문화 A0007

데이터사이언스입문

김 태 완

kimtwan21@dongduk.ac.kr

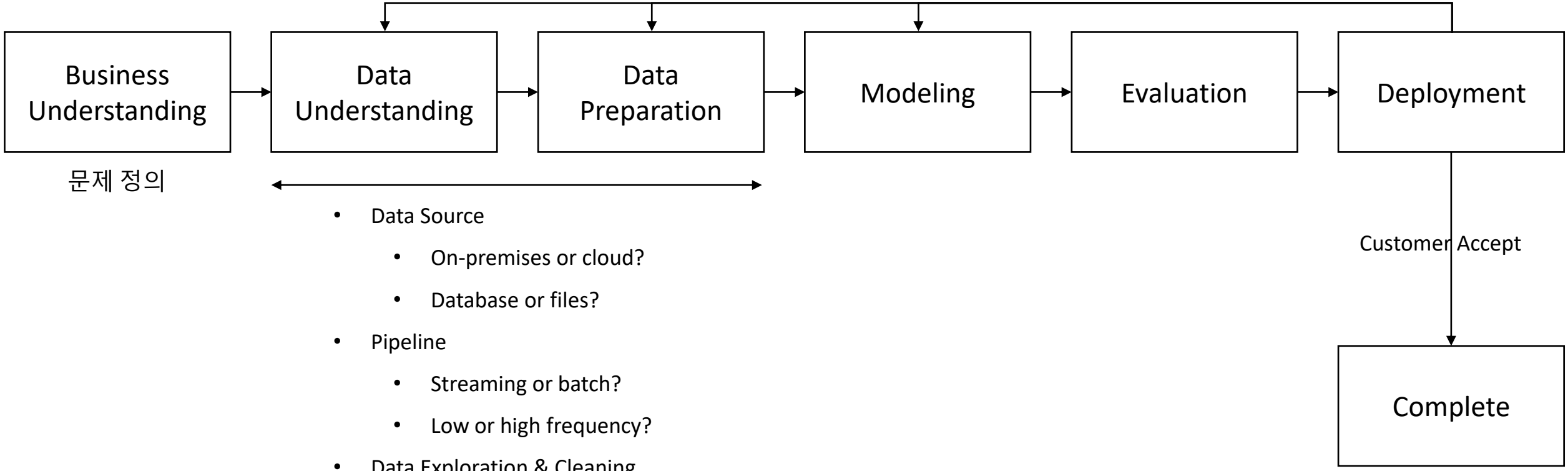
데이터사이언스 프로세스

산업 간 연구 결과, 평균적으로 기업 내 정형 데이터 중 의사결정에 활발히 이용되는 데이터는
절반 이하에 불과하며, 비정형 데이터가 분석되거나 사용되는 비율은 1% 이하다.
70% 이상의 직원들이 불필요한 데이터에 액세스하고 있으며,
데이터 분석가들은 80%의 시간을 데이터 검색과 준비에 쏟고 있다.
기업의 데이터 기술은 아직 충분한 수준에 미치지 못하고 있다

하버드 비즈니스 리뷰(Harvard Business Review)

데이터사이언스 프로세스

- Data Science Process



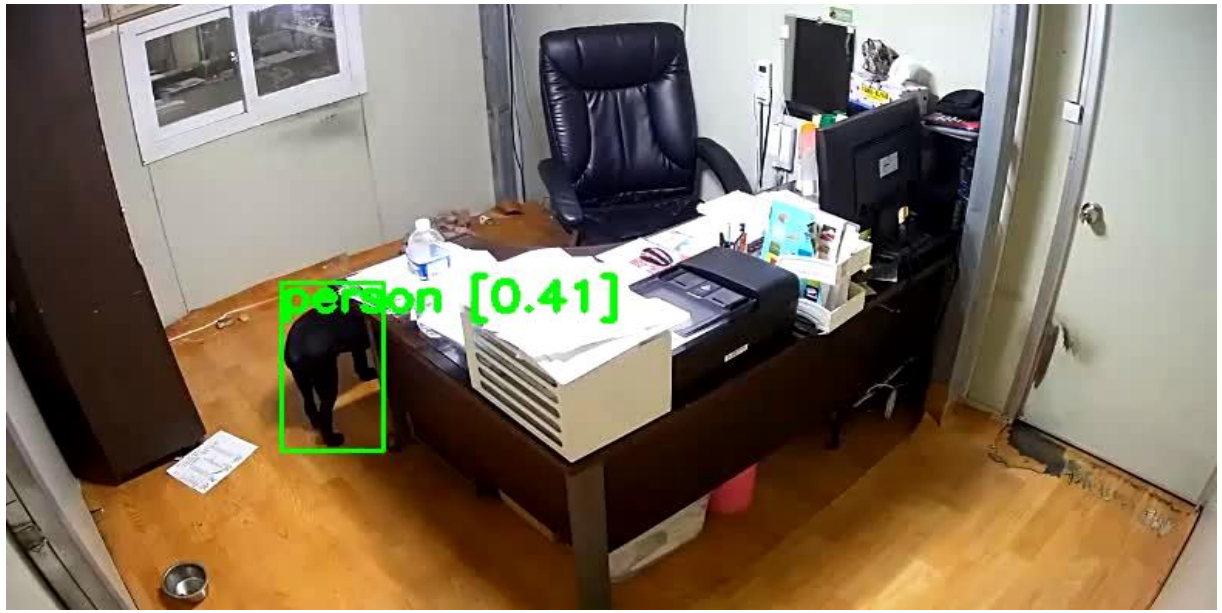
- Data Source
  - On-premises or cloud?
  - Database or files?
- Pipeline
  - Streaming or batch?
  - Low or high frequency?
- Data Exploration & Cleaning
  - Structured or unstructured?
  - Data Validation

