대학 수학 학습을 위한 파이썬 부트캠프

I: Python 프로그래밍의 이해

소개

서울대학교 데이터 사이언스 대학원(Graduate School of Data Science)

- 데이터 사이언스(Data Science)
  - 문제 영역의 지식을 바탕으로 다양한 데이터를 획득, 정제, 모델링, 분석 등 일련의 과정을 통해 의사결정에 유용한 산출물을 얻어내는 학문
  - 전문 지식(Domain Knowledge) + 컴퓨터 공학(Computer Science)

- Contact address

yskim@kdb.snu.ac.kr

jhlee@kdb.snu.ac.kr

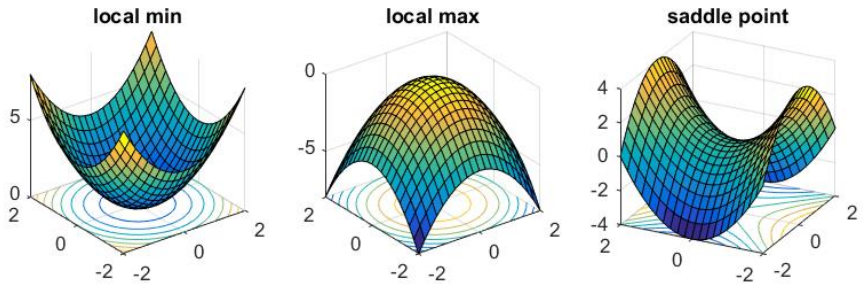
부트캠프 구성

Python 프로그래밍의 이해 (변수와 연산)		
9/25(토) 13:00 ~ 15:00	주제	· Python 소개 · 변수 및 기본 연산자
	실습 예제	· Python 개발 환경 세팅(Anaconda 설치, Jupyter notebook 활용) · Python 코딩 기초 실습
Python의 문법과 자료형 (벡터와 행렬 및 수치 미분)		
10/2(토) 13:00 ~ 15:00	주제	· 기초 문법 (Syntax, comment, indentation) · 기본자료형 · 복합자료형 (list, vector, matrix) · 함수의 정의
	실습 예제	· Python 자료형 활용 예제 – 미적분학 1: 행렬의 연산 · 함수 활용 예제 – 수치 미분
Python 제어문 활용 (테일러 급수와 근 찾기 알고리즘)		
10/9(토) 13:00 ~ 15:00	주제	· 조건문, 반복문 · 재귀함수
	실습 예제	· 반복문 실습 예제 – 미적분학 1: 테일러 급수 연산 · 재귀함수 실습 예제 – 미적분학 1: Newton's method
Python 라이브러리 활용 (다변수함수 시각화)		
10/16(토) 13:00 ~ 15:00	주제	· 라이브러리 활용(Matplotlib, numpy, math, sympy 등) · Python 응용
	실습 예제	· 미적분학 2: 다변수 함수 3D 시각화 (그래프, 등위면, 극점, 안장점 등)

※ Zoom 온라인 미팅으로 실시간 진행됩니다.

※ 주차별 강의는 Co-티칭으로 진행되며, 불참시 녹화 영상이 제공되지 않습니다.

