

# 인공지능 개요

고려대학교 석준희

*ChatGPT: Optimizing  
Language Models  
for Dialogue*

*We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible to challenge incorrect premises, and request more information. ChatGPT is a sibling model to GPT-3, but it is trained to follow an instruction to answer questions.*

# 목차

- 인공지능의 발전
- 기계학습 개요
- 프로그래밍 실습

인공지능 개요

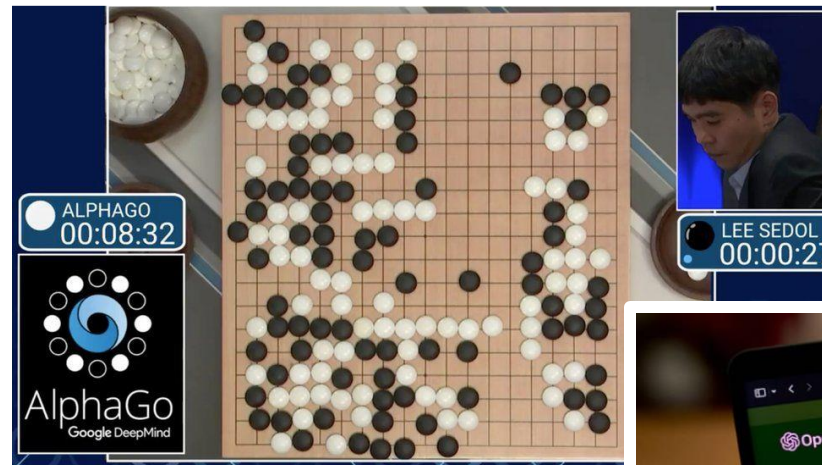
# 인공지능의 발전

# 인공지능 (AI: Artificial Intelligence)



IBM 왓슨의 퀴즈쇼 (2011)

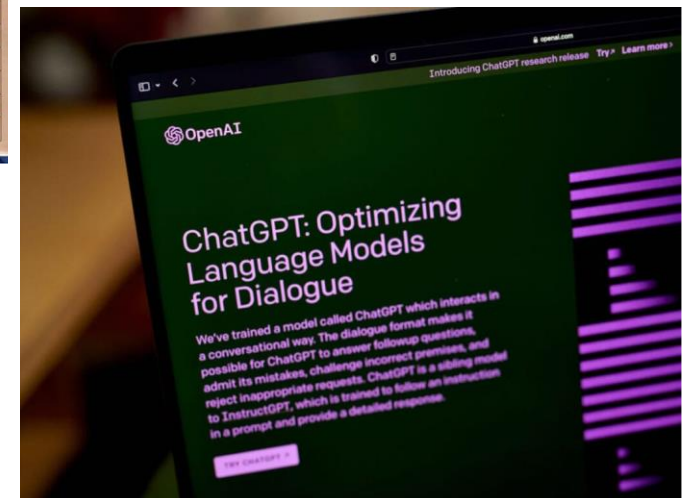
알파고와의 대국 (2016)



<https://www.cbsnews.com/news/ibm-watson-defeats-humans-in-jeopardy/>

<https://www.bbc.com/news/technology-35785875>

ChatGPT 열풍 (2023)

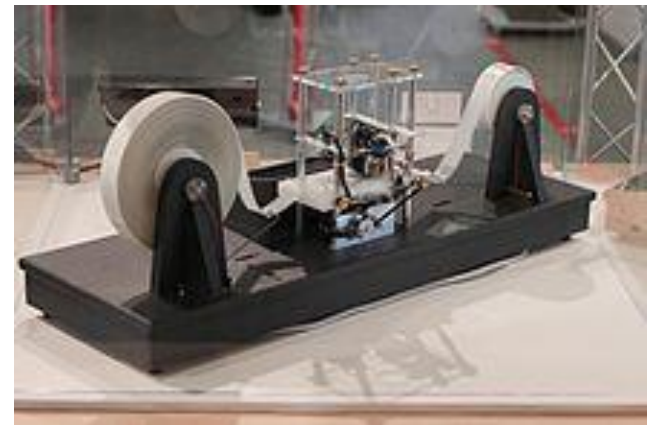
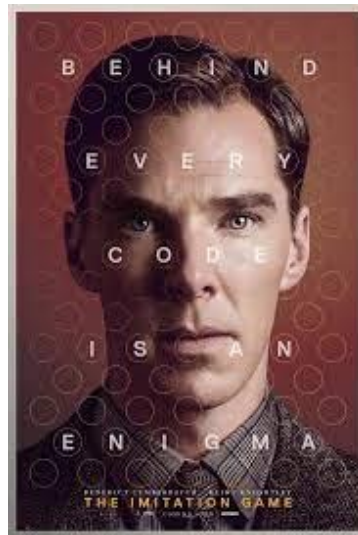


[https://chat.openai.com/chat\(ChatGPT\)](https://chat.openai.com/chat(ChatGPT))



# 인공지능의 시작

- '생각하는 기계' 라는 개념은 앨런 튜링에 의해서 처음으로 제시
  - 앨런 튜링 (Alan Turing, 1912~1954): 컴퓨터 과학의 아버지, 근대적 컴퓨터 모델 (튜링 머신)의 고안
- 계산기 (Calculator): 단순한 수치적 연산, 1640년 파스칼에 의해 발명
- 컴퓨터 (Computer): 논리적 행위의 구현, 튜링에 의해 개념 정립
- 인공지능: 인간이 하는 논리적 행위(인지, 판단 등)를 모방하는 기계

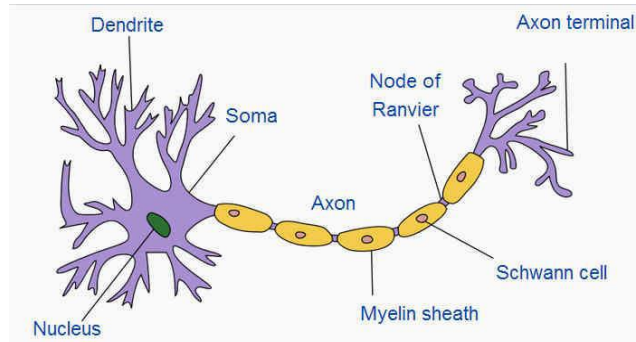
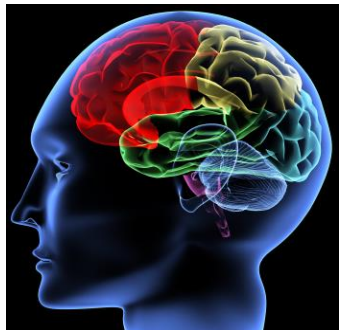




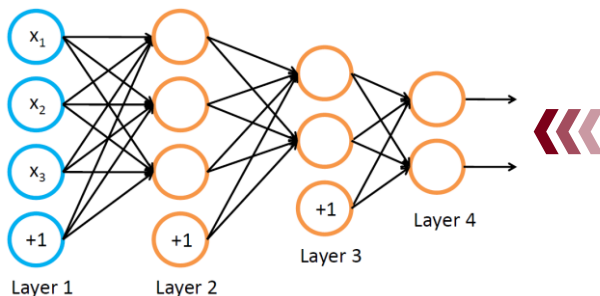
# 인공신경망 (Artificial Neural Network)

- 인공지능을 어떻게 구현해야 하는지는 명확하지 않음
  - 로켓의 궤적: 어떻게 푸는지는 알지만 계산이 어려움
  - 안면 인식: 5살 어린아이도 잘 하지만 컴퓨터로는 어려움
- 인공신경망: 인간의 두뇌를 모방하여 만든 수학적 모델

