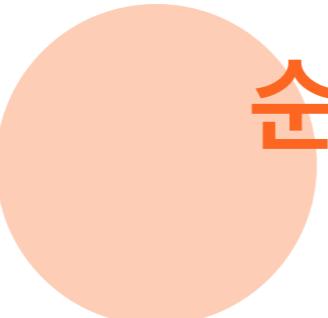




# 인공지능 생태계와 오픈 이노베이션

2017.10.12. 안대진



순서

1. 오픈소스
2. 인공지능

## 작은 서비스의 가치

“요즘은 내가 1-2가지 일을 아주 잘 알고 있다는 게 핵심은 아니다”

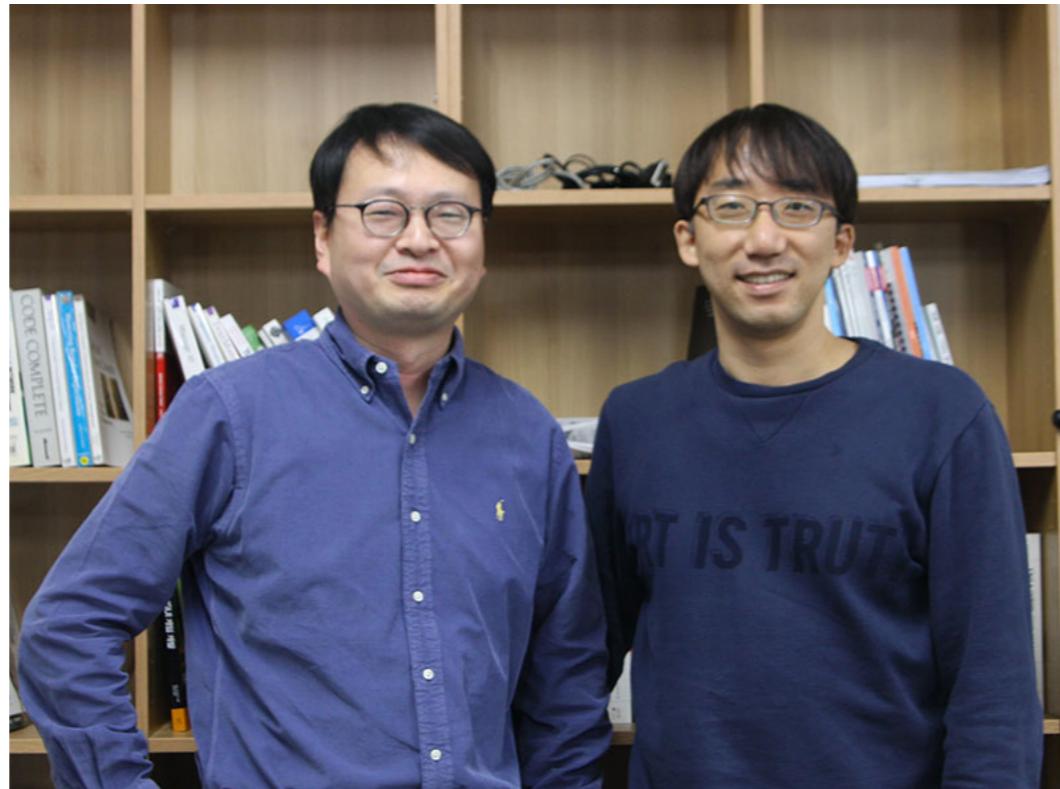
“어떤 문제든 적합한 기술을 찾아내 해결할 수 있는 능력이 중요하다”

“최근에 사람들이 필요로 하는 것들이 작은 서비스로 나오고 그것이 합쳐져서 가치가 나오고 있다”

## 오픈소스의 가치

“요즘은 내가 혼자 방에 앉아서 모든 것을 다 만드는 시대가 아니다.”