프로그래밍 활용



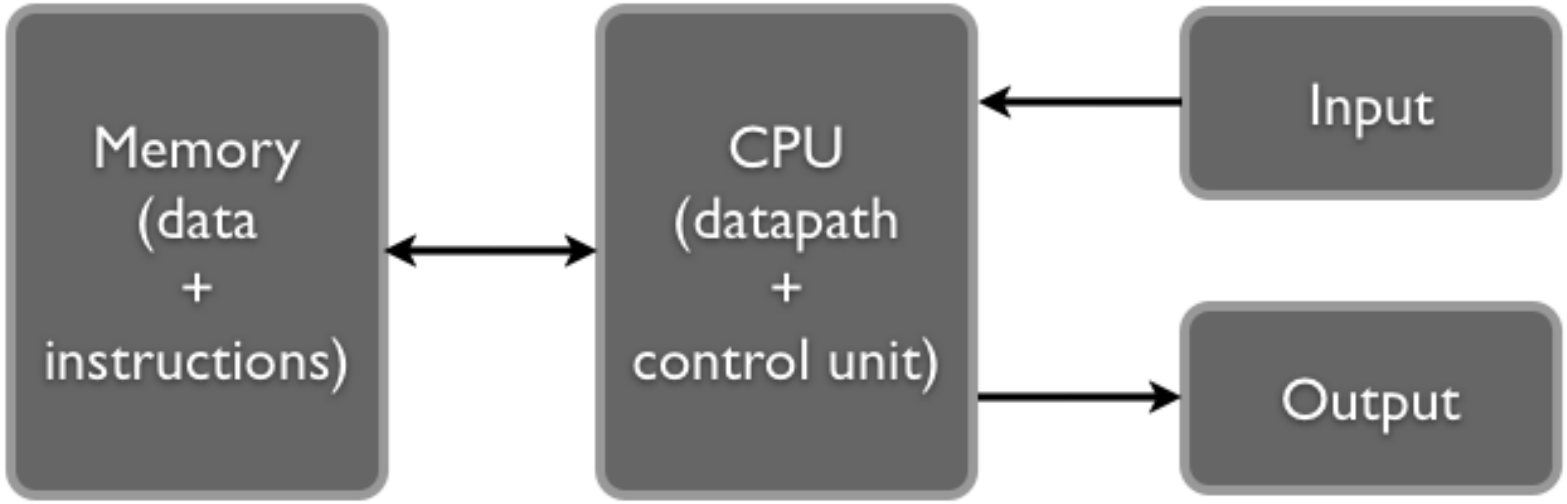
AlphaGo

컴퓨터

1. 방대한 양의 정보를 저장하고 처리할 수 있는 전자 기기(electronic device)
2. 산술 연산이나 논리 연산을 자동으로 수행하도록 프로그래밍 할 수 있는 범용 장치(programmable machine)

폰 노이만 구조

- The Von Neumann architecture (1945)
  - 중앙 처리 장치(Central Processing Unit)
  - 메모리: 데이터 + 프로그램
  - 입출력 장치



컴퓨터 프로그램

- 프로그램(program)이란?
  - 컴퓨터가 특정 작업을 어떻게 수행해야 하는지 알려주는 명령어의 배열
  - 프로그램의 명령어는 기계어의 형태로 변환된 후 실행됨

- 기계어(machine language)
  - 컴퓨터의 CPU가 직접 해독하고 실행할 수 있는 컴퓨터 언어
  - 이진법의 문자열

컴퓨터 프로그램 예시

- 3과목(수학,영어,과학)의 점수를 받아 최대값을 출력하는 프로그램
  1. 과목별 점수를 데이터로 입력한다.
  2. 수학 점수가 영어, 과학보다 크다면,
  3. 수학 점수를 최댓값으로 정한다.
  4. 영어 점수가 수학, 과학보다 크다면,
  5. 영어 점수를 최댓값으로 정한다.
  6. 과학 점수가 수학, 영어보다 크다면,
  7. 과학 점수를 최댓값으로 정한다.
  8. 최댓값을 출력한다

수학	영어	과학
90	80	75

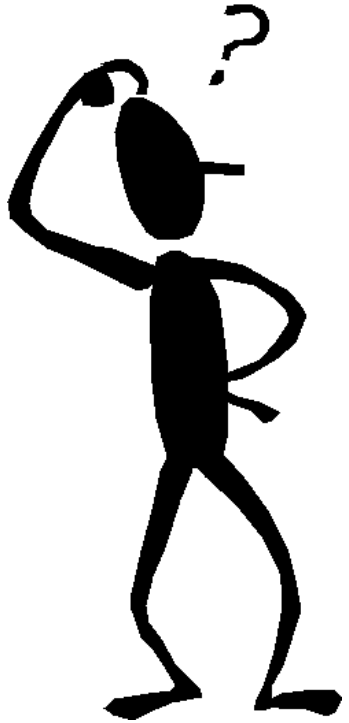
입력

90

출력

프로그래밍 언어

- 컴퓨터는 오직 기계어만을 이해할 수 있다
- (사람이 배우기에는 어려움)