데이터사이언스 프로세스

- Business Understanding
  - 흥미로운 문제를 발굴 하자 (매출 증대, 비용 감소 등)
  - 예시 : 쿠팡 로켓 배송 → 매출 증가 (배송의 속도는 물류센터 내 제품 배치에 따라 결정)
    - 잘 팔리는 상품, 빨리 팔리는 상품을 배송준비가 가장 빨리 이뤄질 수 있는 구역으로 나눠 배치
    - 물류센터 내 제품 배치도 최적의 공간 효율을 낼 수 있도록 각각의 장소에 진열되는
      모든 상품의 위치와 진열장소를 시스템이 실시간으로 계산해 결정

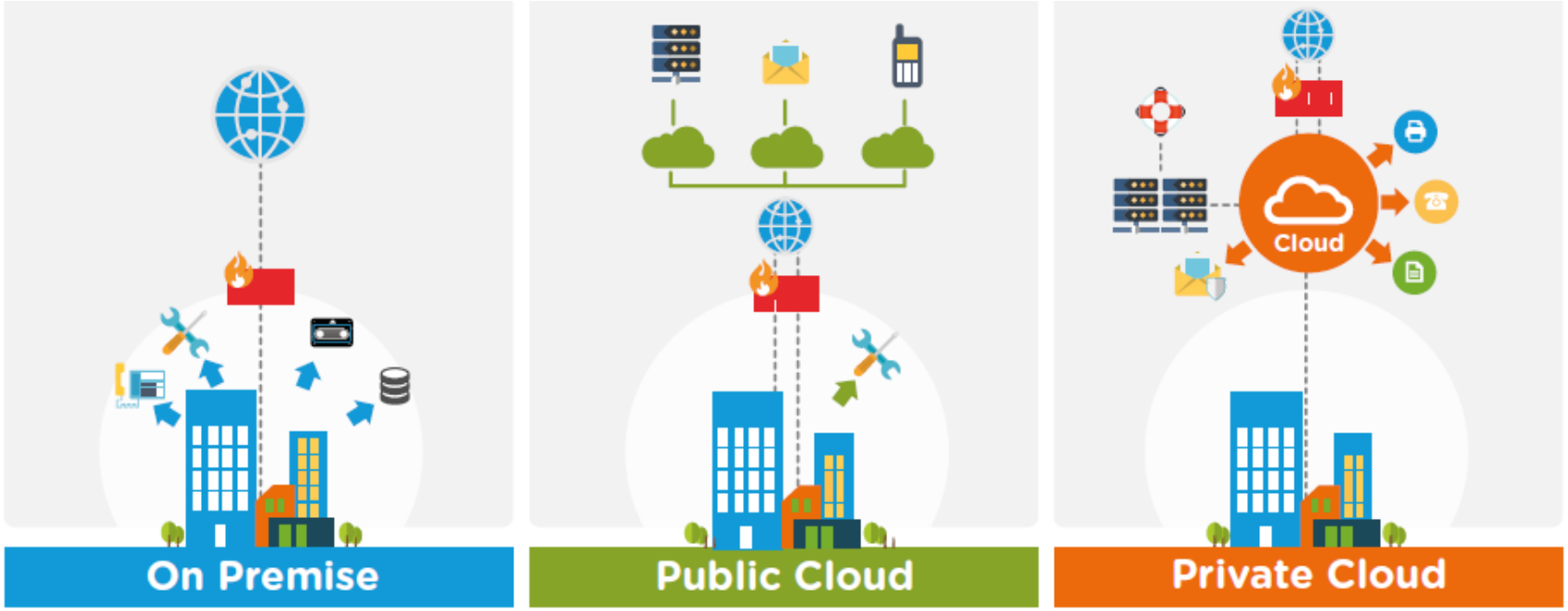
데이터사이언스 프로세스

- Business Understanding
  - 흥미로운 문제를 발굴 하자 (매출 증대, 비용 감소 등)
  - 예시 : 보안 업체 출동 시스템에서 실제 침입이 발생한 비율은 1%에 불과 → 인건비 포함 비용 과다