“이미 좋은 오픈소스가 공개돼 있는데, 그걸 다시 만들겠다는 것은 시간낭비다”



임성수(왼쪽), 이확영 그렙 공동대표

## 자유소프트웨어 (Free Software)

**정의 :** 최종 사용자가 소프트웨어를 사용하고, 연구하고, 공유하고, 수정할 자유를 보장하는 소프트웨어

자유 소프트웨어의 자유(free)는 공짜(gratis)가 아닌 '구속되지 않는다'는 관점의 자유를 의미한다  
‘공짜 맥주’가 아닌 ‘표현의 자유’에 가깝다 (리처드 스탈만)



*Think 'free speech'  
not 'free beer'*

*- Richard Stallman -*



# OSS - 개념

## 오픈소스 소프트웨어 (Open Source Software : OSS)

정의 : 소스코드가 공개되어, 누구나 자유롭게 사용, 복제, 배포, 수정할 수 있는 소프트웨어

“좋은 프로그래머는 어떤 프로그램을 만들어야 할지 안다.

위대한 프로그래머는 어떤 프로그램을 재사용해야 할지 안다” (에릭 레이먼드)



“Good programmers know what to write.  
Great ones know what to rewrite (and reuse).”

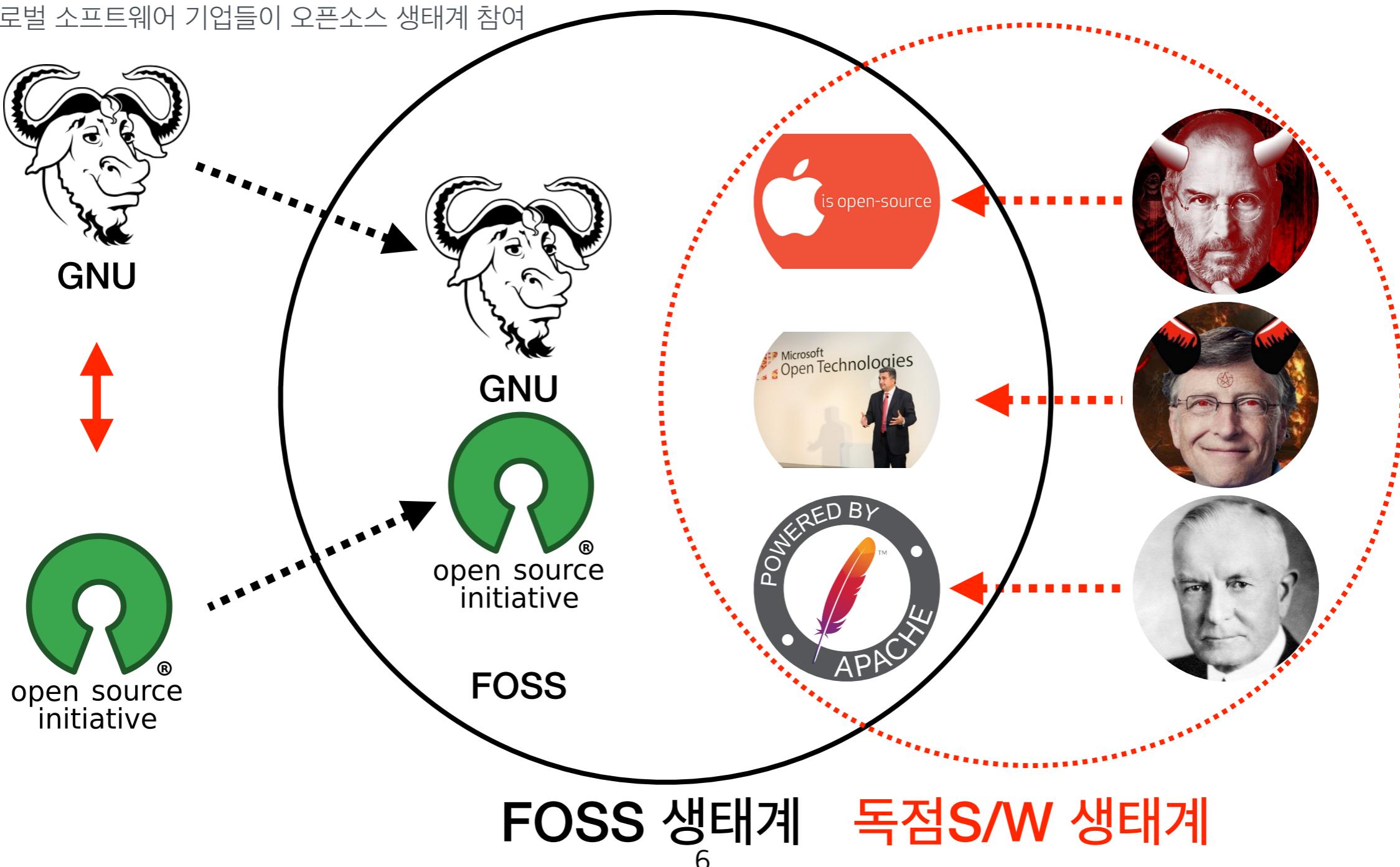
- Eric S. Raymond -



## 자유-오픈소스 소프트웨어 (Free & Open Source Software : FOSS)

정의 : 자유소프트웨어와 오픈소스의 성질을 둘 다 갖춘 소프트웨어

자유소프트웨어와 오픈소스 진영이 대립에서 협력으로 변화,  
글로벌 소프트웨어 기업들이 오픈소스 생태계 참여



# OSS - 개념

---

## 오픈소스 소프트웨어 (**OSS: Open Source Software**)

그러나 '자유(Free)'라는 단어가 '무료'로 인식되고, 엄격한 GPL조항(GNU Public License) 때문에 대다수 기업들이 자유SW운동 참여를 꺼려하자 소스코드 공개에 보다 많은 참여를 이끌어내기 위하여 에릭 레이먼드, 브루스 페런스 등은 1998년 OSI(Open Source Initiative)를 결성하고 '오픈소스(Open Source)'라는 새로운 용어를 제안했다.

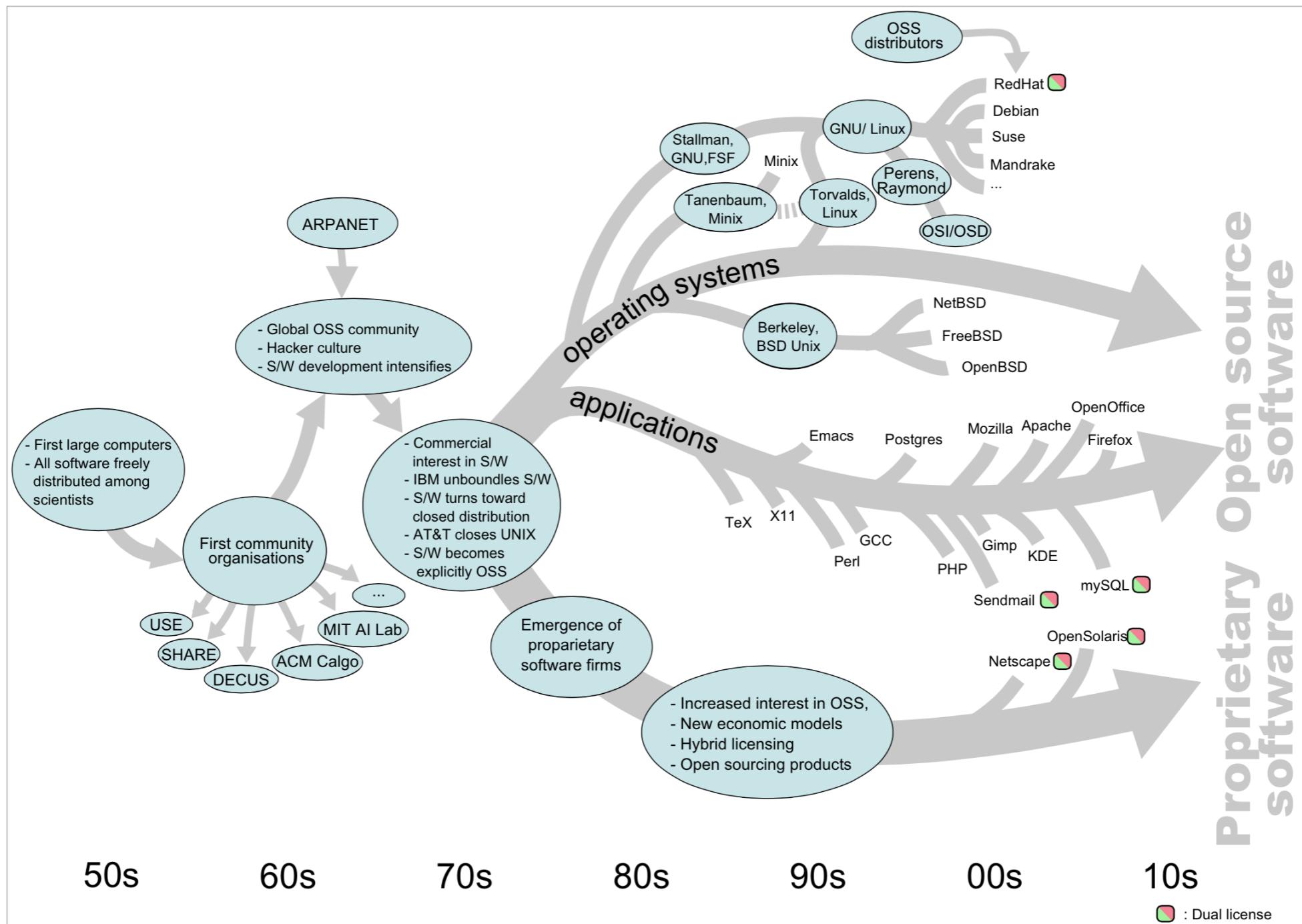
OSI는 오픈소스에 해당하는 라이선스의 최소한의 기준을 정의 (Open Source Definition, OSD)해놓고 이 정의에 따라 인증, 관리 및 촉진시키는 일을 한다.

