

1. 컴퓨터의 이해

1. 데이터와 정보

- 데이터의 개념

- 어떤 현상이나 사실에 대한 설명 또는 설명의 집합
 - 붉다, 밝다, 뜨겁다, **3500도**
 - 질적 데이터 : 붉다, 밝다, 뜨겁다 양적 데이터 : **3500도**

- 데이터의 정의

- 데이터(**data**)
 - 1) 관찰이나 실험, 조사로 얻은 사실 2) 이론을 세우는 데 기초가 되는 사실 또는 바탕이 되는 자료
 - 관찰이나 측정을 통해 얻은 수치, 문자 형태의 표현할 수 있는 질적(**quality**) 또는 양적(**quantity**) 값

- 정보의 개념

- 정보(**information**)
 - 문제 또는 질문을 해결하기 위해 사용할 수 있는 데이터와 데이터의 집합
 - 관찰과 측정을 통해 얻은 데이터를 처리(정렬, 합산, 군집화 등)하여 실제 문제 해결에 도움이 되는 데이터 또는 결과물
 - 1) 어떤 데이터나 소식을 통하여 얻은 지식이나 상태의 총량 2) 관찰이나 측정을 통하여 수집한 자료를 문제 해결에 도움이 될 수 있도록 정리한 지식

- 정보 처리 과정

- 데이터는 현상에 대한 관찰과 측정으로 생성
- 데이터는 기록된 사실이지만, 직접적으로 문제를 해결하는데 도움이 되지 않음
- 처리, 가공을 통해 정보로 변환

- 관찰과 측정

- 관찰
 - 사물 또는 현상을 파악하는 행위
 - 질적 데이터 얻음
- 측정
 - 절대적인 기준을 맞춰 현상을 수치로 나타내는 과정
 - 연속적 데이터(**3500도**) / 범주적 데이터(**A급 화염**)

- 처리와 가공

- 수집된 데이터의 집합을 정렬, 군집화, 계산 또는 표현방법 변경 등의 데이터를 변환하여 정보를 생산하는 과정

2. 컴퓨터의 개념

- 컴퓨터의 정의

- 정의된 방법에 따라 입력된 데이터를 자동으로 처리하여 정보를 생산하는 기계

- 데이터의 기본 단위

- 비트(**bit, binary digit**)
- 바이트(**byte**)

- 컴퓨터와 2진법

- 컴퓨터는 트랜지스터를 통해 2진법 숫자로 데이터를 표현하여 저장 및 처리

- 아날로그와 디지털

- 아날로그(**analog**)
 - 사물이나 개념을 연속적인 물리량 값으로 표현
 - 질적 데이터 또는 양적 데이터로 표현
- 디지털(**digital**)
 - 사물이나 개념을 이산적인 값으로 근사하여 표현
 - 양적 데이터로 표현

- 숫자 데이터의 표현

- 10진수, 12진수 등 --- (진법 변환) ---> 2진수

- 문자 데이터의 표현

- 한글, 알파벳 등 --- (인코딩 체계) ---> 2진수

- ASCII 코드표

- 코드를 통해 정보를 교환하기 위한 미국 표준
- ASCII 코드표는 **문자(Character)**와 **숫자(정수, 0~127)**를 일대일로 매핑한 표

컴퓨터는 숫자라면 진법변환 문자라면 ASCII 변환

↳ 그렇기 때문에 컴퓨터는 숫자, 문자의 구별 정보를 별도 공간에 저장한다

↳ 데이터 타입(data type)의 존재

ASCII 코드표는 왜 8비트(=1바이트)가 아니라 7비트를 사용하는가?

