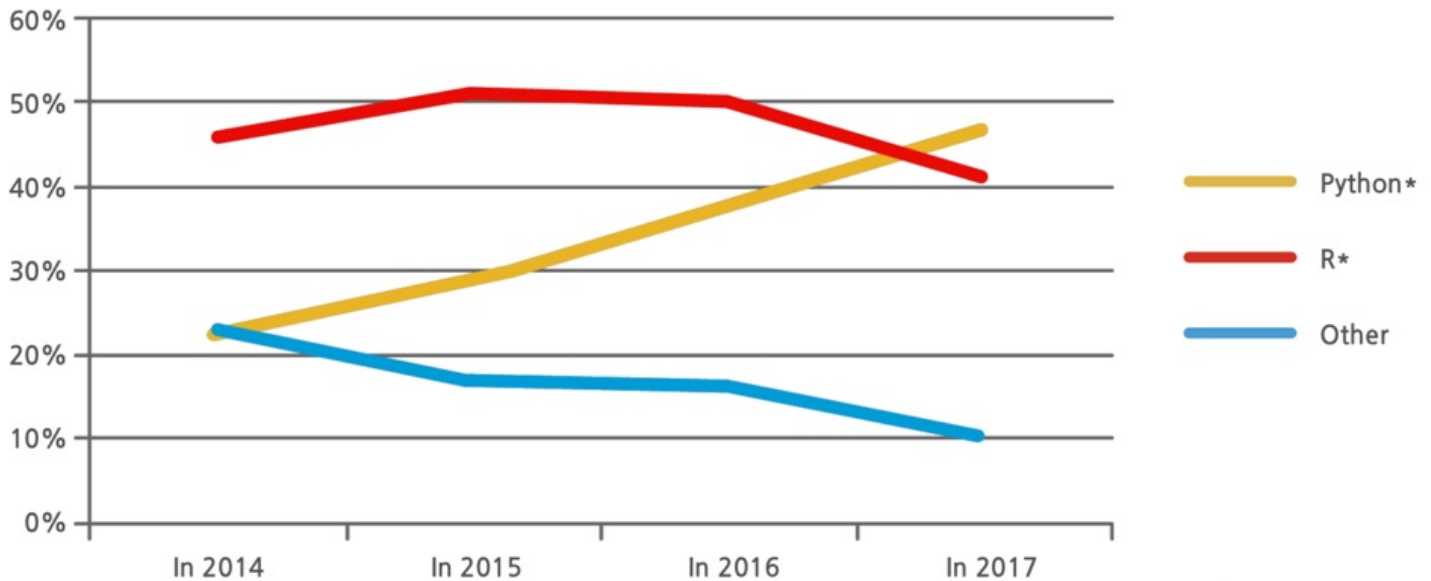


프로그래밍 언어 추이



Trends from 2014 to 2017

Python vs R vs Other for Analytics & Data Science, 2014-17



[출처: kdnuggets.com]

Easy to learn

- 광범위하게 사용하고 있고 초보자들도 쉽게 배울 수 있는 프로그래밍 언어
- 다른 프로그래밍 언어보다 문법이 간단하면서도 엄격하지 않음(e.g, 변수를 사용하기 전에 별도의 자료형 선언을 하지 않아도 됨)
- 문법 표현이 매우 간결하기 때문에 프로그래밍 경험이 없어도 아주 짧은 기간 내 파이썬 문법을 마스터할 수 있음(e.g., 명령문을 구분할 때 중괄호('{', '}') 대신 <들여쓰기(indentation)>를 사용)
 - 사람이 글을 쓸 때 문단을 구분하는 것처럼 들여쓰기를 사용해서 명령문 블록을 구분하기 때문에 글을 써내려가듯이 프로그램 논리를 자연스럽게 표현할 수 있음

크로스 플랫폼 언어(cross-platform language)

- 윈도우즈(Windows), 맥 OS(macOS), 리눅스(Linux), 유닉스(Unix) 등 다양한 운영체제에서 실행 가능

다양한 고급 프로그래밍 기능 제공

- 숫자(정수, 실수, 복소수 등) 또는 문자열(ASCII, Unicode) 등 다양한 내장(built-in) 기본 자료형과 리스트, 튜플, 사전과 같은 내장 복합 자료형을 지원
- 클래스, 다중 상속(multiple inheritance), 모듈, 패키지, 예외처리(exception handling), 자동 메모리 관리 등을 지원
- 생성자(generator), 리스트/세트/딕셔너리 축약(list/set/dictionary comprehension) 등 강력한 고급 프로그래밍 기능을 지원