## 자유소프트웨어 (**Free Software**)

1980년대부터 소프트웨어가 거대 부가가치 산업으로 발전하자, 지적재산권 및 라이선스 계약을 통하여 소프트웨어의 사용, 복제, 배포, 수정에 제한을 가하려는 움직임이 나타났다. 이런 움직임에 반대하여 리처드 스톤만은 1985년 GNU선언문을 발표하고 FSF를 설립하여 자유소프트웨어(Free Software) 운동을 전개하였다.

# OSS - 히스토리

60년대 코드공유문화, 70년대 유닉스(UNIX) 탄생, 80년대 상용 소프트웨어, 1984~2000 자유소프트웨어 운동, 91년 리눅스(Linux) 등장, 98년 OSI 설립, 2000년대 OSS가 주류로(구글, 아마존 등)



# 1960

## 해커 정신 - 코드 공유 문화

1960년대 미국 MIT학생들은 ‘해커 윤리강령’을 만들었다. 스티브 잡스와 빌 게이츠도 해커였다.



**제1항** : 컴퓨터에 대한 접근은 완전한 자유를 보장받아야 한다!

**제2항** : 모든 정보는 개방되어야 하고 공유되어야 한다!

**제3항** : 권력에 대한 불신-분권화를 촉진하라!

**제4항** : 실력과 열정이 해커를 평가하는 기준이다.

**제5항** : 컴퓨터를 통해 예술과 아름다움을 창조할 수 있다!

**제6항** : 컴퓨터는 모든 생활을 보다 나은 방향으로 변화시켜 줄 수 있다.

출처 : <http://www.hackerschool.org/>, <http://opensourceforu.efytimes.com/2012/02/not-crackers-but-hackers-open-source-revolution/>

# 1970

## 유닉스(UNIX) 탄생 - 개인사용자를 위한 운영체계

유닉스가 만들어낸 ‘개방된 프로그램 개발 환경’은 오픈소스의 탄생으로 이어졌다



벨 연구소의 Kenneth Thompson과 Dennis Ritchie

# 1980~1990 소프트웨어의 상업화

“당신들은 소프트웨어를 훔친 것입니다. (altair Basic)

아무도 소프트웨어를 만들 사람이 보수를 받는지는 신경 안쓰는 것입니까?

- 1976년 2월 3일. Bill Gates. 컴퓨터 애호가들에게 보내는 공개 편지



-2-

February 3, 1976

An Open Letter to Hobbyists

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds \$40,000.

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however. 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent of Altair BASIC worth less than \$2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?

Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.

What about the guys who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.

I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.

*Bill Gates*  
Bill Gates  
General Partner, Micro-Soft

1976년 2월 3일

컴퓨터 애호가들에게 보내는 공개 편지

Altair BASIC 개발에 4만달러가 들었습니다.

여러분들은 소프트웨어를 훔친 것입니다.