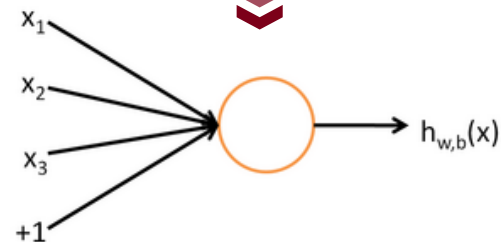
인간의  
두뇌



신경  
세포



인공신경망



퍼셉트론

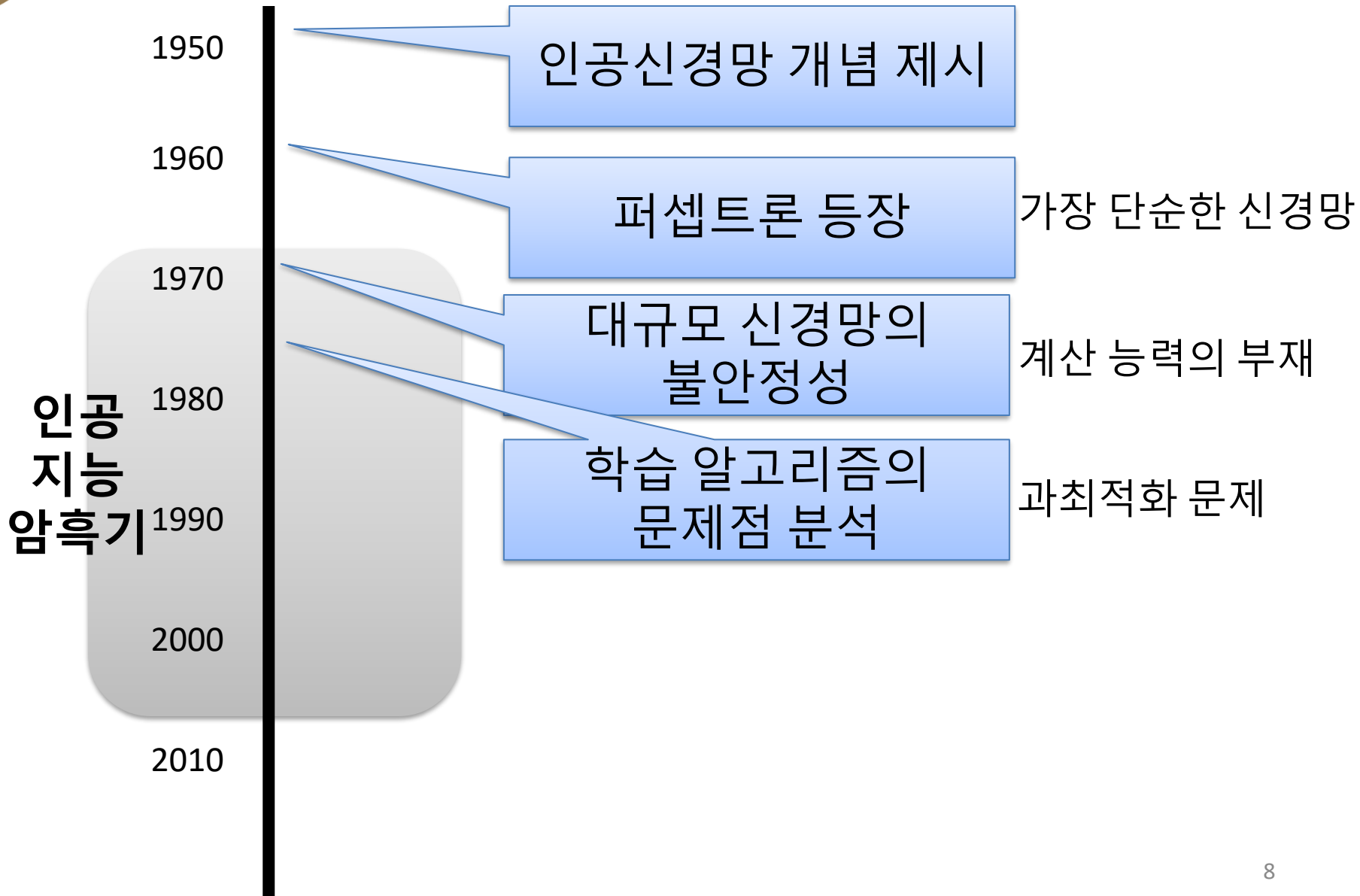


# 인공지능의 구현

- 인공지능을 어떻게 구현해야 하는지는 명확하지 않음.
  - 우리는 지능이 판단을 한다는 것은 알지만 어떻게 판단하는지는 모름
- 규칙기반 인공지능
  - 명확한 규칙이 존재하고 그 규칙에 따라서 판단
  - 다양한 소스로부터 규칙을 수집하고 정리하는 것이 주된 요구사항
  - E.g. clinical decision support system
- 데이터기반 인공지능 (머신러닝, 기계학습)
  - 대략적인 규칙은 존재하지만 명확하지는 않음
  - 많은 사례로부터 대략적인 규칙을 배우는 것이 주된 요구사항
  - E.g. face recognition
- 하나의 인공지능 서비스를 만들기 위해서는 모든 접근법이 적절히 융합되어 사용



# 인공지능의 암흑기





- 단순한 규칙기반 혹은 탐색 기반의 인공지능이 선호됨.
  - 복잡한 지능을 묘사하는 것은 불가능

[illegible]

인공지능을 이용한 증권투자  
기법」을 개발했다.

한국과학기술원(KAIST)과 중앙대학교 컴퓨터  
과학연구소가 공동으로 개발한  
투자기법개발팀이 인간 심리학  
사상을 도입한 인공지능을  
활용한 증권투자 알고리즘을  
개발했다고 밝혔다.

연구팀은 인공지능을 이용해  
「브레인인스」라는 투자지표인  
주요 증권지표인 미지수, 거래  
보합, 기술적지표 등을 합류한  
가중치 산출을 하는 시스템이다.  
연구팀은 주가변동률, 주가  
상승률, 개발재료를 활용한  
「브레인인스」라는 투자지표인  
주요 증권지표인 미지수, 거래  
보합, 기술적지표 등을 합류한  
가중치 산출을 하는 시스템이다.  
연구팀은 주가변동률, 주가  
상승률, 개발재료를 활용한  
「브레인인스」라는 투자지표인  
주요 증권지표인 미지수, 거래  
보합, 기술적지표 등을 합류한  
가중치 산출을 하는 시스템이다.

브레인인스는 다양한 정보사항  
뿐만 아니라 주가의 움직임을  
예측하는데 작용할 것으로  
보인다.

# **인공지능 「브레인인스」 개발** **럭키證券**

기술지표·개별재료등 綜合분석 有望종목추천  
情報 입력 「인덱스펀드」보다 한발 앞선 기법

의 발전 등을 이들 기술지표가  
사로 삼았던 심리를 보충정수  
시스템의 각 지수에 부여한  
신뢰도에 따라 투자여부를 판단  
하게 된다.

브레인인스의 가장 큰 특징은  
시스템을 따라하는 정보라 하면  
아니다. 시스템이 사용하는 투  
자가가 종량선정되어 이루어지는  
정보를 스스로 입입시키도록  
설계했다는 점이다. 투자자는  
시스템의 질의에 대해 다분히  
주관적인 판단이 필요한 답변  
을 제공해야 한다. 즉 「브레인  
인스」에 예상된다. 「복합적  
의 실험성이 기대된다.」

브레인인스는 가장  
은 기법으로, 기술  
지표 및 투자자의 주관적인  
재료판단을 토대로 투자유망  
종목을 제시하게 된다.  
가장 높은 점수를 AA등급  
으로 D등급의 이르기까지 모  
두 10점제의 신뢰도가 부여  
된다.

브레인인스의 개발 및 실험을  
계기로 증권투자기법개발을 위  
한 증권사간 협력이 한층 가열  
될 전망이다.

럭키證券는 「브레인인스」를 일부  
지점에 시험적으로 설치, 운용  
실험을 보이며, 점차적으로 확대  
할 계획이다. <成哲煥기자>

1990년 2월 9일 매일경제



## 정보화 시대의 도래 (1990년대)

- 컴퓨터의 발전 (무어의 법칙)
  - 계산 및 처리 속도가 급격히 증가
  - 데이터 저장 용량의 증가
  - 가격의 하락
- 인터넷의 발전
  - 전산적으로 처리가 가능한 데이터의 증가
  - 흩어져 있던 데이터의 수집과 공유가 가능
- 데이터 처리 기법의 발전
  - 대용량 데이터 저장, 관리, 분석 기법의 개발
  - 새로운 데이터 예측 기법의 발견



**Big  
Data**



# 빅데이터 (Big Data)

- 복잡한 대규모 데이터(빅데이터)에 새로운 분석 방법론과 관리도구를 적용하여 기존에는 찾지 못했던 새로운 정보를 찾을 수 있음
  - Volume: 대규모 데이터
  - Variety: 언어, 이미지, 비디오 등을 포함하는 다양한 형태
  - Velocity: 빠르게 생성, 유통, 분석, 소비됨
- 데이터 수집
  - 어떻게 대규모 데이터를 효과적으로 수집할 것인가?
- 데이터 관리
  - 대규모 데이터를 어떻게 관리하고, 저장하고, 유통할 것인가?
- 데이터 분석
  - 기존의 데이터에서는 찾지 못했던 새로운 가치를 어떻게 찾을 것인가?