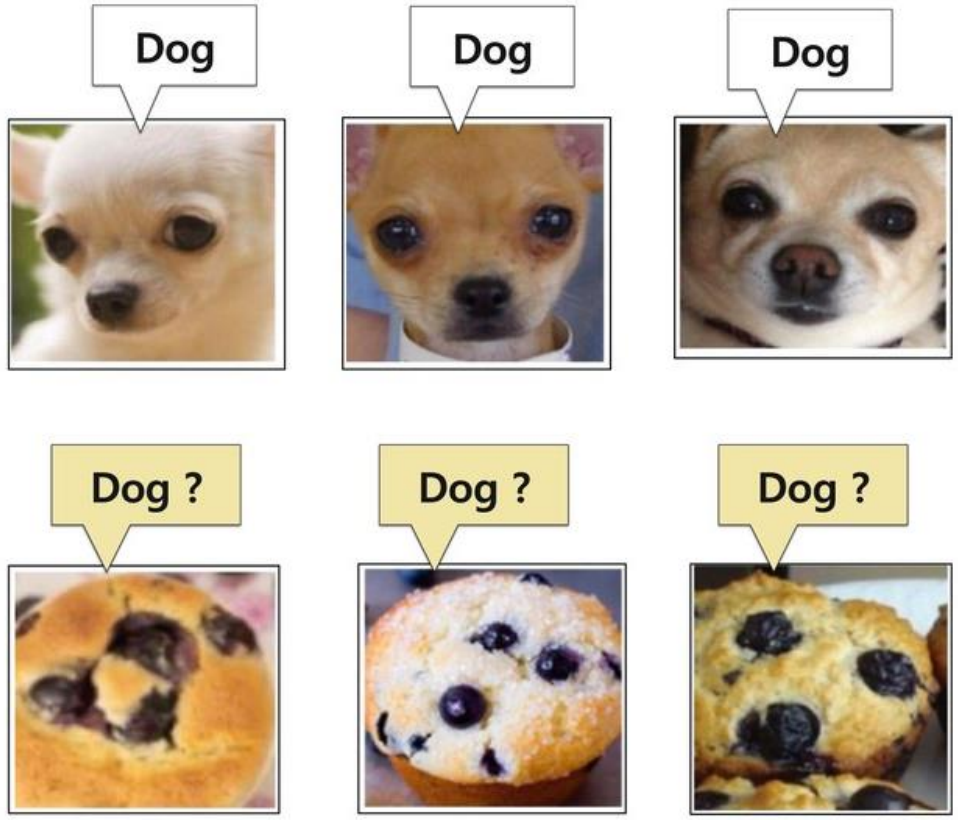
데이터사이언스 프로세스

- Data Understanding
  - 해결하고자 하는 문제와 관련이 있는 데이터 찾기
  - 데이터의 Quality가 매우 중요 → **Garbage-in Garbage out**
  - 고려 사항 예시
    - 현장에 데이터가 장비에 보관되어 있는 지 클라우드 상에 있는지?
    - 데이터가 잘 수집되는 시스템이 있는지?
    - 데이터가 현재에도 실시간으로 streaming으로 저장되고 있는 지, batch 형태로 처리되고 있는 지?



데이터사이언스 프로세스

- Data Preparation
  - Transformation: 데이터를 문제 해결에 용이한 형태로 변형한다면 더 빠르고 정확하게 해결 가능
  - Cleaning : 실제 현실 데이터 중에는 잡음 (noisy)이 있는 데이터가 생각보다 많음
  - 필요하다면 전문가의 지식을 적극 활용해야 함 (domain knowledge)

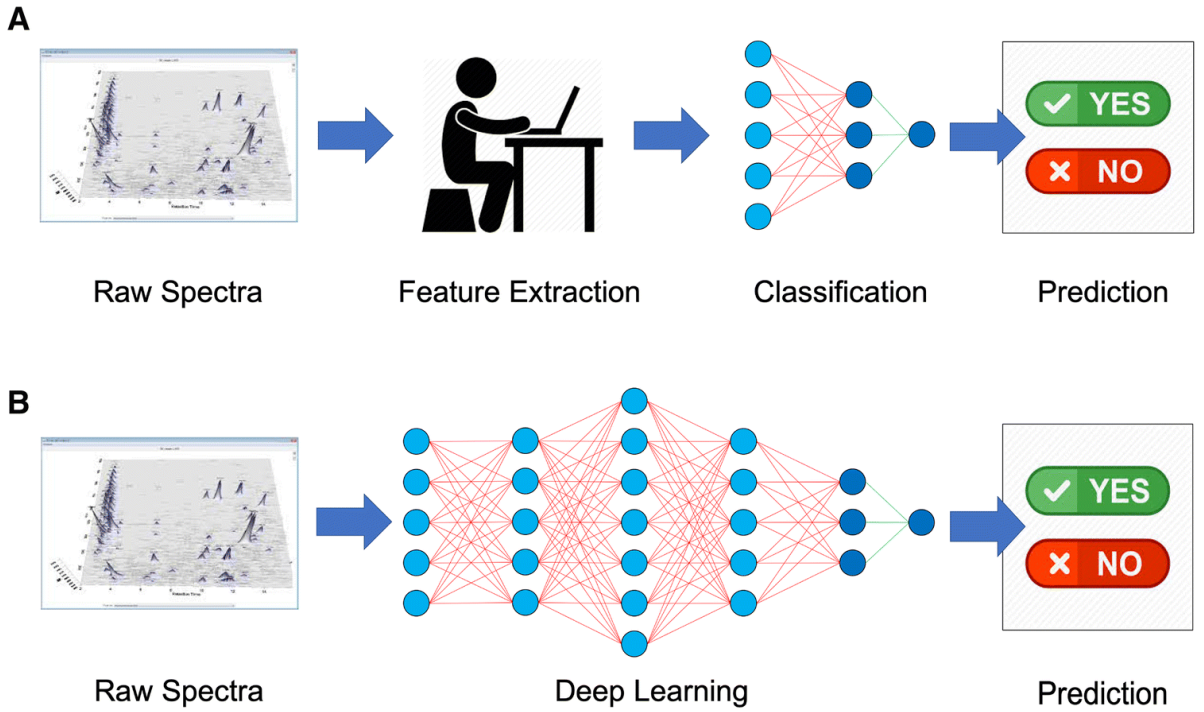


데이터사이언스 프로세스

- Data Preparation
  - 모델링 단계 이전에 충분히 데이터를 탐색하자
  - 데이터 시각화 도구 사용 권장 : 이 단계를 무시한다면 결국 다시 여기로 돌아오게 됨