↳ 7비트 (128개의 문자)만 사용 남은 1비트는 통신 제어나 오류 검출 등에 쓰려고 비워둔 것

↳ 1비트(=8번째 비트)

패리티 비트(Parity Bit) : 통신 오류를 감지하기 위한 비트 (짝수/홀수 비트 계산)

확장 ASCII : ISO-8859-1 등에서 1비트를 사용해서 문자 개수를 256개로 확장

제어 신호 : 일부 통신 장비에서 제어 목적으로 사용

3. 컴퓨터와 프로그램

프로그램의 이해

컴퓨터는 정보를 처리하기 위한 방법과 과정을 자의적으로 결정 불가능

처리 방법과 과정이 정의된 프로그램을 사용

↳ 프로그램 (program)

↳ 프로그램이란 컴퓨터가 어떠한 작업을 자동으로 처리할 수 있도록 처리 방법 및 순서를 컴퓨터가 이해할 수 있는 언어 형태로 기술한 것

↳ 유사한 유형의 여러 문제를 추상화시킨 알고리즘(algorithm)을 구현한 결과물

정렬 알고리즘

↳ 모든 카드를 오름차순으로 정렬하는 문제

선택정렬 알고리즘

알고리즘 예시

1. 정렬되지 않은 공간에서 가장 작은 수 하나를 찾는다.

2. 맨 앞의 카드와 맞바꾼다.

3. 카드가 정렬되었는지 확인한다.

4. 정렬이 되었으면 완료

5. 정렬이 되지 않았으면 다시 첫번째 정렬로 돌아간다.

알고리즘의 정의

문제를 풀기 위한 단계별 절차를 체계적 명령의 형태로 기술한 것

주어진 명령어를 처리하는 컴퓨터에게 문제를 해결하도록 만드는 정형화된 절차

프로그래밍 언어

사람과 컴퓨터 사이의 의사소통 도구

프로그래머는 "컴퓨터가 이해할 수 있는 언어"를 사용하여 프로그램을 작성

프로그래밍 언어의 분류

↳ 기계어(1c) -> 어셈블리어(2c) -> 자연어와 유사(3c) -> 자동화(4c)

프로그램의 실행

고급언어로 작성된 프로그램은 기계어 밖에 인식할 수 없는 컴퓨터에 의해 실행 불가능

↳ 컴파일러 또는 인터프리터를 사용

파이썬은 컴파일러, 인터프리터 중에 어떤 것을 사용하는가?

✓ 기본적으로 인터프리터 언어

↳ 실행 시, 바로 한 줄씩 해석하며 실행합니다. → 즉, 컴파일 없이 실행 가능

☑ 파이썬은 다단계 실행 구조

↳ 파이썬 소스 코드 (.py) → 파이썬 인터프리터가 이를 **바이트코드(.pyc)**로 컴파일함 → 그 바이트코드를 **Python 가상 머신(PVM)**이 인터프리팅 실행

파이썬은 인터프리터 언어이지만, 내부적으로는 바이트코드 컴파일 → 인터프리팅 구조를 갖는다.

2. 컴퓨터의 구성

1. 하드웨어와 소프트웨어

컴퓨터의 구성요소

하드웨어

↳ 컴퓨터를 구성하고 있는 물리적 부품

소프트웨어

↳ 다양한 장치들을 동작시켜 특정 작업을 해결하는 프로그램

사람의 문제해결 과정

1. 입력
2. 기억&저장
3. 제어
 - 문제 푸는 과정을 먼저 찾아내고 이후에 문제를 풀기 시작한다.
 - $5 + (2 \times 4)$
4. 연산
5. 출력

하드웨어

- 기능에 따라 입력장치, 기억-저장장치, 제어장치, 연산장치, 출력장치로 구분
- 폰 노이만 구조

소프트웨어

- 각각의 고유한 기능을 수행하는 하드웨어가 언제 어떻게 동작하여 문제를 해결할 수 있는지 지시하는 명령어 집합
 - 응용 소프트웨어
 - 사용자의 업무나 목적에 맞게 문제 해결을 위한 처리 절차를 표현한 명령어 집합
 - 시스템 소프트웨어
 - 하드웨어를 제어, 관리하여 응용 소프트웨어를 실행할 수 있는 환경을 제공

2. 하드웨어의 이해

입력장치

- 명령과 데이터를 컴퓨터에 전달하는 장치
- 컴퓨터가 처리할 수 있는 2진수 형태로 변환
- 입력장치의 종류
 - 키보드, 마우스, 스캐너, 터치스크린
- 특수 입력장치
 - 마이오(Myo) 립 모션(Leap Motion)