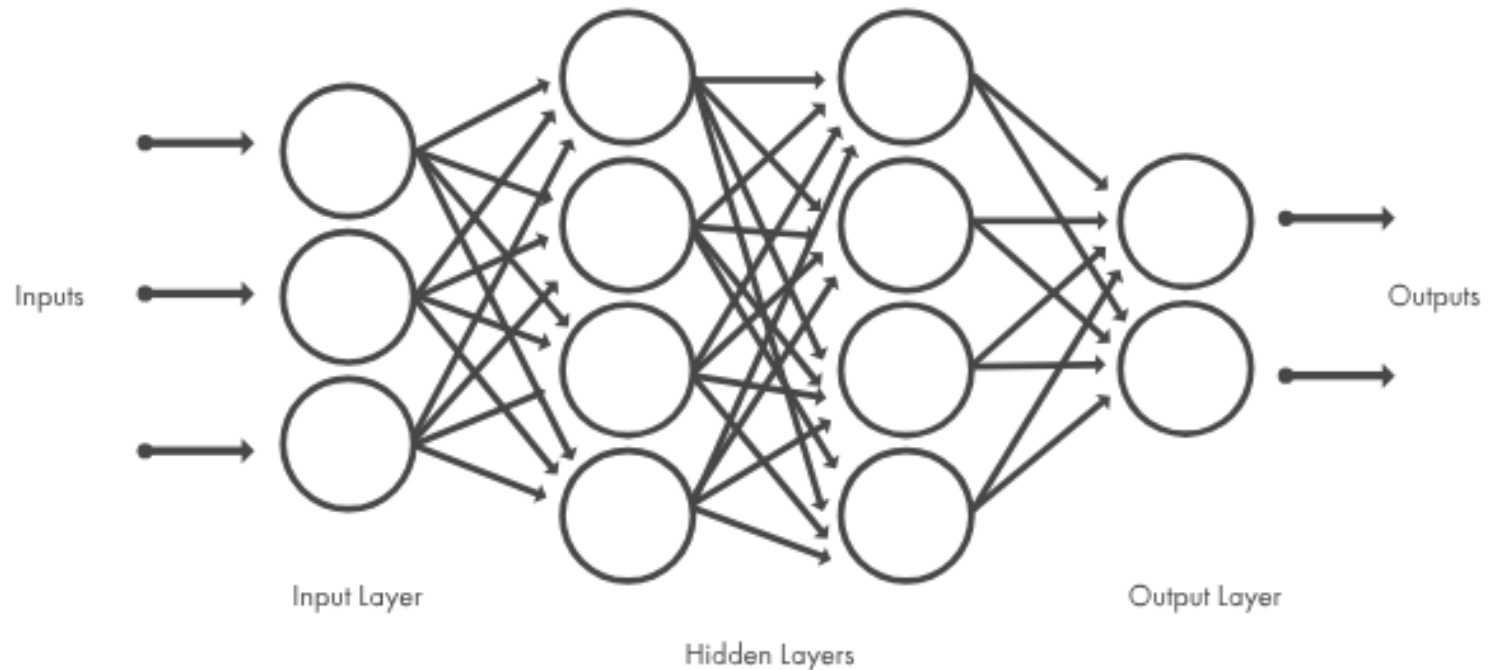
# 딥러닝(Deep Learning)의 출현





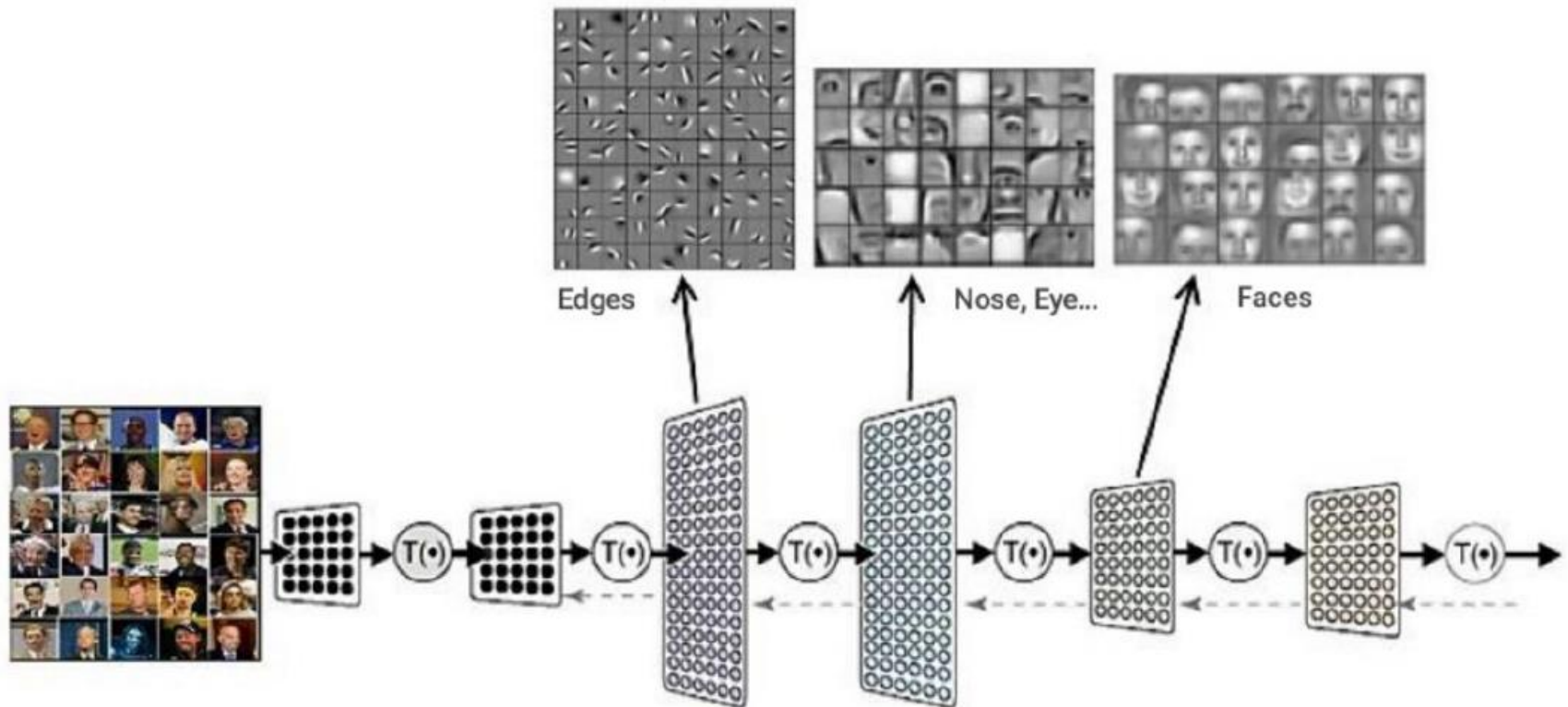
# 딥러닝 모델

- 딥러닝 모델
  - 외부로 나타나지 않고 숨겨진 계층(hidden layer)가 다수 존재하는 신경망 모델
  - 숨겨진 변수가 학습을 통해 필요한 정보(feature)를 추출하는 모델
- 전통적 접근 vs. 딥러닝 접근
  - 전통적 접근: 인간 전문가가 유용한 정보를 선정하고 이를 데이터에서 계산
  - 딥러닝 접근: 모델이 자동으로 유용한 정보를 추출하여 사용



# 딥러닝 모델

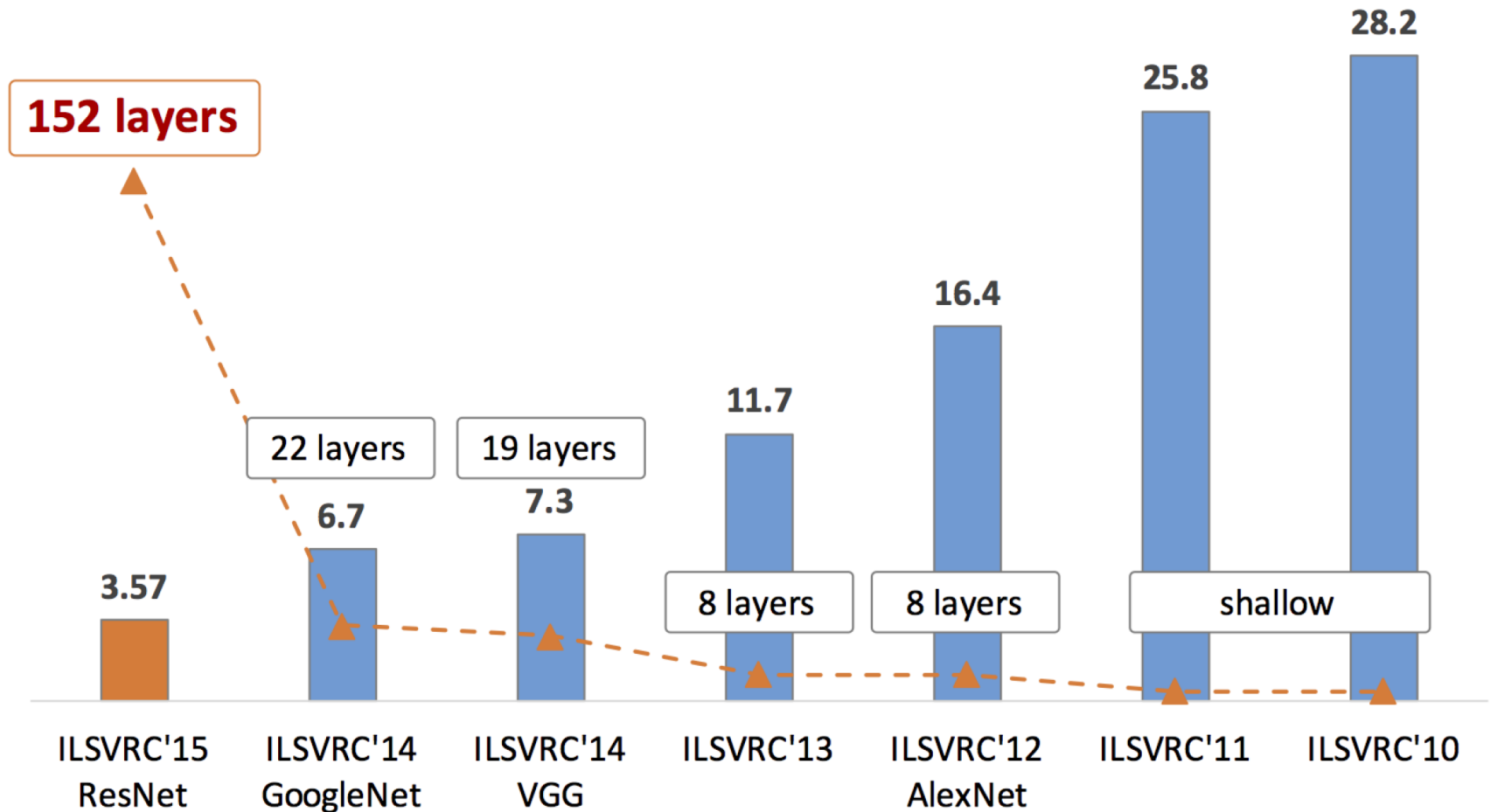
- 딥러닝을 이용한 영상 인식의 예





# 딥러닝 모델

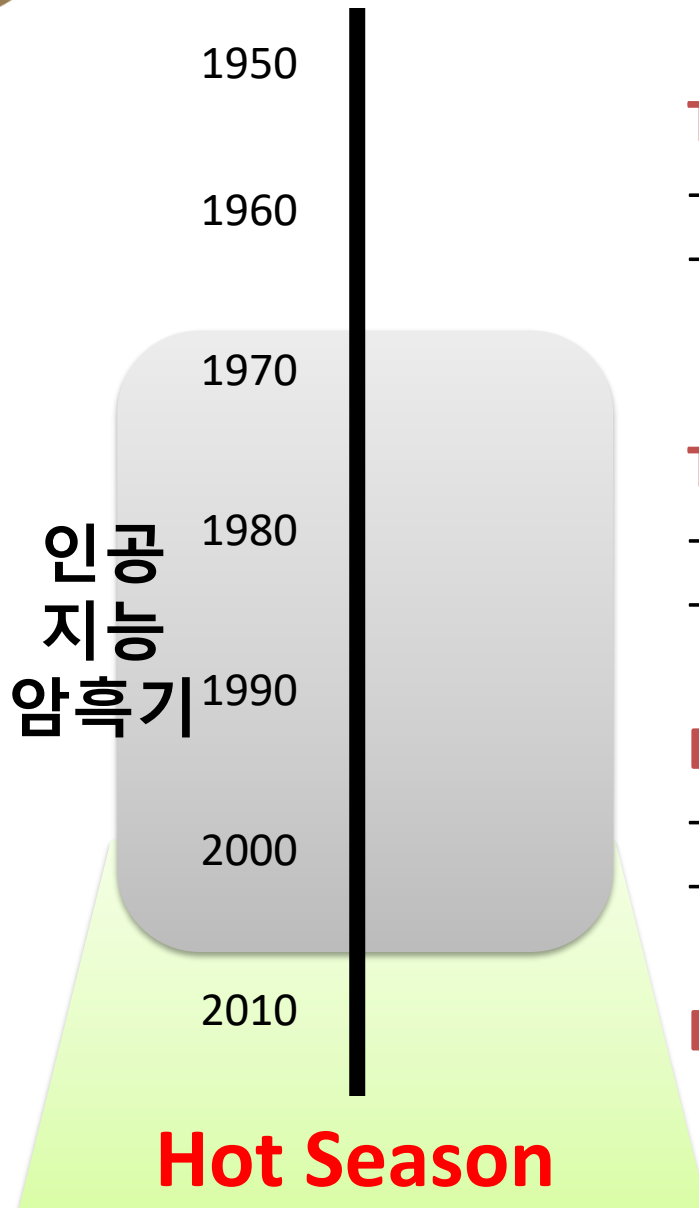
## ImageNet Large Scale Visual Recognition Challenge (ILSVRC)



Top-5 error rate, (Human : 5.1%)



# 인공지능의 봄-여름



## The hottest topic in speech recognition

- Keep breaking the previous records
- MS and Google deployed DL-based speech recognition in their products

## The hottest topic in computer vision

- Top recorder holder in competition
- Image search of Google, Baidu, and Facebook.