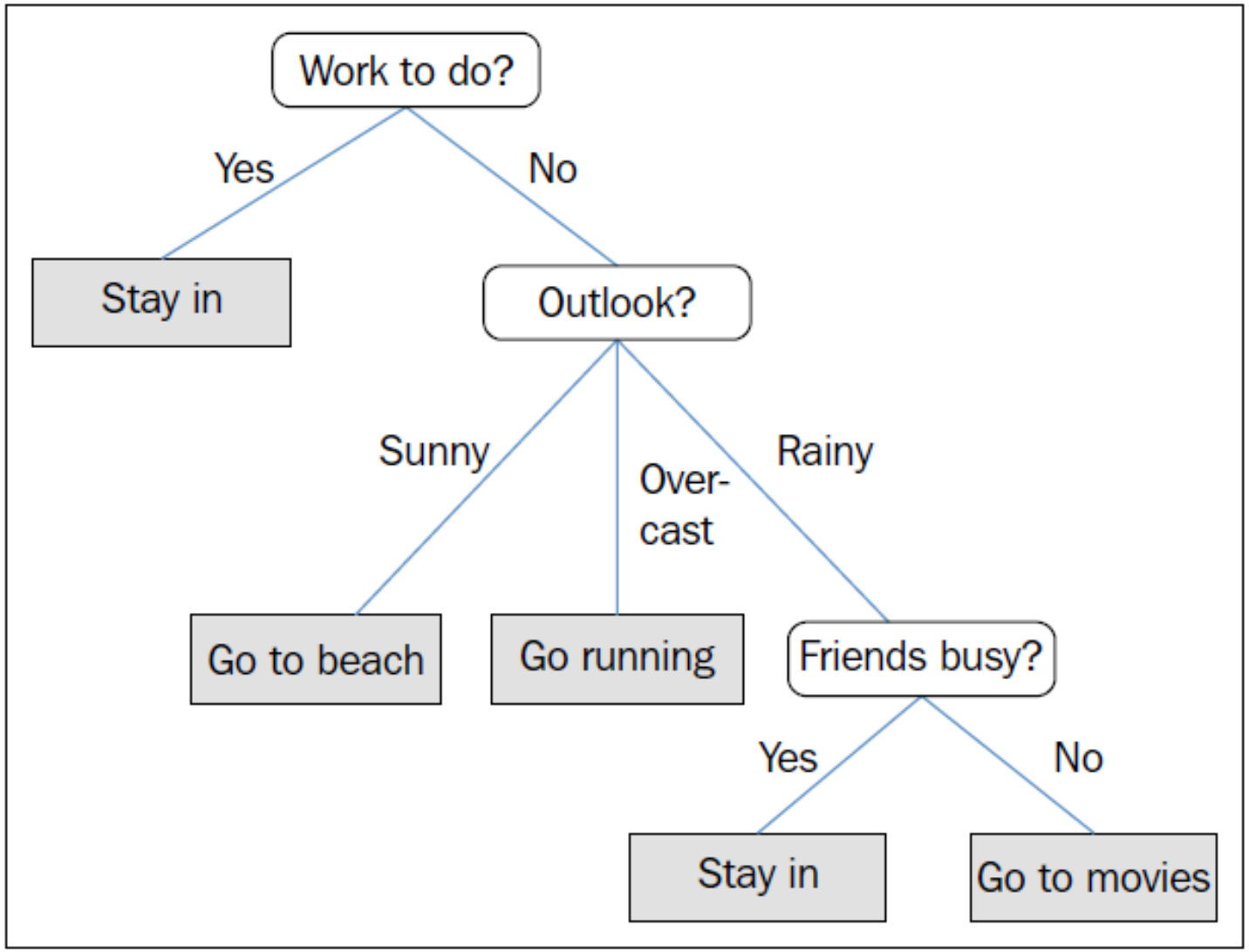
데이터사이언스 프로세스

- Modeling
  - Feature Engineering
    - Select important features and construct more meaningful ones using the raw data that you have
    - Feature selection → Feature extraction → Feature Engineering → Feature Learning
  - Predictive Modeling
    - Train machine learning models and evaluate their performance, and use them to make predictions



데이터사이언스 프로세스

- Modeling
  - 예시 : decision tree

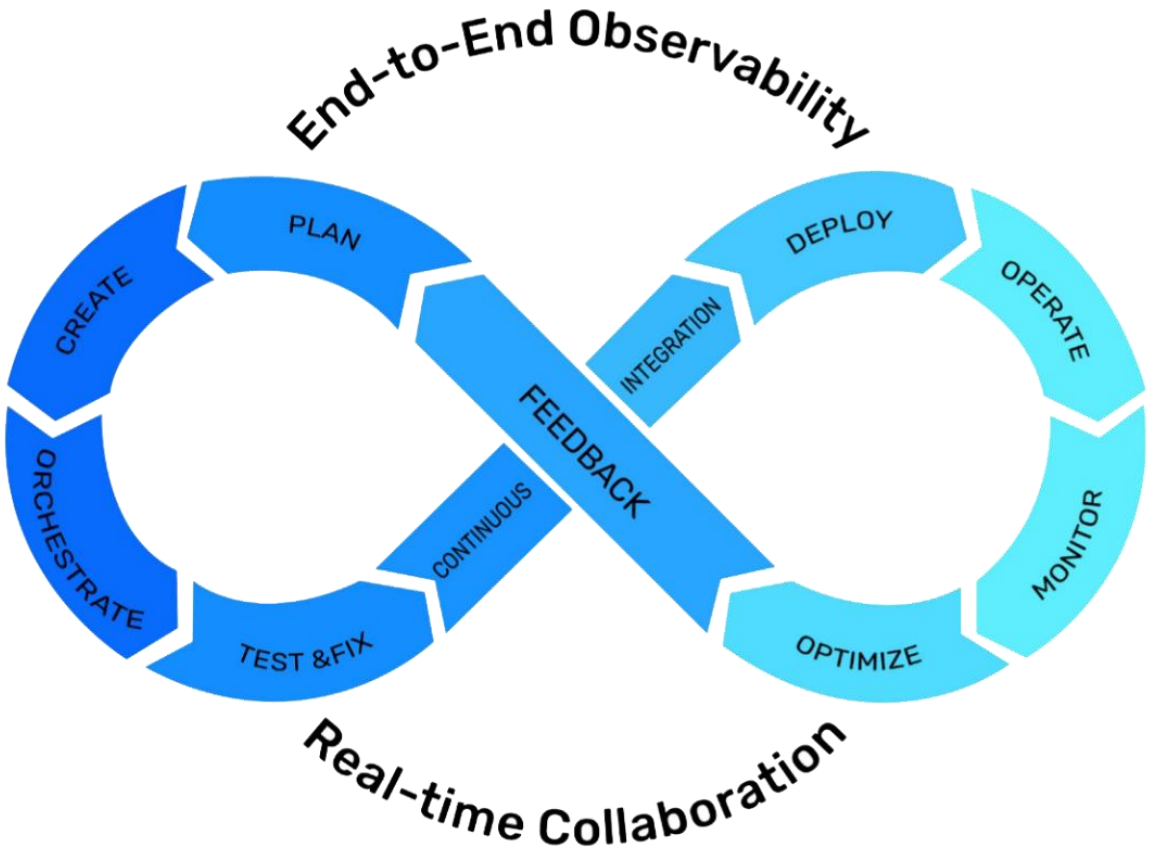


데이터사이언스 프로세스

- Evaluation
  - 모델링 결과 평가 및 개선안 수립
  - 산출물로는 모델결과 평가표와 해당 결과가 현장 (고객)에서 수용 가능한 수준인지 판단
  - 모델링 결과가 좋을 경우 현장 (고객)에게 충분히 설명 (검증) 후 배포
  - 모델링 결과가 만족스럽지 못한 경우 다시 앞의 단계를 반복해서 진행