비용을 지불하십시오. 10명의 프로그래머를 고용해서 좋은 소프트웨어를 만들겠습니다.

- Bill Gates

**1984****GNU - 자유소프트웨어 운동의 시작**

“ 요리법을 공유하는 것이 요리의 역사만큼 오래된 것처럼,  
소프트웨어를 공유하는 것은 컴퓨터의 역사만큼이나 오래된 것이었다 ”

-1985. Richard Stallman



# 1990

## OSI - 오픈소스 운동의 시작

“좋은 프로그래머는 어떤 프로그램을 만들어야 할지 안다.  
위대한 프로그래머는 어떤 프로그램을 재사용해야 할지 안다”

-1997. Eric Raymond & Bruce Perens



# 1991

### Linux 등장 - 오픈소스 시대의 개막

“저는 386 컴퓨터에서 작동하는 (공개, free) 운영체제를 만들고 있습니다”

“이건 단지 취미입니다”

- 1991. Linus Torvalds



# 2000

## 소프트웨어는 공공재인가?

“컴퓨터와 인터넷은 100% 공공 자금으로 공공이 주창해 개발한 것이다.  
그리고 이제 막 빌 게이츠같은 자들에게 넘어가고 있다”

- 1998. Noam Chomsky 코프워치 인터뷰



# 2017

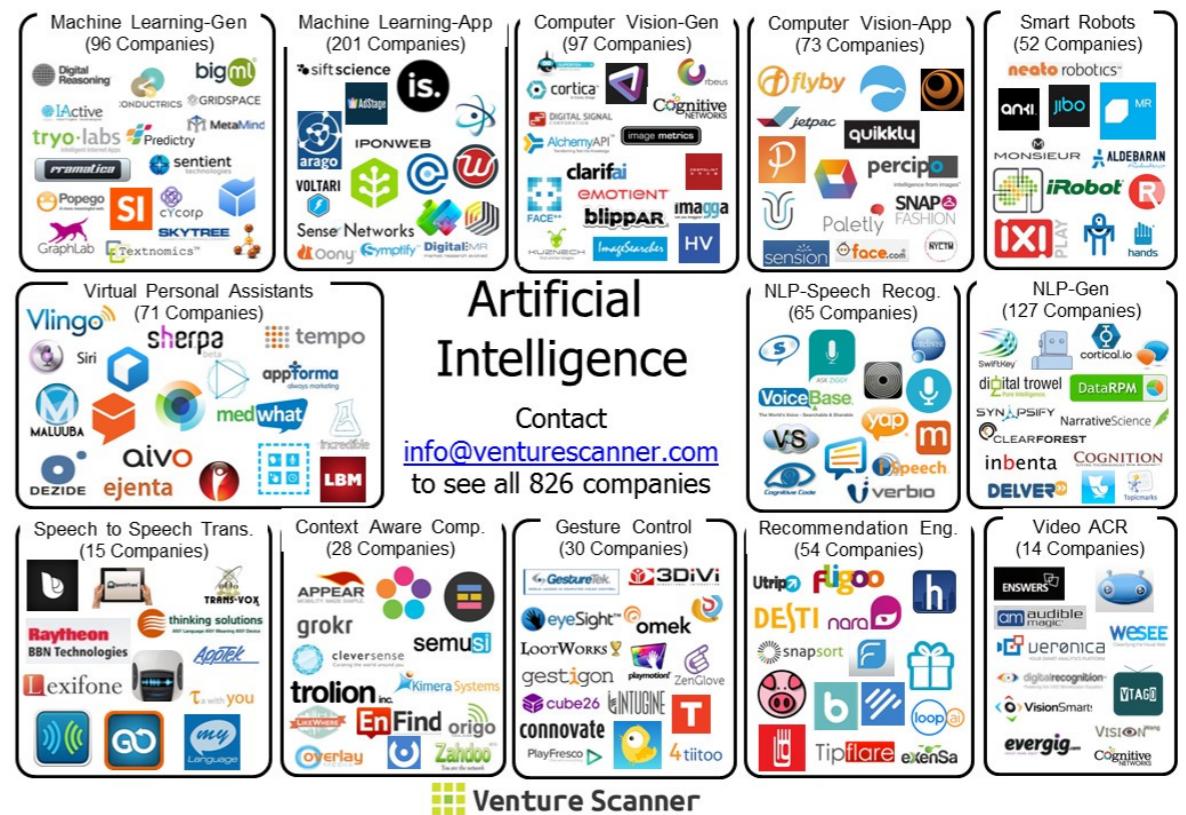
## 오픈소스가 대세 - 지속 가능한 현대적인 모듈

“누구나 sw를 쉽게 만들고 그것이 파괴력을 가지는 시대”