## Becoming hot in natural language

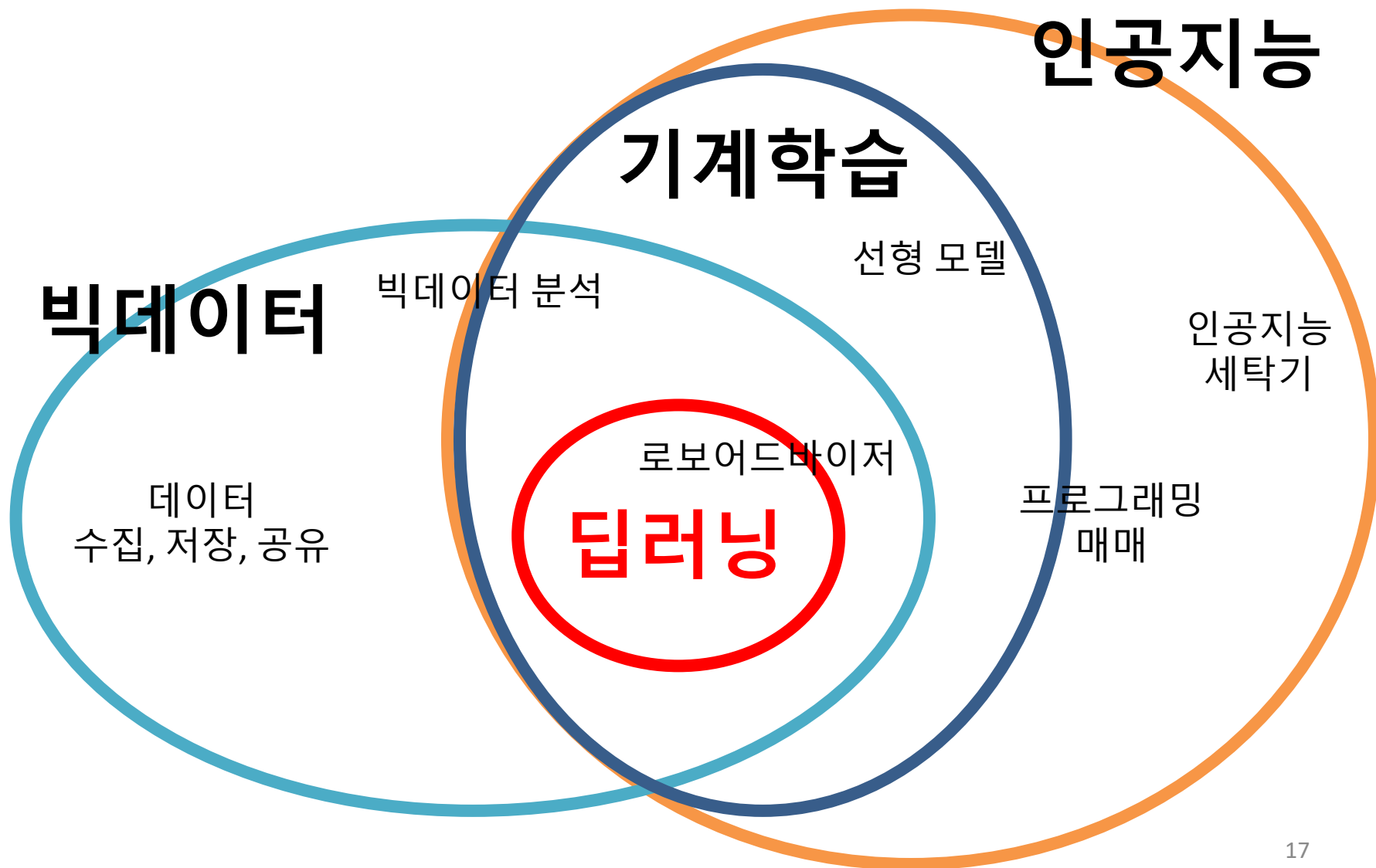
- Semantic search & deep Q&A in IBM Watson
- Large scale language model

## Becoming hot in applied mathematics





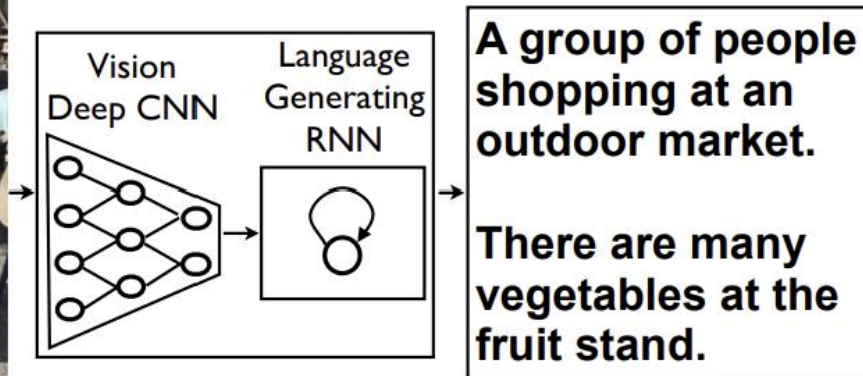
# 인공지능, 기계학습, 빅데이터, 딥러닝??





# 인공지능의 발전 방향

- 일반적인 인공지능 (AGI: Artificial General Intelligence)
  - 현재의 인공지능: 하나의 프로그램이 하나의 문제를 해결
  - 인간의 지능: 한 사람이 다양한 문제를 해결하는 것이 가능
  - AGI: 인간과 같이 한 프로그램이 다양한 문제를 해결
- **멀티모달 학습 (Multimodal Learning)**
  - 다양한 종류의 데이터를 학습하여 다양한 문제를 해결
  - E.g. Image captioning, Video Q&A, ChatGPT v4



Vinyals et al, 2014

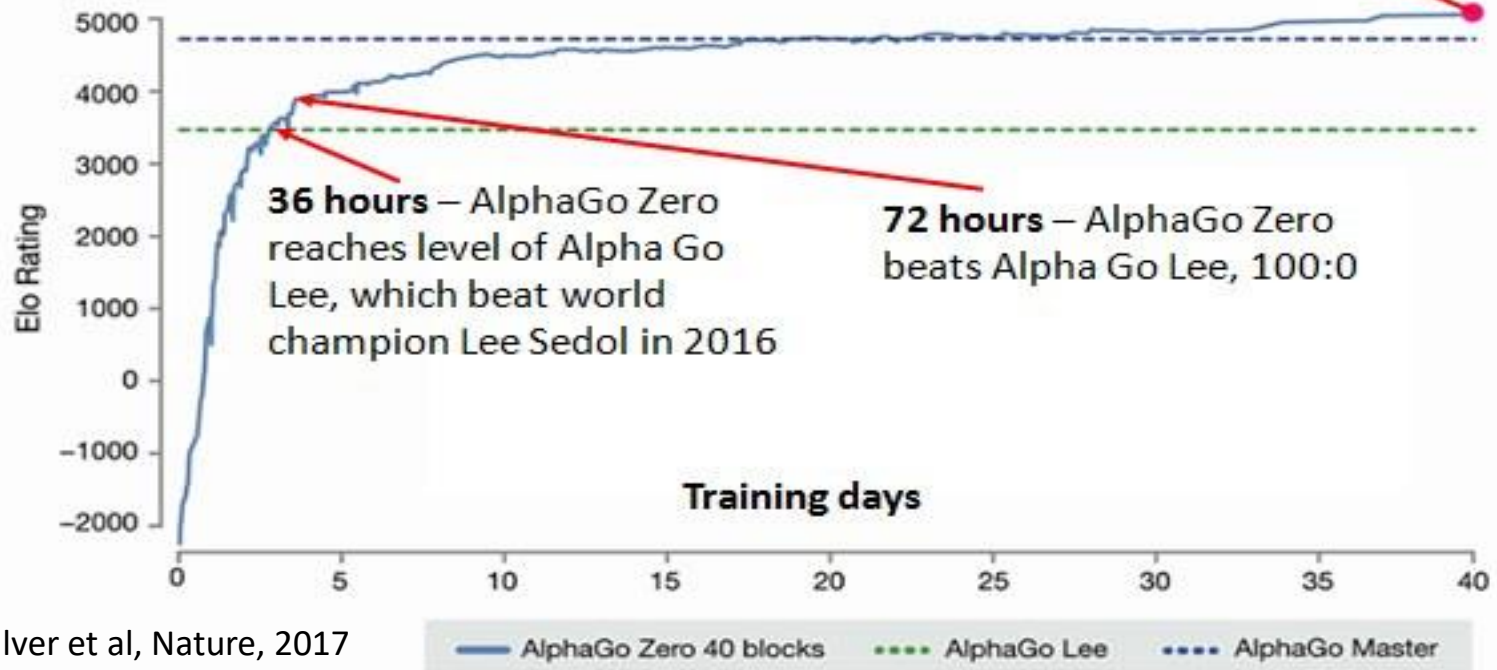


# 인공지능의 발전 방향

- 인공초지능 (ASI: Artificial Super Intelligence)
  - 기존의 인공지능: 인간의 해결법을 학습 (지도학습)
  - 미래의 인공지능: 새로운 해결법을 발견 (**강화학습**)