데이터사이언스 프로세스

- Deployment
  - 구축한 모델을 실제 시스템에 적용하는 단계
  - 지속적인 성능 모니터링일 필요하며, 필요시 업데이트 작업 수행
  - 데이터옵스 (Dataops.) 필요



Computational Thinking

- 컴퓨팅 사고
  - 우리나라의 소프트웨어를 공교육에 자리매김한 계기
  - Jeannette Wing 교수의 논문에서 정의
    - Computational Thinking은 읽기, 쓰기, 셈하기와 더불어 누구나 갖춰야 하는 기본 역량
    - Computational Thinking은 컴퓨터 과학의 이론, 기술, 도구를 활용하여 현실의 복잡한 문제를 해결하는 사고 방식이다. 의학, 법, 경제, 정치, 예술 등 사회 모든 분야에서 보편적으로 필요한 핵심 능력이다.

컴퓨터 과학자처럼 생각하기

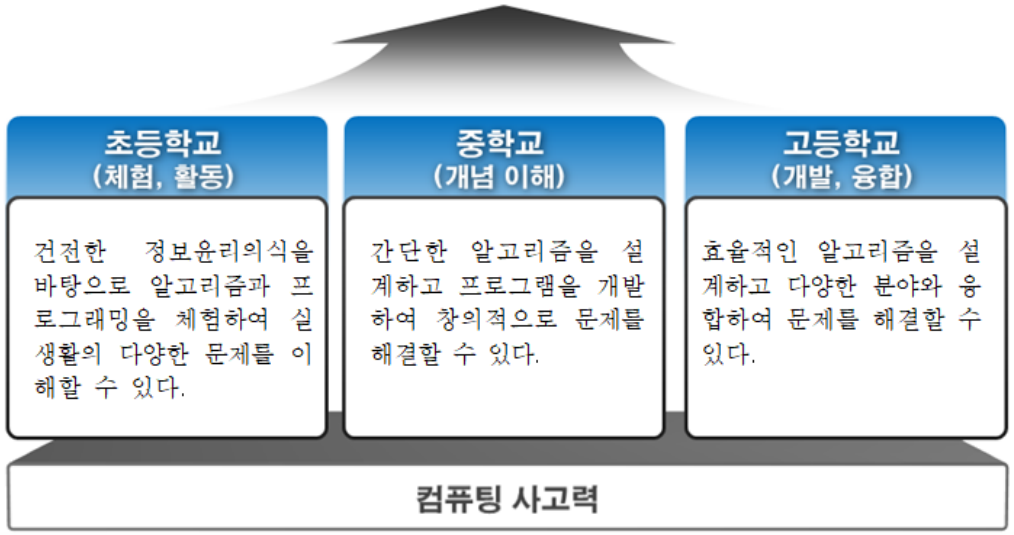
Jeannette Wing

Speaking at the World Economic Forum in Davos, Switzerland, on January 26, 2013.

Born	Jeannette Marie Wing
Nationality	American
Alma mater	Massachusetts Institute of Technology
	Scientific career
Fields	Computer science
Institutions	Columbia University, Carnegie Mellon University, University of Southern California
Thesis	*A Two-Tiered Approach to Specifying Programs* (1983)
Doctoral advisor	John Guttag[1]
Doctoral students	Greg Morrisett[1]
Website	cs.cmu.edu/~wing/

Computational Thinking

- 컴퓨팅 사고 정의 (교육부 운영지침)



○ **컴퓨팅 사고력**

컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력

○ **컴퓨팅 사고력의 구성 요소***

- 문제를 컴퓨터로 해결할 수 있는 형태로 구조화하기
- 자료를 분석하고 논리적으로 조직하기
- 모델링이나 시뮬레이션 등의 추상화를 통해 자료를 표현하기
- 알고리즘적 사고를 통하여 해결방법을 자동화하기
- 효율적인 해결방법을 수행하고 검증하기
- 문제 해결 과정을 다른 문제에 적용하고 일반화하기

Computational Thinking

- 컴퓨팅 사고
  - 컴퓨터가 문제를 해결하는 방식을 이해하고 이를 현실 문제 해결에 적용
  - 사람이 추상적으로 갖고 있는 생각을 컴퓨터가 자동적으로 처리하게 하는 과정
  - 추상화 : 실제 세계의 문제를 해결 가능한 형태로 표현하기 위한 사고 과정
  - 자동화 : 추상화 과정에서 만들어진 해결 모델을 컴퓨터가 이해할 수 있는 프로그래밍 언어로 표현
  - 컴퓨팅 사고력은 인간의 사고력으로 처리하기에 어렵거나 시간이 걸리는 작업을 대신해줌으로써 인간의 문제 해결 능력을 확장 시켜주는 역할