프로그래밍 언어

- 프로그래밍 언어는 사람이 이해할 수 있는 high-level 인터페이스
- 코드(code): 프로그램을 사람이 읽을 수 있는 프로그래밍 언어로 기술한 텍스트 파일

```python
# The green lines are comments
if __name__ == "__main__":
    # The program starts from here
    math = 90
    english = 80
    science = 75

    maximum_score = 0
    if math >= english and math >= science:
        maximun_score = math
    elif english >= math and english >= science:
        maximun_score = english
    else:
        maximun_score = science

    print("최고점은 %d점 입니다." % maximum_score)
```

```
최고점은 90점 입니다.
```

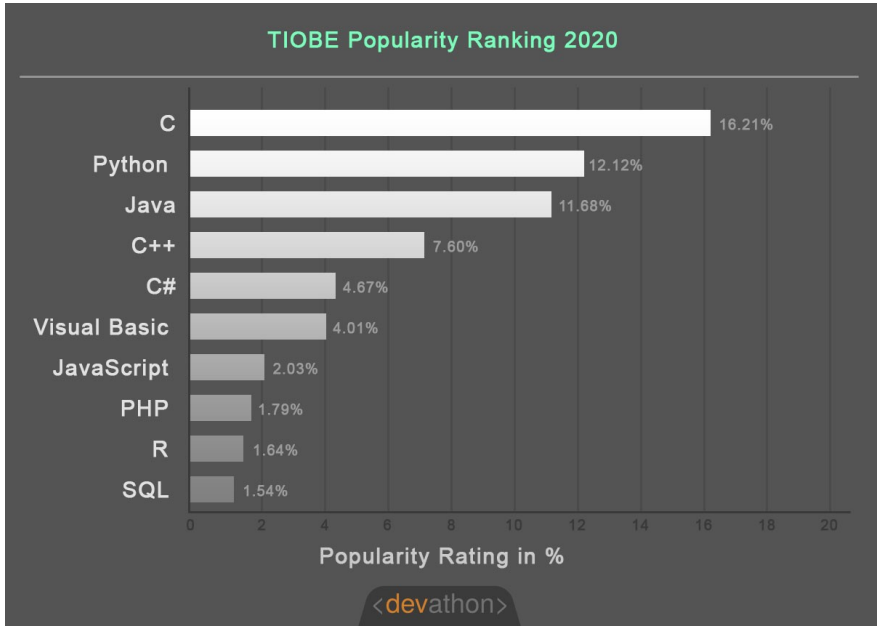
프로그래밍 언어

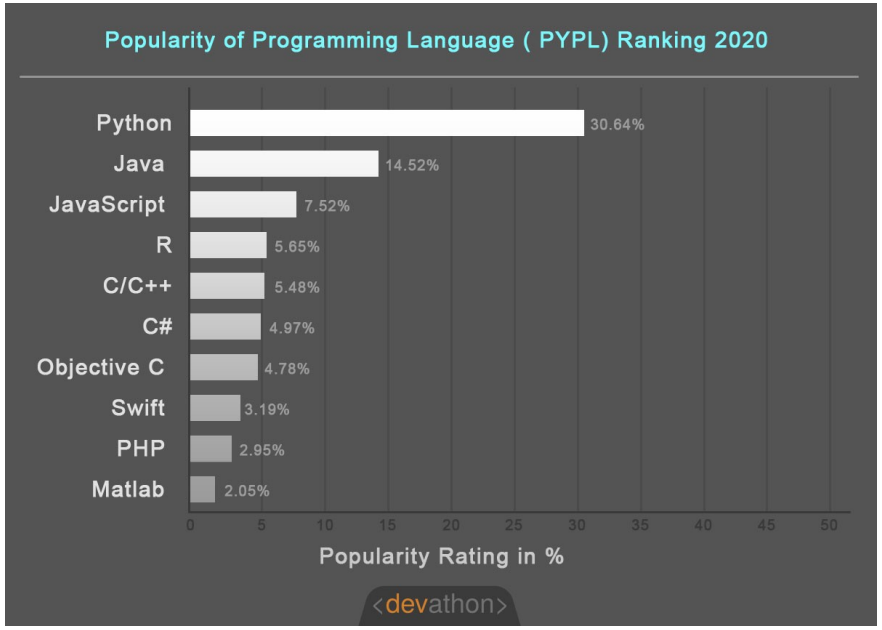
- 컴퓨터 프로그램을 작성하는 형식 언어
  ex) Python, C, C++, Java, FORTRAN, Lisp, Scheme, ML, etc.
- 프로그래밍 언어는 문법과 의미로 구성
  - Syntax: 프로그램이 문법적으로 타당한 구조인가
  - Semantics: 프로그램이 유효한 의미를 가지는가