- AlphaGo Lee vs. AlphaGo Zero

**40 days** – AlphaGo Zero surpasses all previous versions, becomes the best Go player in the world



Silver et al, Nature, 2017

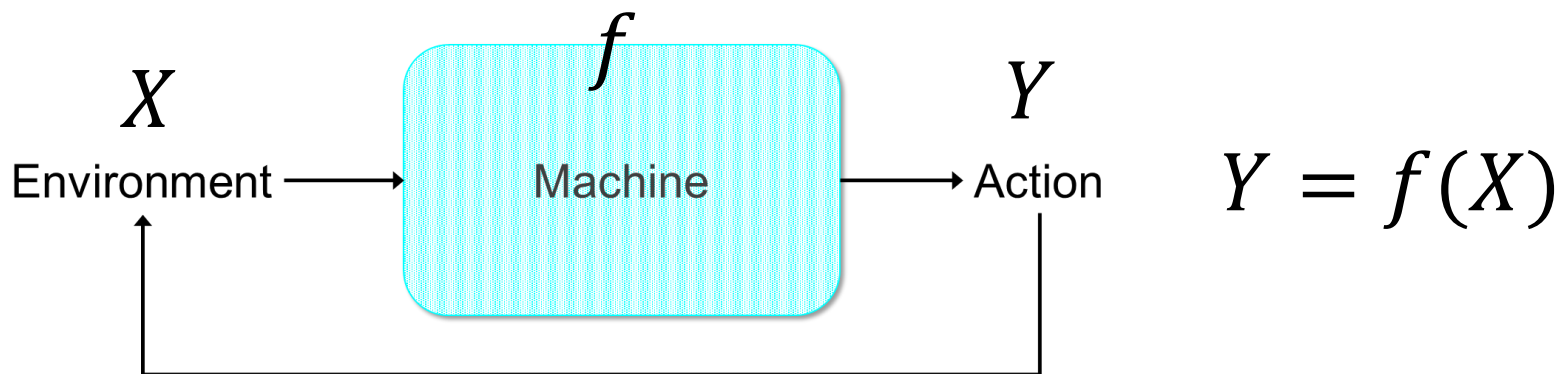
인공지능 개요

# 기계학습 개요

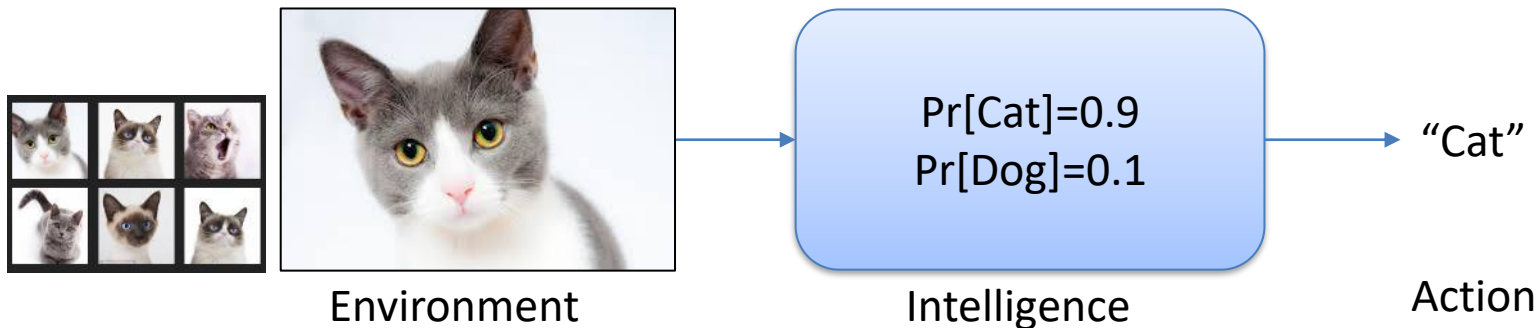


# 인공지능의 수학적 표현

- 인공지능은 인간의 지능을 모방하여 관측된 데이터를 바탕으로 어떤 행동에 대한 판단을 내리는 주체



- Environment: 어떠한 분포로부터 임의로 주어지는 데이터
- Action: 확률적 예측에 바탕을 두고 결정





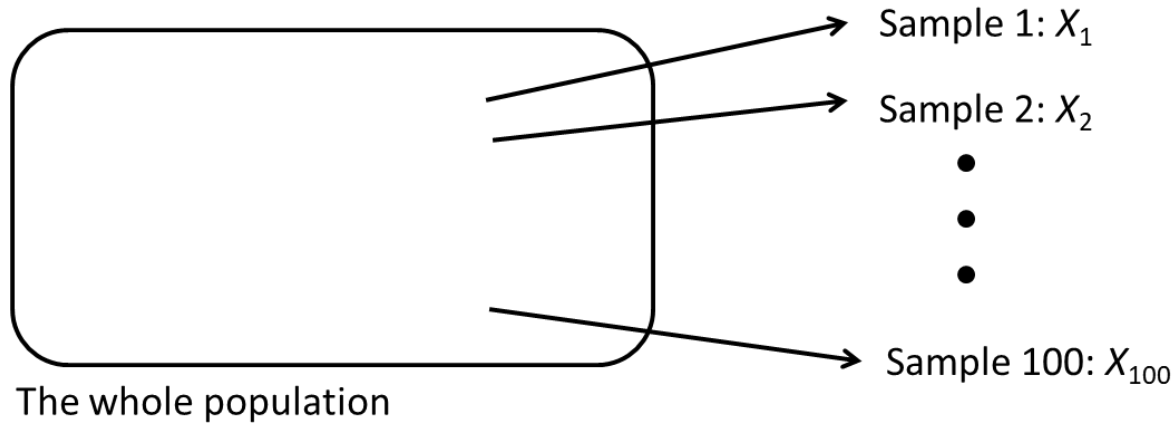
# 인공지능의 구현

- 인공지능을 어떻게 구현해야 하는지는 명확하지 않음.
  - 우리는 지능이 판단을 한다는 것은 알지만 어떻게 판단하는지는 모름
- **규칙기반 인공지능**
  - 명확한 규칙이 존재하고 그 규칙에 따라서 판단
  - 다양한 소스로부터 규칙을 수집하고 정리하는 것이 주된 요구사항
  - E.g. clinical decision support system
- **데이터기반 인공지능 (머신러닝, 기계학습)**
  - 대략적인 규칙은 존재하지만 명확하지는 않음
  - 많은 사례로부터 대략적인 규칙을 배우는 것이 주된 요구사항
  - E.g. face recognition
- 하나의 인공지능 서비스를 만들기 위해서는 모든 접근법이 적절히 융합되어 사용



# 임의의 관측 데이터

- 학습은 데이터를 바탕으로 이루어짐. 데이터란 무엇인가?



- 임의의 관측 데이터 (Observed random data)
  - 모든 관측 데이터는 모집단으로부터 임의로 추출된 데이터
  - 데이터를 다시 수집하면 기존과는 비슷한 성질을 지니는 다른 데이터가 수집됨
  - 보통 모집단은 과거부터 현재, 미래까지 발생할 수 있는 무수히 많은 샘플을 갖고 있는 집단으로 생각함
- 학습의 목표: 관측된 데이터로부터 모집단에서의 관계를 찾는 것
  - **추정(Estimation)**: 관측된 데이터로부터 모집단의 실제 값을 찾는 과정

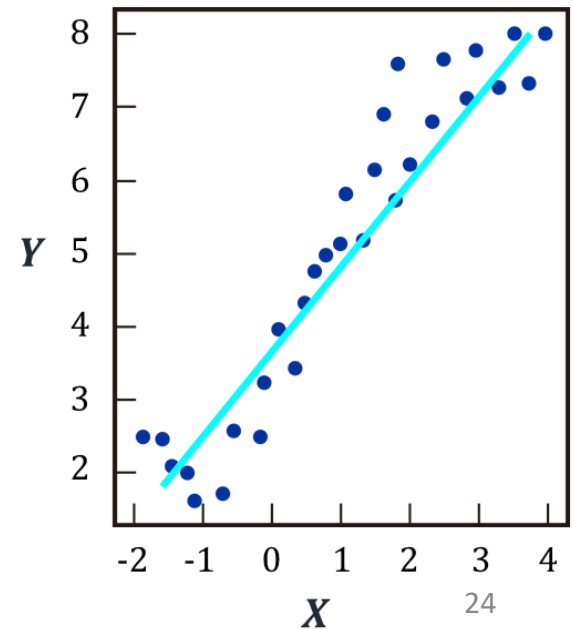
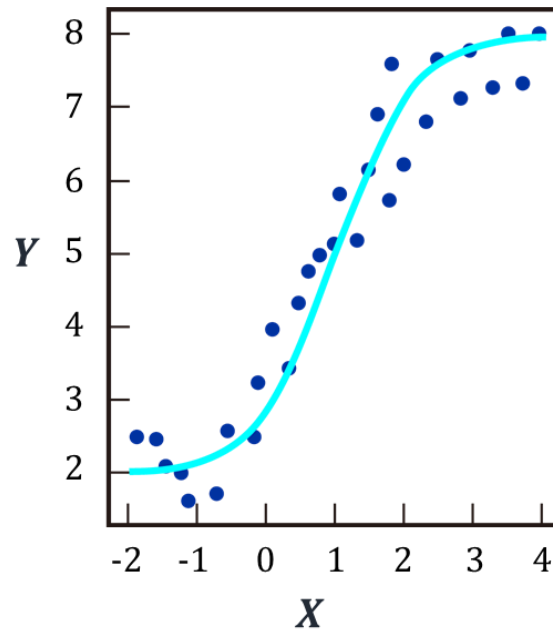
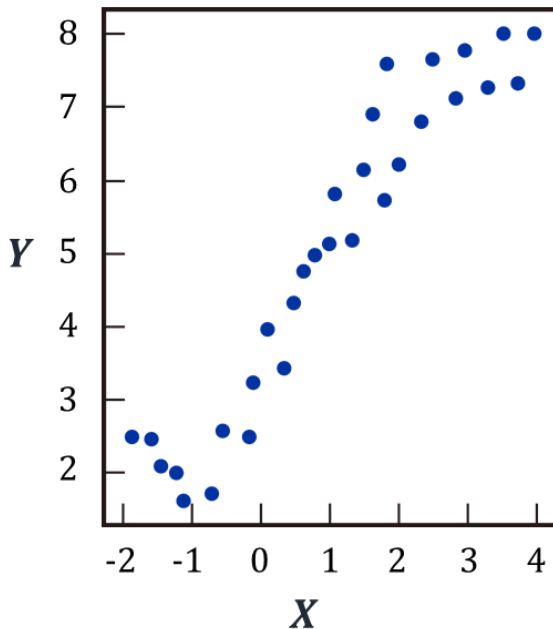


# 학습의 목표

- 출력변수  $Y$ 와 입력변수  $X = (X_1, X_2, \dots, X_p)$ 에 대하여, 둘 사이의 관계는 아래와 같이 표현 가능

$$Y = f(X) + \epsilon$$

- 함수  $f()$ 를 주어진 관측데이터로부터 추정하는 것이 학습의 목표
- 어떠한 함수가 좋은 함수인가에 대한 선택이 필요
  - 분석의 목적에 따라 달라질 수 있음: 예측 vs. 해석