Computational Thinking

- 미국의 컴퓨터과학 교사 협회 (CSTA)는 컴퓨팅 사고력을 9가지의 세부 요소로 나누어 제시

구성 요소	정의
자료 수집 Data Collection	• 적절한 자료를 수집하는 과정
자료 분석 Data Analysis	• 자료의 의미를 이해하고, 패턴을 찾으며, 결론을 도출함
자료 표현 Data Representation	• 자료를 적절한 그래프, 차트, 글, 그림 등으로 도식화하고 조직화
문제 분해 Problem Decomposition	• 문제를 해결 가능한 수준의 작은 문제로 나눔
추상화 Abstraction	• 문제 해결을 위해 반드시 필요한 핵심 요소를 파악하고, 복잡한 것을 단순화함
알고리즘 및 절차 Algorithms & Procedures	• 문제를 해결하거나 어떤 결과를 이루기 위해 일련의 절차화된 순서를 취함
자동화 Automation	• 반복적이고 지루한 작업을 실행하기 위해 컴퓨터나 기계를 활용함
시뮬레이션 Simulation	• 하나의 절차를 표현하거나 모델화함 • 시뮬레이션은 모델을 활용한 실험을 실행하는 것을 포함
병렬화 Parallelization	• 공통의 목표에 도달하기 위해 과업들을 동시해 실행하도록 자원을 조직함

Computational Thinking

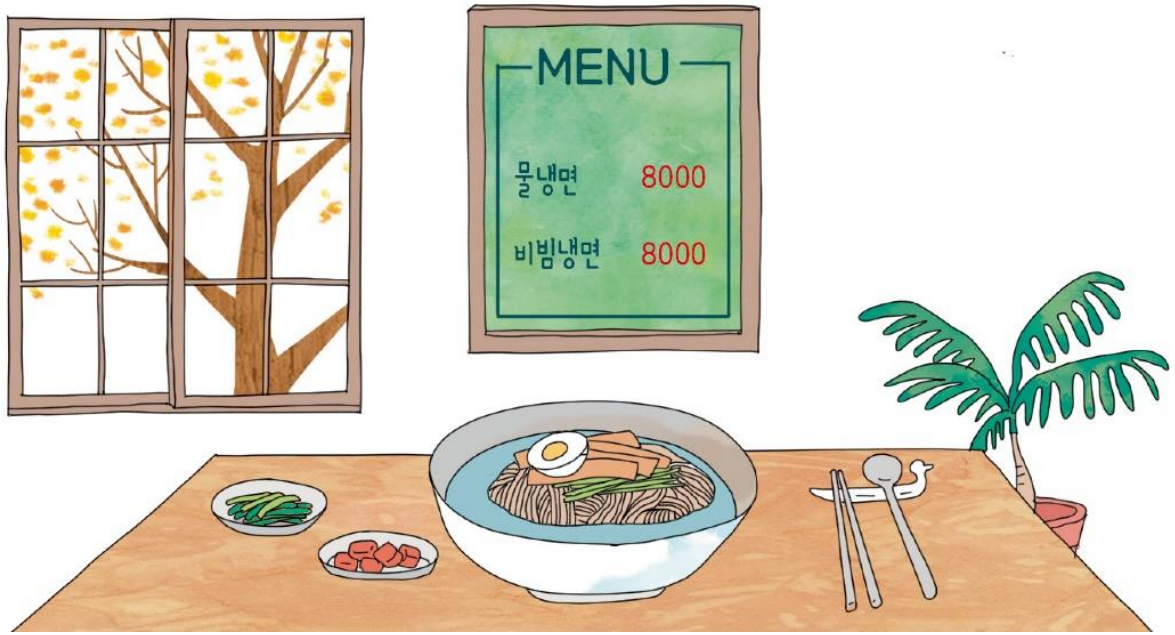
- MIT media lab.에 출판한 creative computing (Brennen. 2011)
  - 컴퓨팅 사고력의 요소를 개념, 연습, 관점의 측면에서 제시

개념 (concepts)	연습 (practices)	관점 (perspectives)
시퀀스 반복 병렬처리 이벤트 조건 연산 데이터	점진적인 시도와 개발 테스팅과 디버깅 재사용과 재조합 추상화와 모듈화	표현하기 연결하기 질문하기

*시퀀스: 프로그래밍 개념 과제를 위한 일련의 단계를 정의
*반복: 같은 시퀀스를 여러 번 실행하는 것
*병렬처리: 동시에 일이 일어나도록 만드는 것
*이벤트: 이것을 통해 다른 것이 일어나도록 만드는 것
*조건: 조건에 따라 결정하도록 만드는 것
*연산: 수학적, 논리적 표현식에 따라 계산하는 것
*데이터: 값을 저장, 검색, 수정하는 것

Computational Thinking

- 컴퓨팅 사고의 사례 #1 : 2진수와 식당
  - 10진수의 사칙연산을 회로로 구성하는 것은 매우 어려움
  - 2진수로 사칙연산을 한 후 결과만 10진수로 보여 줄 수 있는 계산기를 만든다면 효과적
  - 메뉴 수가 적을수록 회전율이 빠름

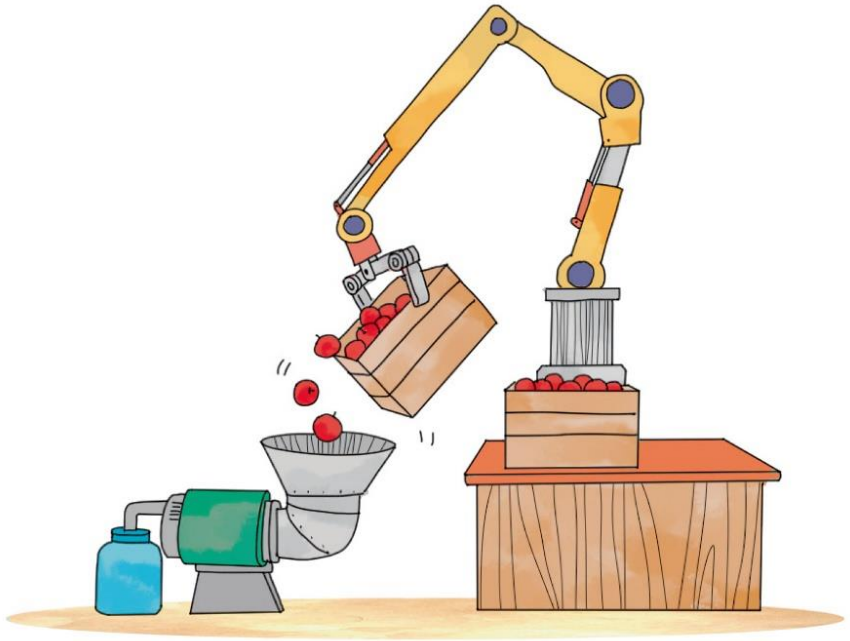


Computational Thinking

- 컴퓨팅 사고의 사례 #2 : 입출력 채널과 푸드코트
  - 입출력 장치를 느린 채널과 빠른 채널로 분리하여 사용하면 전체 입출력 효율이 올라감
  - 줄을 설 때 조리 속도가 다른 한식과 분식을 분리하면 효율적