- 프로그래밍 언어의 기본 제공 기능
  - 입력: 파일이나 키보드 마우스 등의 장치로부터 데이터 입력
  - 출력: 데이터를 컴퓨터 화면에 출력하거나 파일 또는 외부 장치로 저장
  - 처리: 연산/순차/조건/반복 처리

파이썬(Python)이란?

- 1991년 귀도 반 로섬(Guido Van Rossum)이 발표한 프로그래밍 언어
  - 플랫폼에 독립적
  - 기계학습, 딥러닝의 발전과 함께 점유율이 상승 후 C, Java와 함께 Top3 점유율 차지
  - 구글 사내 표준 언어 중 하나
    (C++, Java, Javascript, Python, …)
  - python 2 & python 3 버전 존재 (호환x)





파이썬의 장점

- 간결한 코드, 쉬운 문법: 직관적이고 간결한 문법 형식으로 초보자도 쉽게 배울 수 있음
- 다양성: 다양한 응용 프로그램 개발에 범용적으로 사용 가능
  - 데이터 분석
  - 웹 개발
  - 시스템 관리
  - 기계 학습, 딥러닝
  - 소프트웨어 테스팅
- 풍부한 라이브러리 및 개발 도구
  - Numpy: 다차원 행렬 계산과 수치 계산 도구를 갖춘 파이썬 라이브러리
  - Scipy: 과학 컴퓨팅 및 기술 프로젝트에 사용되는 라이브러리 (선형대수, 신호 처리 등)
  - Tensorflow: 기계 학습 모델 구축 작업을 용이하게 하기 위해 구글에서 만든 플랫폼
  - Keras: 딥러닝 모델의 생성과 운영을 용이하게 하는 라이브러리
  - Matplotlib: 데이터 시각화를 용이하게 하는 프레임워크
  - Pandas: 데이터 계산 및 분석에 사용되는 라이브러리
- 대규모 개발자 커뮤니티: 다양한 참고자료와 정보가 존재
  - Stackoverflow: 프로그래밍 관련 Q&A 사이트
  - Github: 오픈 소스 코드 공유 및 협업 서비스