강력하고 풍부한 라이브러리(library) 제공 : 개발 속도가 빠름

- 프로그래밍에 자주 사용되는 다양한 기능이 장착된 표준 라이브러리(standard library)를 기본적으로 제공
 - 라이브러리(library) : 다양한 기능을 제공하는 함수나 모듈을 프로그램에서 쉽게 불러 사용할 수 있게 미리 작성해서 컴파일해 놓은 코드(모듈)들의 모음
 - e.g., 웹 서버 연결, 정규표현식(regular expression)을 통한 문자 검색, 파일 읽고 쓰기 등
- 대규모의 사용자 커뮤니티가 존재
- 도움을 손쉽게 받을 수 있을뿐 아니라 참고자료도 풍부
 - e.g., 스택오버플로우(<https://stackoverflow.com/>)에서 파이썬 관련 많은 도움을 받을 수 있음

무료(free) 소프트웨어

- 무료로 다운로드 및 사용이 가능
- 개인이 제작한 애플리케이션에 탑재시킬 때도 무료
- 파이썬은 오픈소스 소프트웨어이기 때문에 자유(free)롭게 수정 및 재배포가 가능

프로그램과 프로그램 개발 절차

프로그램(program)이란?

특정한 작업을 어떻게 수행해야 하는지 그 순서를 일련의 명령어로 나열한 것이다.

프로그램 언어의 공통 기본기능

파이썬(Python), 자바(Java), C 언어를 포함한 모든 프로그래밍 언어는 공통적으로 다음과 같은 기본 기능들을 제공한다.

입력

- 키보드나 파일 또는 별도의 장치로부터 데이터를 입력한다.

출력

- 데이터를 컴퓨터 화면에 보여주거나 파일 또는 별도의 장치로 전송한다.

처리

- **연산 처리** : 더하기, 빼기, 곱하기, 나누기 같은 기본적인 수학적 연산을 수행한다.
- **순차 처리** : 순서대로 작업을 수행한다.
- **선택(조건) 처리** : 특정한 조건에 따라 그에 맞는 적절한 코드를 선택해서 실행한다.
- **반복 처리** : 주어진 횟수나 조건에 따라 특정한 작업을 (주로 약간의 변화를 주면서) 반복적으로 수행한다.

프로그램 개발 과정

프로그램 개발은 다음과 같은 과정을 거친다.

1. 프로그램 논리 설계 및 개발
2. 프로그램 코드화(coding)
3. 기계어(machine language)로 변환(컴파일)
4. 프로그램 실행과 검증

Step 1 : 프로그램 논리 설계 및 개발

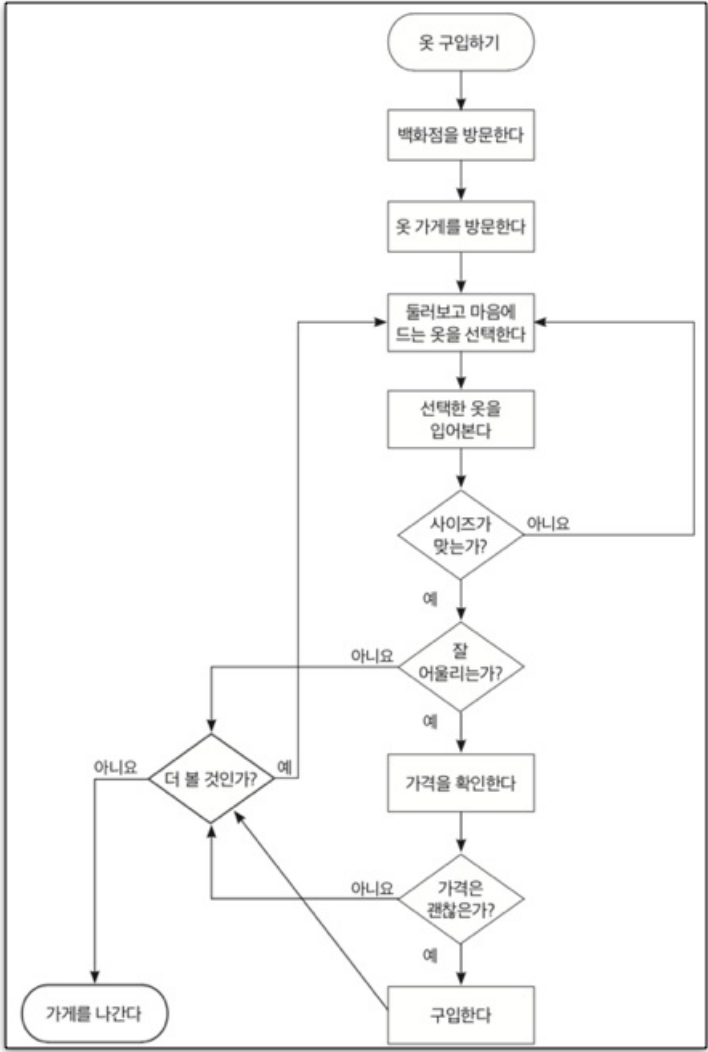
핵심 단계

- 프로그램 개발 단계 중 가장 중요한 핵심 단계다.
- 알고리즘(algorithm) 개발이라고도 부른다.
- 어떤 작업 절차를 어떤 순서로 실행할지 결정하는 단계다.