Computational Thinking

- 컴퓨팅 사고의 사례 #3 : 버퍼 (buffer)와 공장
  - 사과를 잘게 부스는 것은 매우 빠르므로 기계에 사과를 하나씩 옮겨 집어넣는 것은 비효율적
  - 기계 속도를 맞추려면 사과를 큰 바구니에 담아 통째로 옮기는 것이 효율적
  - 버퍼 (buffer): 속도 차이가 많이 나는 두 장치 사이에 끼어서 속도 차이를 완화해 주는 장치



- 컴퓨팅 사고의 사례 #4 : 캐시 (cache)와 조미료 통
  - 요리를 할 때마다 대용량 포장에서 조미료를 조금씩 덜어 쓰는 것은 매우 불편하기 때문에 조미료를 조금씩 덜어 놓은 조미료 통을 사용하는 것이 효과적
  - 캐시 (cache): 앞으로 사용이 예상되는 것을 미리 가져다 놓은 것

Computational Thinking

- 컴퓨팅 사고의 사례 #5 : 병렬처리와 병원
  - 환자 수가 많은 병원에 의사 한 명이서 모든 환자를 진료하는 것보다, 공간을 일부 나누고 의사를 채용하여 더 많은 환자를 동시에 진료할 수 있게 됨
  - 병렬 처리 : 동시에 복수의 작업을 처리하는 기법

Computational Thinking

- 4 Key concepts of Computational Thinking
  - **Decomposition (분해)**
    - Break down data and problems into smaller parts
  - **Patter Recognition (패턴인식)**
    - Observe patterns and trends in data
  - **Algorithms (알고리즘)**
    - Determine what steps are needed to solve a problem
  - **Abstraction (추상화)**
    - Remove details and extract relevant information

Computational Thinking

- Decomposition (분해)
  - 말 그대로 어떤 큰 문제를 작은 단위로 나누어 생각해보며 해결하는 과정
  - 예시1: 여행 계획
    - 여행 계획을 세울 때 일정, 교통, 숙박, 식사, 볼거리, 예산 등으로 구분하여 계획하기
  - 예시2: 어버이날 선물
    - 어머니 선물과 아버지 선물을 구분하여 계획하기

- Pattern Recognition (패턴 인식)
  - 데이터에서 일정한 경향, 반복되는 규칙, 공통적 속성 등을 탐색하여 찾는 것
  - 예시1: 다음에 올 숫자는 무엇일까? 1,1,2,3,5,8,13, ?
  - 예시2: 내가 학교에 등교할 때 매일 똑같은 루트의 대중 교통을 이용 (3번 버스→6호선 지하철(5정거장) →도보)

Computational Thinking

- Algorithm (알고리즘)
  - 문제를 해결하기 위해 추상화된 핵심 원리를 단계적이고 반복적인 절차로 나타내는 것
  - 알고리즘 표현 방법
    - 자연어
      - 사람들이 일상생활에서 사용하는 언어 (자연어)로 알고리즘을 표현하는 것은 복잡
    - 순서도
      - 약속된 기호와 선을 사용하여 문제 해결 과정을 표현
      - 복잡한 프로그램에서는 한계점
    - 의사코드 (pseudo code)
      - 특정 프로그래밍에 사용하는 언어와 유사한 서술로 표현
      - 의사코드를 작성하면 특정한 프로그래밍 언어로 쉽게 변환 가능
    - 프로그래밍 언어
      - Low level language : 사람이 이해하기 힘든 언어 (기계어, 어셈블리어)
      - High level language : 사람이 사용하는 단어로 이해하기 쉽게 만든 언어 (C, Java, Python, …)

Computational Thinking

- Algorithm (알고리즘)

```
시작
    X에 3, Y에 5를 대입한다.
    X 값을 Z에 대입한다.
    Y 값을 X에 대입한다.
    Z 값을 Y에 대입한다.
    X와 Y 값을 출력한다.
끝
```

자연어처리

기호	명칭	의미
	단말	순서도의 시작과 끝을 의미한다.
	흐름선	각 기호를 연결하며, 순서도의 흐름을 나타낸다.
	처리	계산 등 자료의 연산 또는 처리를 나타낸다.
	준비	변수의 초깃값, 기억 장소의 설정 등 작업의 준비 과정을 나타낸다.
	판단	조건을 판단하여 '예' 또는 '아니오'로 이동한다.
	입출력	자료의 입력과 출력을 나타낸다.
	출력	출력 장치를 통한 출력을 나타낸다.