프로그래밍 과정

1. 프로그램 로직 설계

2. 코딩

3. 컴파일

4. 실행 및 디버깅

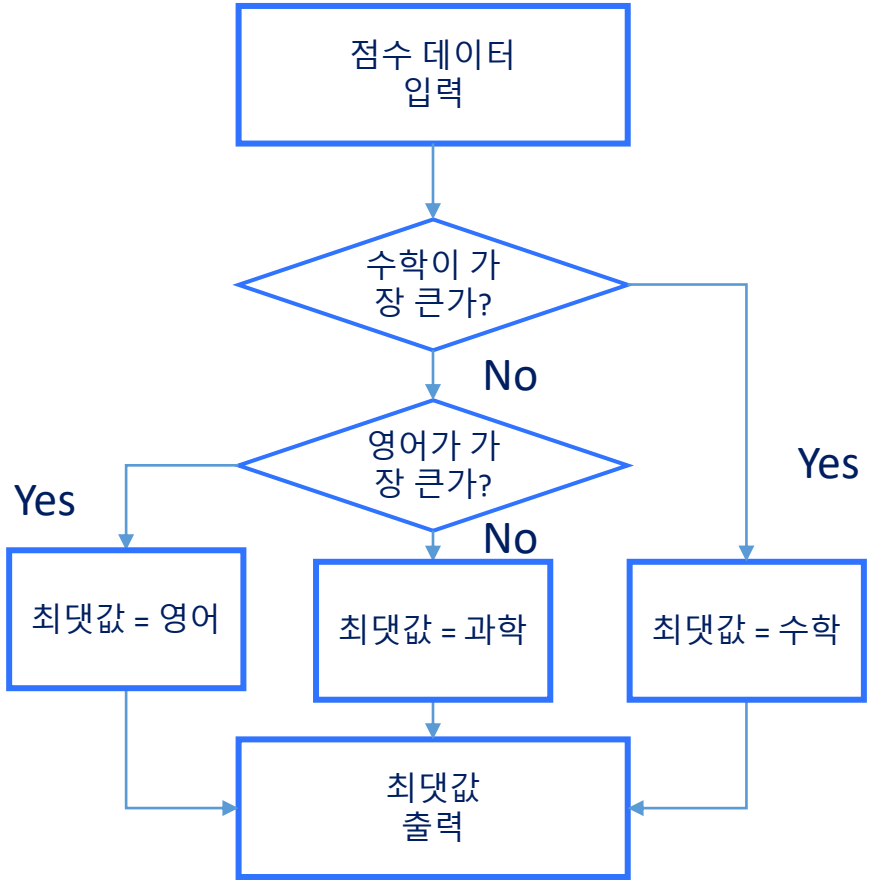
프로그램 로직 설계

- 프로그래밍 과정 중 가장 중요한 핵심 단계
  - 설계의 정밀도에 따라 실행 및 검증단계에서 소요되는 시간이 결정

- 알고리즘(Algorithm)
  - 어떠한 문제를 풀기 위한 실행 절차나 방법을 공식화한 형태로 표현한 것

- 로직 설계 기법
  - 순서도(Flowcharts)
  - 의사코드(Pseudocode)

Flowchart & Pseudo code



Flowcharts

1. 과목별 점수를 데이터로 입력한다.
2. 수학 점수가 영어, 과학보다 크다면,
3. 수학 점수를 최댓값으로 정한다.
4. 영어 점수가 수학, 과학보다 크다면,
5. 영어 점수를 최댓값으로 정한다.
6. 과학 점수가 수학, 영어보다 크다면,
7. 과학 점수를 최댓값으로 정한다.
8. 최댓값을 출력한다

Pseudo Code

코딩

- 선언(declaration) : 프로그램에서 사용될 자료형(data type)를 컴퓨터에게 알려주는 단계
  - 데이터는 사용되기 전에 반드시 미리 선언 (일반적으로 코딩 작업 앞부분에서 선언)
  - 자료형을 선언함으로써 메모리에 차지할 저장 공간과 쓰임새를 컴퓨터에게 미리 알려줌

- 입력(input) : 컴퓨터에 데이터를 입력하는 것
  - 키보드, 마우스, 펜 등 사람의 입력
  - 텍스트 파일, 데이터베이스, 웹 페이지 업로드 등 파일 형태
  - 카메라, Lidar 센서 등 센서 측정 값의 입력

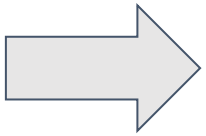
- 처리(processing) : 프로그램에 의해 수행되는 작업

- 출력(output) : 결과 또는 답
  - 컴퓨터 화면에 표시
  - 파일로 저장
  - 스피커, 프린터 등 출력 장비로의 결과 전달

코딩

1. 과목별 점수를 데이터로 입력한다.
2. 수학 점수가 영어, 과학보다 크다면,
3. 수학 점수를 최댓값으로 정한다.
4. 영어 점수가 수학, 과학보다 크다면,
5. 영어 점수를 최댓값으로 정한다.
6. 과학 점수가 수학, 영어보다 크다면,
7. 과학 점수를 최댓값으로 정한다.
8. 최댓값을 출력한다