논리 설계에 사용되는 툴(tool)

- 흐름도(flowcharts) / 순서도
- 의사(擬似)코드(슈도코드, pseudocode)
- 일상어(자연어)

흐름도(flowcharts) 예시



의사(擬似)코드(슈도코드, pseudocode)

- 실제 프로그래밍 언어가 아닌 유사한 형태의 언어나 일상어 형태로 작성한 코드
- 프로그램 설계 시에만 사용하므로 문법(구문) 오류를 신경 쓸 필요가 없음
 - 컴파일 또는 실행에 사용하는 코드가 아님(가짜 코드)
- 의사코드는 어떤 프로그램 언어든지 실제 코드로 옮겨 적는 것이 가능

- e.g., 녹차 만들기
 - 주전자에 물을 넣는다;
 - 주전자의 물을 끓인다;
 - 주전자의 물이 끓으면 불을 끈다;
 - 컵에 티백을 넣는다;
 - 컵에 물을 붓는다;
 - 약간 기다린 뒤 티백을 꺼낸다;
 - 마신다;

Step 2 : 프로그램 코드화

프로그래밍(programming)

- 파이썬(Python), 자바(Java), R, 루비(Ruby), C, C++, 비주얼 베이직(Visual Basic), SQL 등의 해당 - 프로그래밍 언어 문법에 맞추어 코드를 작성
- 인간 언어처럼 프로그래밍 언어도 각기 다른 표현과 문법을 가지고 있음
- 일반적으로 프로그램 논리개발보다 코드화 작업이 더 쉬움

코드화의 기본 작업

선언(declaration) : 프로그램에서 사용될 자료형(data type)을 컴퓨터에게 알려주는 단계

- 데이터는 사용되기 전에 반드시 미리 선언
 - 일반적으로 코드화 작업 앞부분에서 선언
- 자료형을 선언함으로써 메모리에 차지할 저장 공간과 쓰임새를 컴퓨터에게 미리 알려줌

입력(input) : 컴퓨터에 데이터를 입력하는 것

- 사람으로부터 : 키보드, 마우스, 펜 등
- 파일로부터 : 텍스트 파일, 데이터베이스, 웹 페이지 등
- 센서로부터 : 빛, 동작 감지, 생체인식 등

처리(processing) : 프로그램에 의해 수행되는 작업

- e.g., 급여 계산, 체스 게임에서 말의 이동

출력(output) : 결과 또는 답

- 화면에 표시
- 파일로 저장
- 종이에 출력

입력받은 값에 2를 곱하여 출력하는 프로그램 예시

numeric	mynumber	← 선언
input	mynumber	← 입력
myanswer =	mynumber * 2	← 처리
output	myanswer	← 출력

Step 3 : 기계어(machine language)로 변환(컴파일)

컴퓨터는 1 또는 0 밖에 모른다.

- 소프트웨어를 통해 프로그램을 컴퓨터가 이해할 수 있는 기계어(0과 1)로 변환
- 다양한 프로그래밍 언어가 있지만 컴퓨터는 오로지 1 또는 0만 이해할 수 있기 때문에 어떤 언어든지 기계어로 변환이 필요

문법 오류(syntax error)

- 컴파일러가 코드를 변환할 수 없을 때 문법 오류가 발생
- 원인
 - 오타
 - 잘못된 문법
- 코드에서 잘못된 부분을 수정한 뒤 다시 변환(recompile)해야 함

Step 4 : 프로그램 실행과 검증

프로그램 오류의 종류와 디버깅(debugging)

- **문법 오류(syntax error)** : 오류 수정이 쉬움(즉, 버그 잡기가 쉬움)
- **런타임 오류(runtime error)** : 예외(exception) 처리를 해야 함
- **논리 오류(logical error)** : 오류 없이 프로그램이 실행되지만 예상한 결과가 나오지 않는 경우 e.g., 1 / 2 * 3
- 문법 오류는 컴파일 단계에서, 런타임 오류와 논리 오류는 실행 단계에서 발생