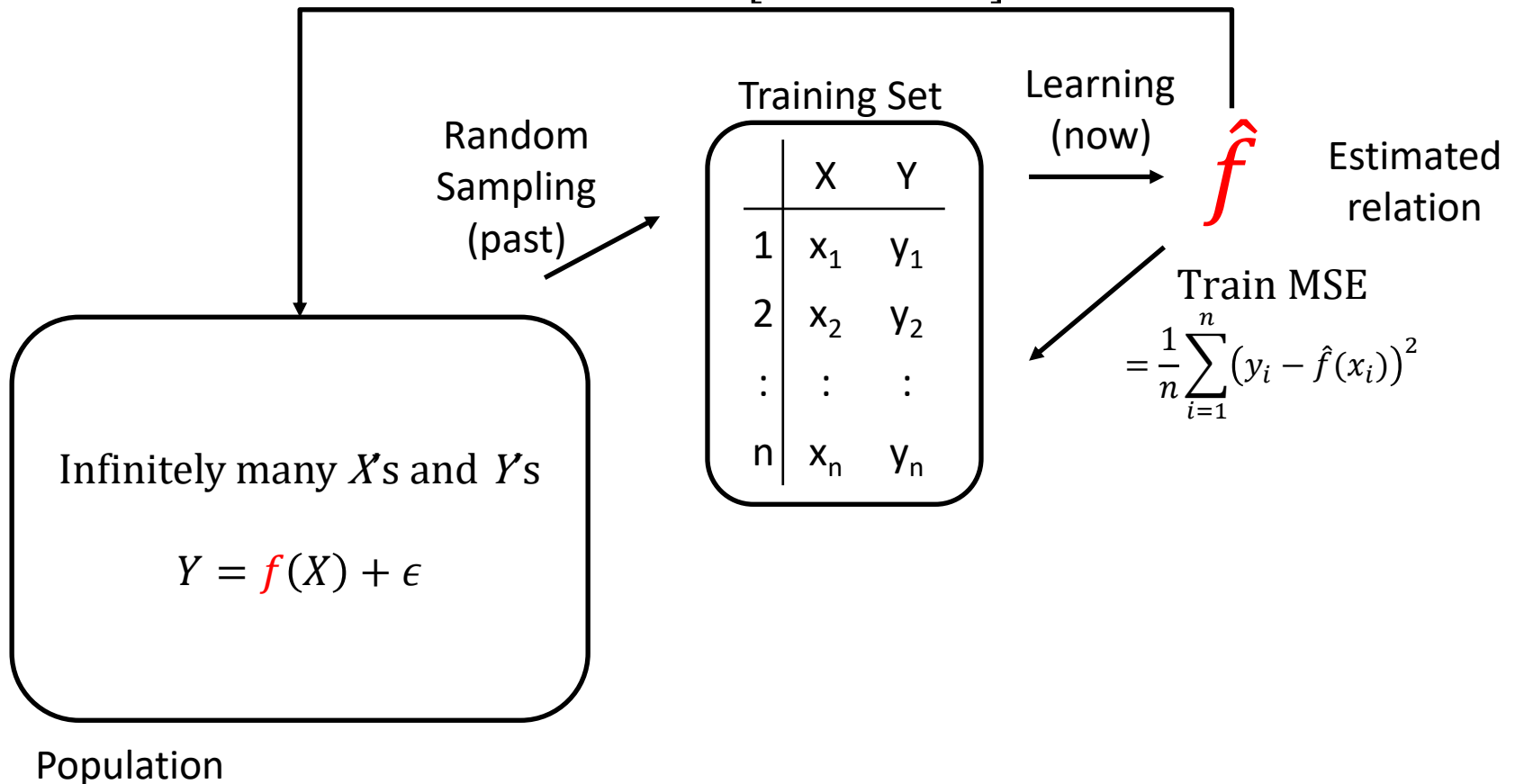




# 학습의 목표

- 학습의 문제: 모집단에서 에러를 최소화하는  $\hat{f}$  을 **훈련데이터**로부터 추정

$$\text{True Error} = E \left[ (Y - \hat{f}(X))^2 \right]$$

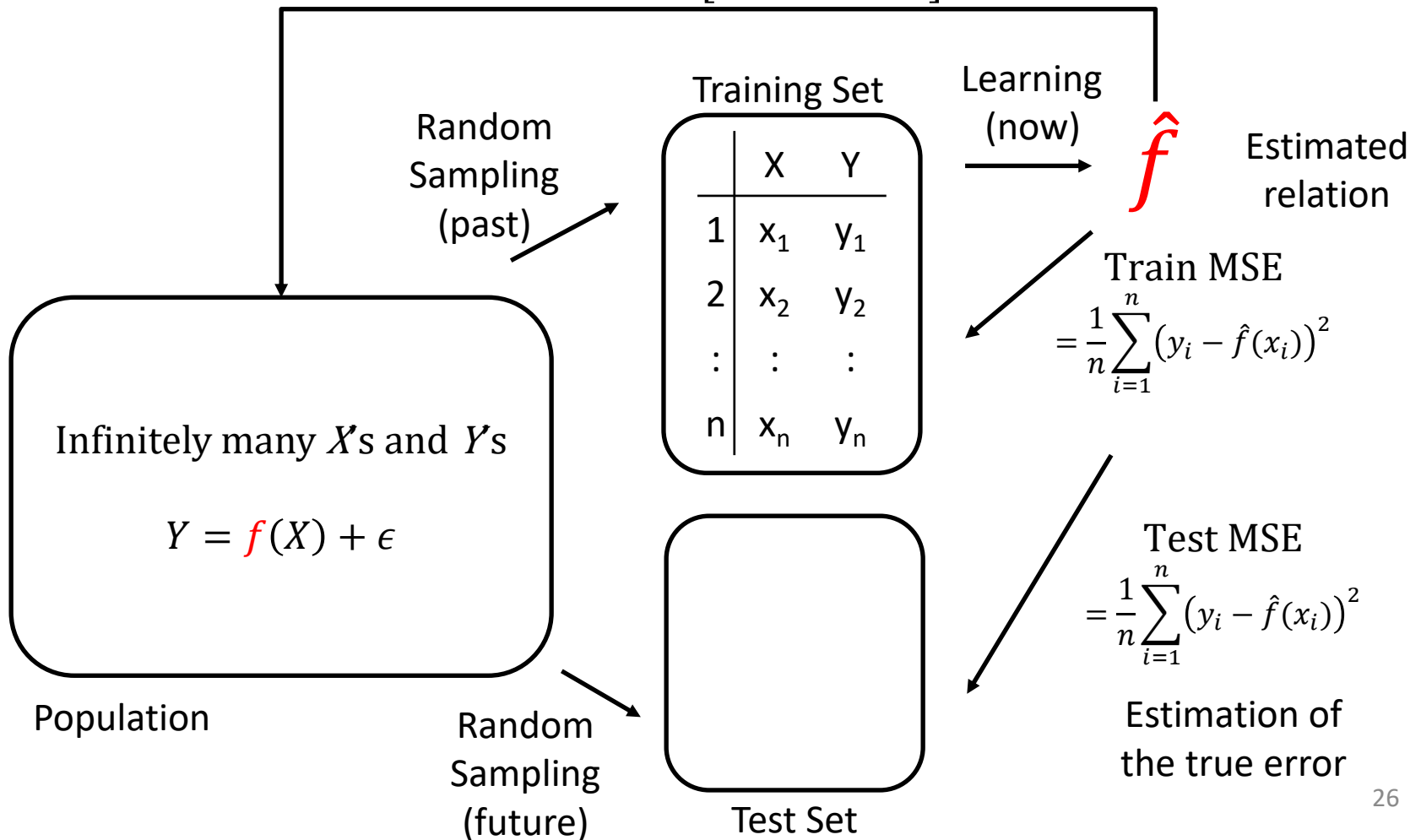




# 학습의 목표

- 학습의 문제: **평가데이터**에서 에러를 최소화하는  $\hat{f}$  을 훈련데이터로부터 추정

$$\text{True Error} = E \left[ (Y - \hat{f}(X))^2 \right]$$





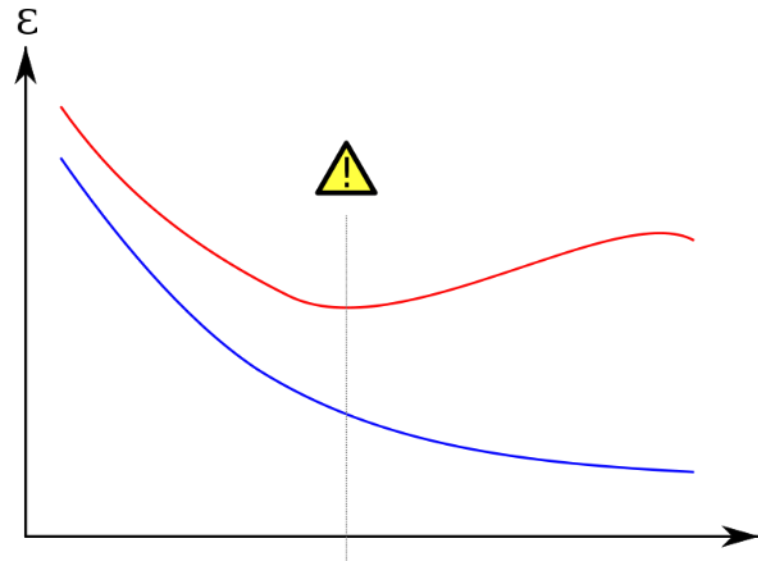
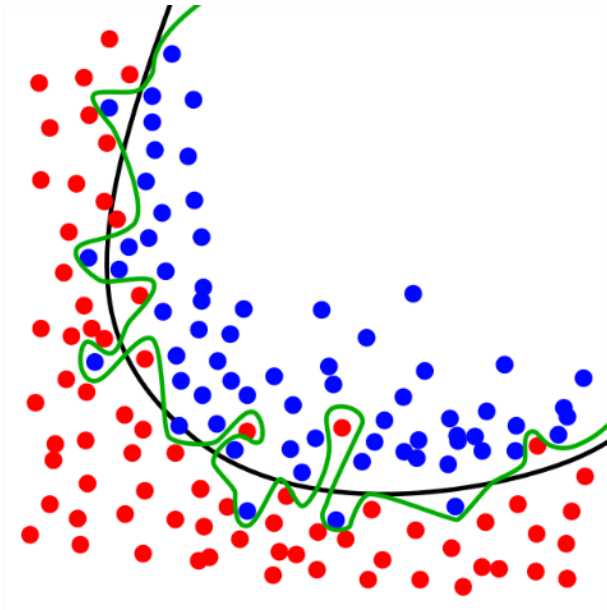
# 모델의 복잡성

- 모델 복잡성 (Model Complexity)
  - 어떤 모델이 표현한 수 있는 데이터 패턴의 범위
  - 수치적으로 잘 정의되지는 않지만 모델들끼리의 비교를 통해 상대적으로 확인 가능
    - 어느 한 모델이 다른 모델이 표현할 수 있는 모든 패턴의 데이터를 표현할 수 있으면 더 복잡한 모델
  - 유연성(flexibility), 자유도(degree of freedom) 등으로 불리기도 함
- 예시
  - $f_1(x) = ax^2 + bx + c$  vs.  $f_2(x) = ax + b$
  - 선형 회귀 모델 vs. 인공신경망
- 복잡한 데이터를 학습하기 위해서는 더 복잡한 모델이 필요
- 하지만 반드시 복잡한 모델이 좋은 것은 아님!



## 복잡성에 따른 모델의 성능

- 모델이 복잡할 수록 훈련데이터에서의 성능은 향상
- 일반적인 성능(평가데이터에서의 성능)은 그렇지 않음
  - 너무 복잡하지도 너무 단순하지도 않은 최적의 모델이 존재



<https://en.wikipedia.org/wiki/Overfitting>

- Underfitting: 모델이 너무 단순해서 훈련/평가데이터 둘 다 못 맞추는 현상
- Overfitting: 모델이 너무 복잡해서 훈련데이터는 잘 맞추지만 평가데이터는 못 맞추는 현상