Pseudo Code



```python
# The green lines are comments
if __name__ == "__main__":
    # The program starts from here
    math = 90
    english = 80
    science = 75

    maximum_score = 0
    if math >= english and math >= science:
        maximun_score = math
    elif english >= math and english >= science:
        maximun_score = english
    else:
        maximun_score = science

    print("최고점은 %d점 입니다." % maximum_score)
```

선언 & 입력

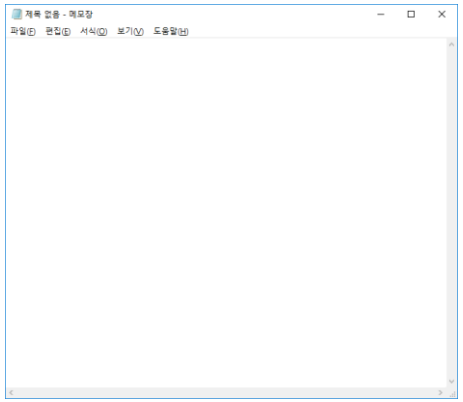
처리

출력

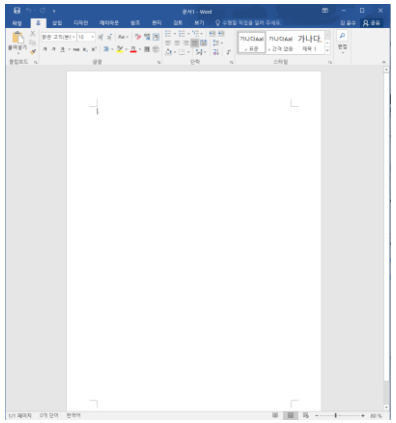
Python Code

Integrated Development Environment

- 통합 개발환경(IDE, Integrated Development Environment)
  - 편집기, 컴파일러 등 프로그램 개발에 필요한 도구들을 결합한 소프트웨어
- 통합 개발환경 예시
  - PyCharm: 대표적인 python 개발환경
  - Visual Studio Code: Microsoft에서 개발한 개발환경. 파이썬뿐 아니라 다양한 언어 지원
  - Atom
  - Jupyter Notebook: 웹 브라우저에서 파이썬 코딩, 실행 지원



vs



Text Editor

IDE

프로그램 컴파일

- 작성한 프로그램은 컴퓨터가 읽을 수 있는 기계어(machine language)로 변환(compile)된 후 실행 할 수 있다.

- 컴파일러(compiler)
  - 대표 언어: C, C++, Java
  - 프로그래밍 언어로 작성된 프로그램을 기계어로 변환하여, 컴퓨터가 실행 가능한 형태(.exe, .bat 등)로 만든다.

- 인터프리터(interpreter)
  - 대표 프로그램 언어: Python
  - 프로그램의 명령어 하나씩 기계어 명령어로 변환하여 실행한다.

실행 및 검증

- 프로그램을 실행하고, 원하는 결과물을 얻었는지 검증
  - 프로그램에 오류가 있는 경우, 디버깅(debugging)을 통하여 프로그램 수정 후 재검증 필요

- 프로그램 오류의 종류
  - 문법 오류(syntax error) : 오류 수정이 쉬움(즉, 버그 잡기가 쉬움)
    - ex) print('Hello') (O), pnint('Hello') (X)
  - 런타임 오류(runtime error) : 예외(exception) 처리를 해야 함
    - ex) 'Hello' + 1234
  - 논리 오류(logical error) : 오류 없이 프로그램이 실행되지만 예상한 결과가 나오지 않는 경우
    - ex) x = 1/6 ➔ x = 1 / 2 * 3

- 문법 오류는 컴파일 단계에서, 런타임 오류와 논리 오류는 실행 단계에서 발생