출력장치

- 출력 장치의 종류
 - 모니터, 스피커, 프린터, 초 지향성 스피커, 3D 프린터
- 특수 출력 장치
 - HUD : 현실의 정보와 가상 정보를 함께 처리
 - AR(Augmented Reality)
 - HMD : 가상 정보만 처리

기억(저장) 장치

- 명령과 데이터를 기억(저장)하는 하드웨어
- 역할에 따라 주기억장치 보조기억장치로 구분
- 기억(저장) 장치의 종류
 - ROM, RAM, 플로피디스크, 광학디스크, 플래시 드라이브
 - 플로피디스크, 자기디스크, 광학디스크는 왜 전기가 차단되어도 데이터가 남아 있을까?
- 특수 기억(저장) 장치
 - RAM을 이용한 저장장치 빠른 속도, 외부 충격에 강함, 적은 전력소모가 강점 최근 하드디스크를 대체할 저장장치로 각광
- 제어와 연산 장치
 - 현대 컴퓨터는 제어와 연산 장치를 구분하지 않는다.
 - CPU가 제어와 연산 모두를 수행함
 - 명령어와 데이터를 읽고 데이터를 연산 또는 처리하는 장치
 - 레지스터(register)
 - CPU 내부에서 초고속으로 데이터를 저장하거나 읽어냄
 - 제어 유닛 (control unit)
 - 연산 논리 유닛(ALU)

3. 소프트웨어의 이해

소프트웨어

- └ 소프트웨어는 고유한 기능을 수행하는 하드웨어가 언제 어떻게 동작하여 문제를 해결할 수 있는지 절차를 지시하는 명령어 집합
 - └ 응용 소프트웨어
 - └ 시스템 소프트웨어
 - └ 다양한 장치들을 서로 유기적으로 동작시켜 특정 작업을 수행할 수 있는 환경을 조성하는 프로그램
 - └ 운영체제(커널)
 - └ 사용자가 컴퓨터를 효율적으로 운영/관리/사용할 수 있도록 하드웨어를 제어하는 소프트웨어
 - └ 컴파일러
 - └ 소스코드를 컴퓨터가 이해할 수 있는 기계어로 번역하는 소프트웨어
 - └ 유틸리티
 - └ 추가적인 기능을 제공하여 사용자가 컴퓨터를 효율적이고 편리하게 관리할 수 있도록 지원해주는 소프트웨어
- └ 운영체제의 개념
 - └ 응용 소프트웨어가 효과적으로 작동할 수 있는 환경을 조성
- └ 운영체제의 역할
 - └ 컴퓨터의 자원(하드웨어)를 효율적으로 관리하고 응용 프로그램에 자원을 할당
 - └ 1) 사용자 인터페이스 2) 프로세스 관리 3) 네트워크 관리 4) 기억장치 관리 5) 입출력장치 관리
 - └ 사용자 인터페이스
 - └ 컴퓨터와 사용자를 매개하여 상호작용하는 방법을 제공
 - └ CLI(Command Line Interface)
 - └ GUI(Graphic User Interface)
 - └ GUI 기반의 운영체제 / CLI 기반의 운영체제
 - └ 프로세스 관리
 - └ 프로세스는 실행되고 있는 상태의 프로그램
 - └ 프로그램은 보조기억장치에 저장되어 있는 코드
 - └ 여러 프로그램 실행이 요청되면 한정된 자원(기억장치 등)을 효과적으로 사용하도록 조율
 - └ 네트워크 인터페이스
 - └ 컴퓨터는 네트워크를 통해 상호 데이터 교환
 - └ 응용 프로그램이 통신할 수 있는 환경 제공 및 통신 장치(하드웨어) 관리
 - └ 기억, 저장장치 관리
 - └ 보조기억장치에 저장된 컴퓨터 프로그램은 실행되기 위해서 주기억장치에 적재
 - └ 주기억장치의 크기는 보조기억장치의 크기보다 매우 작기 때문에 효율적 관리가 요구

3. 파이썬의 이해