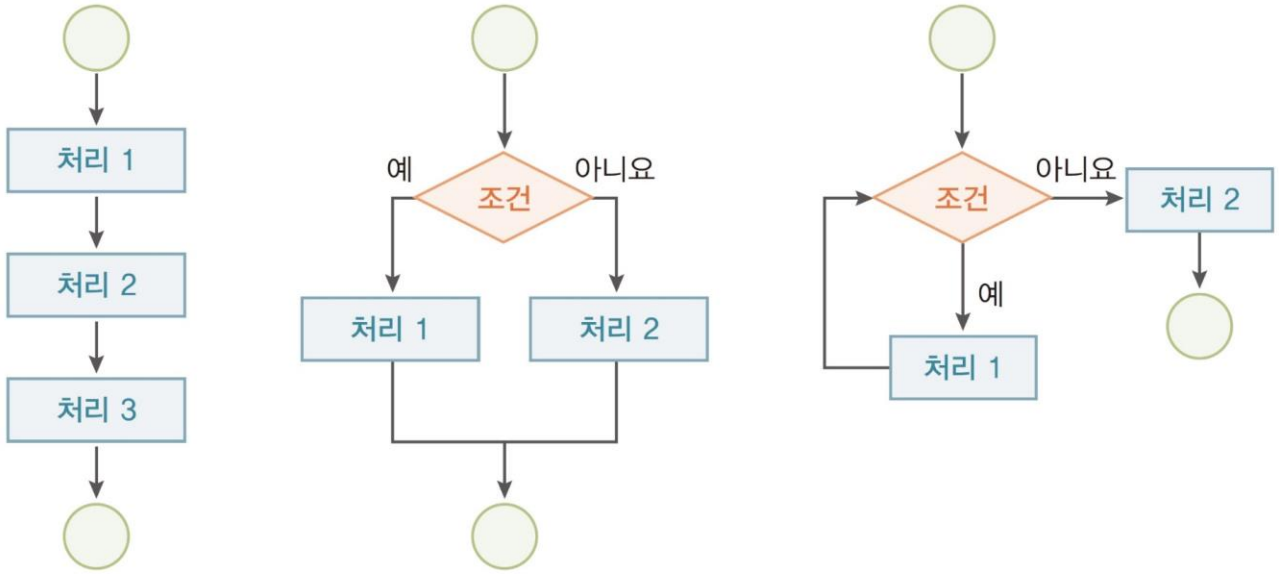
순서도

```
START
    X=3, Y=5
    Z=X
    X=Y
    Y=Z
    PRINT X, Y
END
```

의사코드

스크래치로 배우는 컴퓨팅 사고, 한빛아카데미

- Algorithm (알고리즘)
  - 알고리즘의 조건
    - 입력: 알고리즘에 입력되는 자료가 0개 이상 존재
    - 출력: 알고리즘이 실행되면 적어도 결과 값이 1개 이상 생성
    - 유한성: 알고리즘은 반드시 종료되어야 함
    - 명확성: 알고리즘의 명령이 모호하지 않고 명확해야 함
    - 수행 가능성: 알고리즘의 명령은 수행 가능해야 함



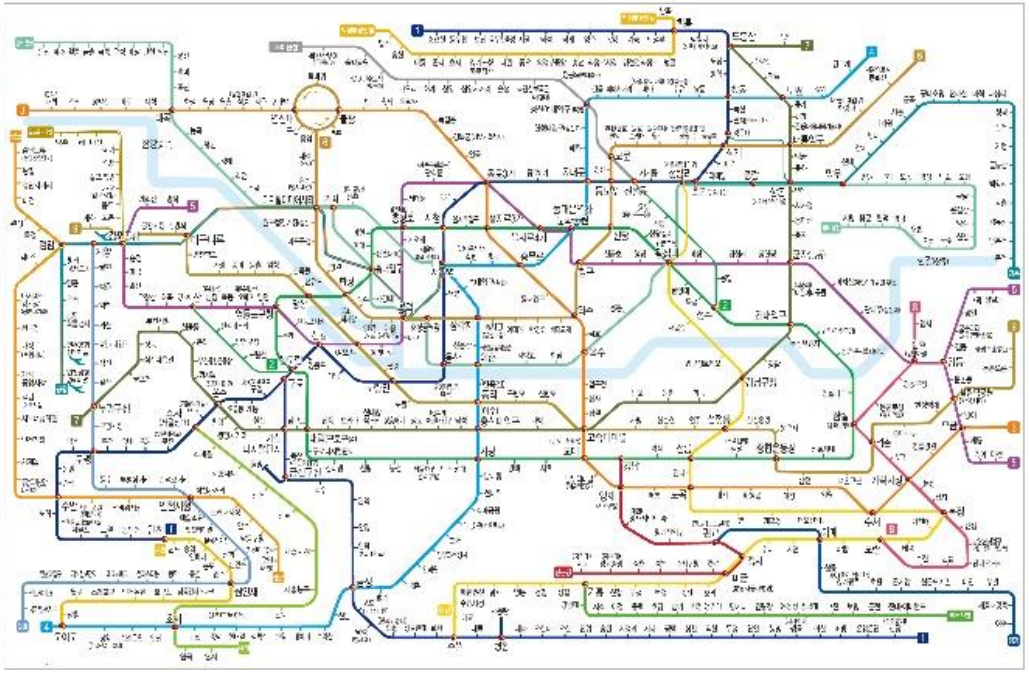
(a) 순차 구조 (b) 선택 구조 (c) 반복 구조

스크래치로 배우는 컴퓨팅 사고, 한빛아카데미

Computational Thinking

- Abstraction (추상화)
  - 복잡한 문제에서 필요하지 않은 특징이나 세부적인 사항을 없애고 핵심적인 요소만을 남겨서 일반화된 모델을 만드는 것
  - 구글맵: 모든 도로까지 다 표시되어 차로 이동하거나 도보로 이동 시 편리
  - 지하철 노선도: 구글맵은 지하철 이동을 위해 필요 없는 정보가 너무 많아 지하철 노선도가 편리

(a) 구글맵



(b) 지하철 노선도

스크래치로 배우는 컴퓨팅 사고, 한빛아카데미

Computational Thinking

- Abstraction (추상화)
  - 예시2 : 픽토그램
    - 추상화해 놓은 네온사인 (픽토그램)은 멀리서도 무엇을 팔고 있는 가게인지 쉽게 인지 가능
  - 불필요한 것을 제거함으로써 문제 본질을 더욱 정확하게 파악 가능

스크래치로 배우는 컴퓨팅 사고, 한빛아카데미

Computational Thinking

감사합니다

kimtwan21@dongduk.ac.kr

김 태 완