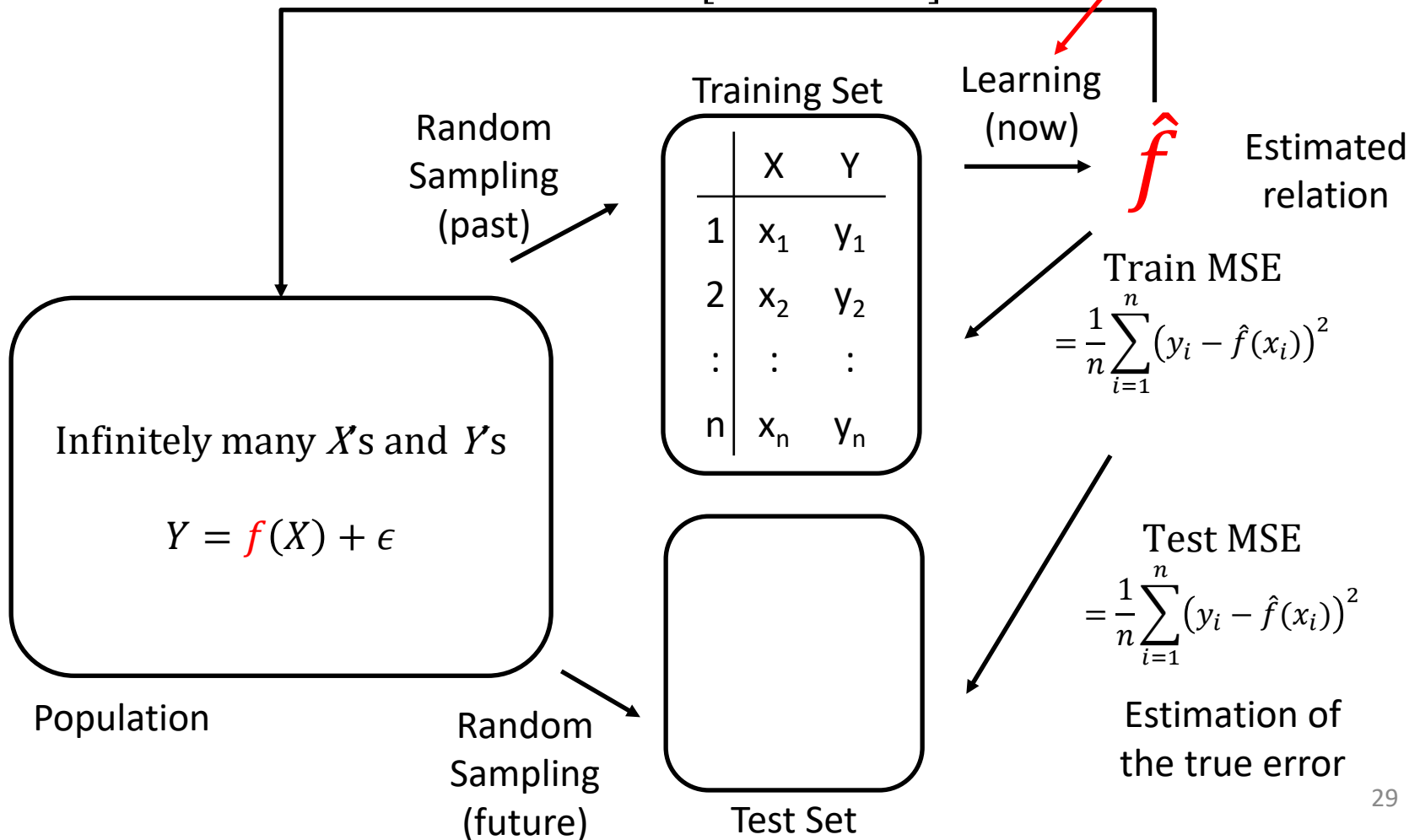


# 학습의 목표

- 학습의 문제: 평가데이터에서 에러를 최소화하는  $\hat{f}$  을 훈련데이터로부터 추정

$$\text{True Error} = E[(Y - \hat{f}(X))^2]$$

Model Selection





# 기계학습의 분류

- **지도학습 (Supervised Learning)**
  - 데이터에  $Y$ 가 주어져 있어  $Y$ 를 잘 예측/설명하는  $f$ 를 찾는 문제
  - 본 강좌 및 실무에서 주로 다룰 문제
- **비지도학습 (Unsupervised Learning)**
  - $Y$ 가 명시되지 않았거나 관심이 없고  $X$ 의 패턴 자체에 관심이 있음
  - 군집분석, 차원 축소
- **Semi-supervised Learning (Self-supervised)**
  - 지도학습과 비지도학습의 양쪽 성격을 모두 갖고 있음
  - Unlabeled data의 활용, 분류 기반 군집분석 등
- **강화학습 (Reinforcement Learning)**
  - 주어진 환경에서 최대의 보상을 달성하기 위한 정책의 학습
  - 임의 추출된 데이터가 아닌 전략적으로 선택된 데이터를 이용해 학습



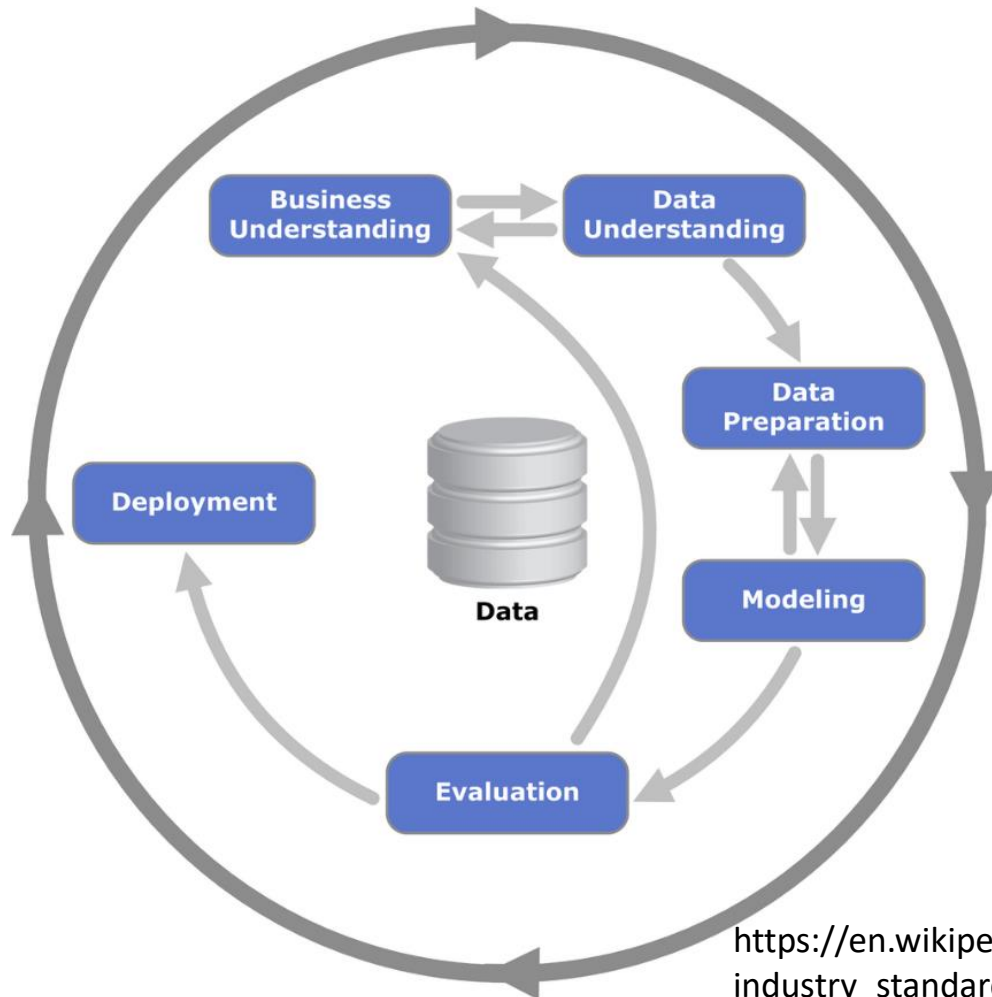
# 기계학습의 절차

- 문제의 설정 → 종속변수  $Y$ 가 무엇인가?
- 데이터 수집 → 관련 데이터 수집 ( $X, Y$ )
  - (기존분석) 실험의 설계하고 수행하여 데이터 수집
  - (빅데이터분석) 이미 존재하는 DB에서 관련된 모든 데이터를 수집
- 탐색적 데이터 분석 (EDA, Exploratory data analysis)
  - 데이터에 대해서 배우는 과정
  - 시각화, 결측치, 이상치 탐색 등을 포함
- 본격적 데이터 분석 (예측 모델)
  - 클린 데이터로부터 시작 ( $n = 100M, p = 10k$ )
  - 트레이닝 셋과 테스트 셋을 분리 (보통 시간순서에 따라)
  - 불필요한 종속변수 제거 (Feature selection)
  - 학습 모델 후보 선정 (EDA에 따라 4~5개정도 후보 선정)
  - (교차)검증 기법을 이용하여 모델 선정
  - 테스트셋을 이용하여 최종 성능 평가
- 완전히 새로운 데이터셋으로 다시 평가 (필드테스트)



# 기계학습의 절차

- CRISP-DM (Cross-industry standard process for data mining)
  - 데이터 마이닝을 위한 일반적인 절차에 대한 표준



[https://en.wikipedia.org/wiki/Cross-industry\\_standard\\_process\\_for\\_data\\_mining](https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining)



인공지능 개요

# 프로그래밍 실습

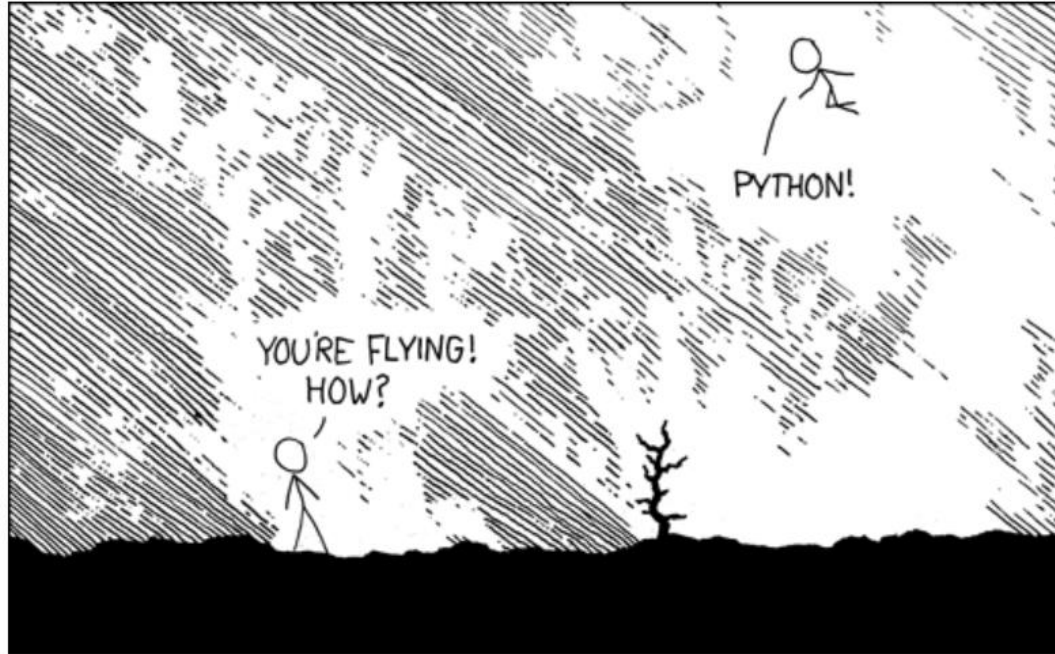


# 인공지능 프로그래밍

- 일반적인 코딩/프로그래밍
  - Java, C, C++, Javascript 등등
- R
  - 통계학자들이 개발한 통계분석 전문 언어
  - 통계 전문 언어 중에서는 어렵지만, 일반적인 프로그래밍 언어로서는 쉬움
  - 시각화와 통계분석이 장점
  - 빅데이터 분석에 많이 사용
- 파이썬 (Python)
  - 컴퓨터 학자들이 개발한 일반적인 언어
  - R보다는 어렵지만 프로그래밍 언어 중에서는 쉬움
  - 빠른 처리와 범용성이 장점
  - 인공지능에 많이 사용



## 왜 파이썬인가?

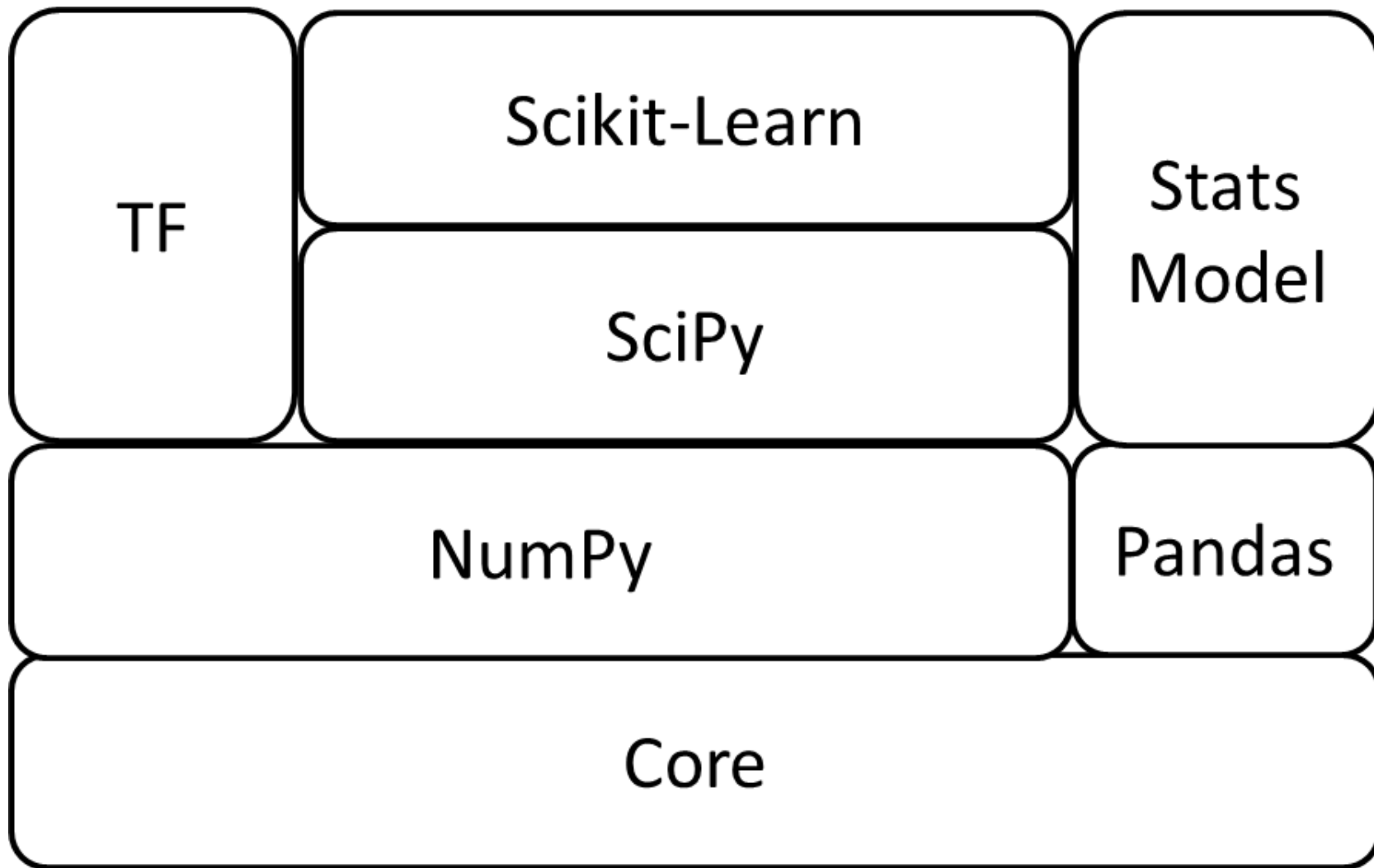


<https://xkcd.com/353/>

- 쉽고 빠르게 학습 및 이용이 가능
- 범용적 프로그래밍 언어
- 다양한 최신 프로그래밍 기법이 도입 (객체지향, 동적타이핑, ...)
- 잘 조성된 생태계



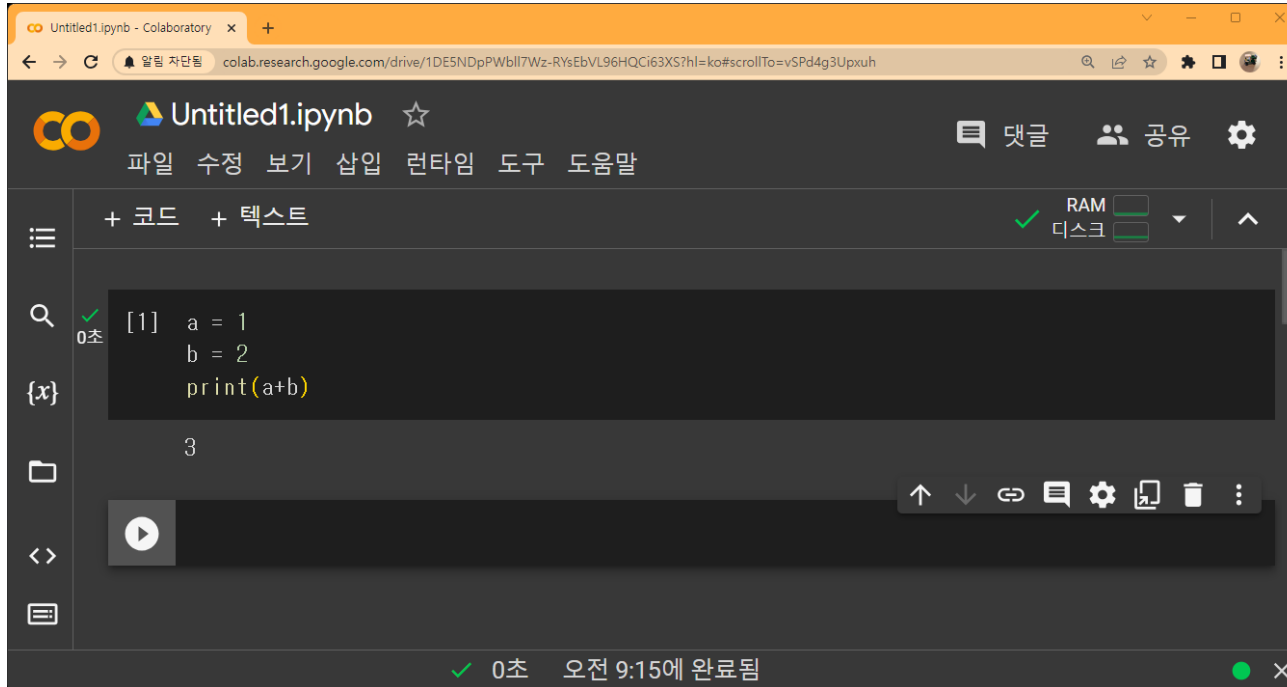
## 인공지능을 위한 파이썬 라이브러리





# 파이썬의 사용

- 컴퓨터에 직접 설치 후 사용
  - 아나콘다 패키지 설치 ( <https://www.anaconda.com/download> )
  - Jupyter notebook을 사용하여 파이썬 실행
- 온라인 클라우드 사용
  - 구글 코랩을 이용 ( <https://colab.research.google.com/?hl=ko> )
  - Jupyter notebook과 유사한 인터페이스
  - 구글 계정이 필요





# 데이터 타입

## 숫자형 데이터

```
[ ] a = 1
    b = 2
    c = a+b
    print(a, '+', b, '=', c) # 출력함수
```

1 + 2 = 3

```
[ ] # 사칙연산: 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지
    a = 5
    b = 3
    print(a+b, a-b, a*b, a/b, a%b)
```

8 2 15 1.6666666666666667 2



# 데이터 타입

## 문자열 데이터

```
▶ a = 'Hello' # 문자열은 ' ' 혹은 ""로 표현
  print(a, len(a)) # len() 문자열의 길이
  # 문자열에서 각 문자 참조
  # 순서대로 0, 1, 2, ...; 뒤에서부터 -1, -2, -3 ...
  print(a[0], a[1], a[-1])
```

```
☞ Hello 5
   H e o
```

```
[ ] a = 'Hello'
    b = 'World'
    print(a+b) # 문자열 합치기
```

HelloWorld



# 데이터 타입

## 리스트 데이터

```
▶ # 리스트: 여러개의 값을 갖고 있는 데이터 타입  
# 각 원소는 0, 1, 2... 로 참조 (인덱싱)  
# 뒤에서부터는 -1, -2 ...  
a = [1, -3.1, 'hello', [5, 'b']]  
print(a[0])  
print(a[1])  
print(a[3], a[3][0])
```

```
☞ 1  
-3.1  
[5, 'b'] 5
```

```
[ ] a.append('added') # 새로운 값의 추가  
print(a)
```

```
[1, -3.1, 'hello', [5, 'b'], 'added']
```





# 데이터 타입

## 딕셔너리 데이터

```
[ ] # 딕셔너리: 키-값 의 쌍을 갖는 데이터 타입  
# 변수[키]로 값을 참조  
a = {'one':1, 'two':2, 3:'three', 4:4}  
print(a['one'],a['two'],a[3],a[4])
```

1 2 three 4

```
▶ a['five'] = 'fivefive' # 새로운 키로 새로운 값을 추가  
print(a)
```

```
☞ {'one': 1, 'two': 2, 3: 'three', 4: 4, 'five': 'fivefive'}
```



# 조건문과 반복문

## 조건문

```
[ ] a = 10
    if a < 5:
        print(a, 'is less than 5')
    elif a < 10:
        print(a, 'is larger or equal to 5 and less than 10')
    else:
        print(a, 'is larger or equal to 10')
```

10 is larger or equal to 10

```
[ ] b = 'hello'
    if b == 'world':
        print('equal')
    else:
        print('not equal')
```

not equal



# 조건문과 반복문

## 반복문

```
[ ] A = [1, 'two', [3,4]]  
    for a in A:  
        print(a)
```

1  
two  
[3, 4]

```
▶ s = 0  
  for i in range(10):  
      s = s+i  
  print(s)
```

☞ 45

```
[ ] s = 0  
    for i in range(10,100):  
        if i%3 == 0:  
            s = s+i  
    print(s)
```

1665

**감사합니다**