**정치넷**

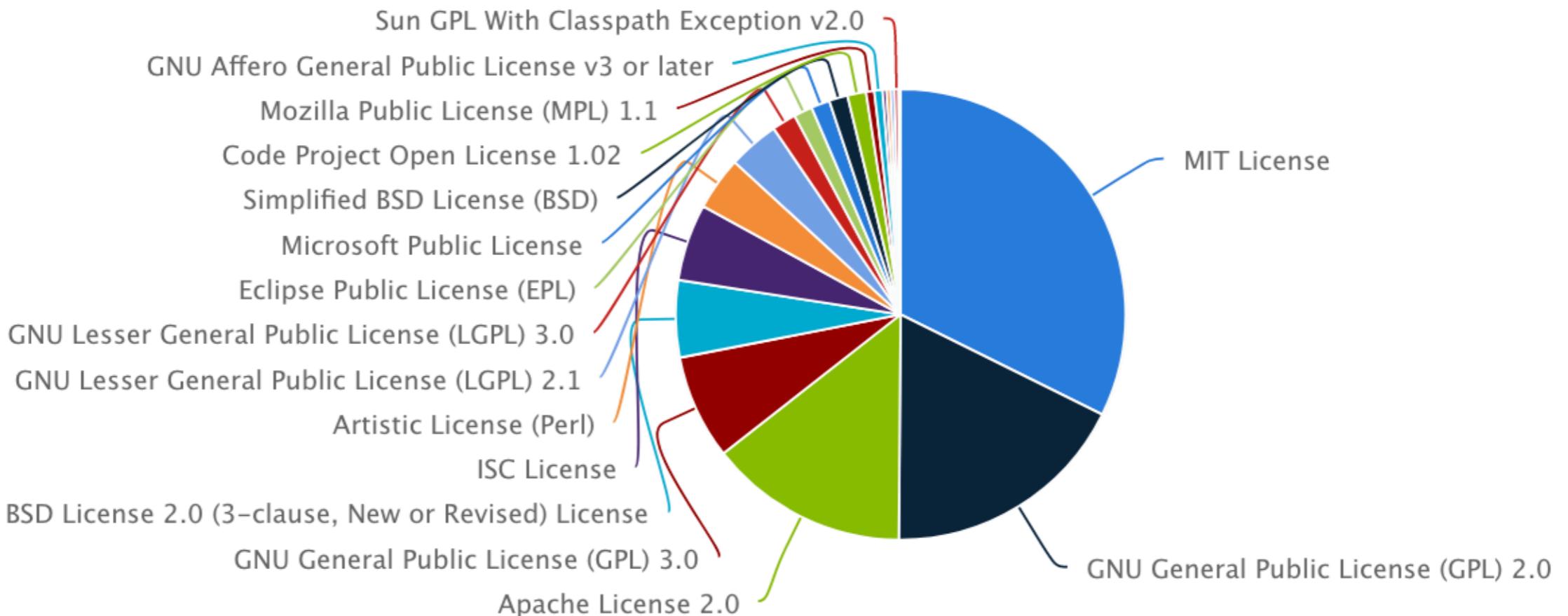
<http://jungchi.net/>



# OSS - 라이선스

## 라이선스 = 사용허가권

- 이용자는 소프트웨어 개발자가 부여한 라이선스 조건에 따라야 함 (위반 시 법적 제재)
- OSS라이선스는 독점SW 라이선스와 동일하게 저작권 등의 법적 보호를 받으나 큰 차이가 있음
- 기본적으로 자유로운 사용, 수정, 배포를 보장
- 대표적 라이선스 : *GPL, MIT License, Apache License, BSD License*
- 라이선스 비교표 : <https://olis.or.kr/license/compareGuide.do#>

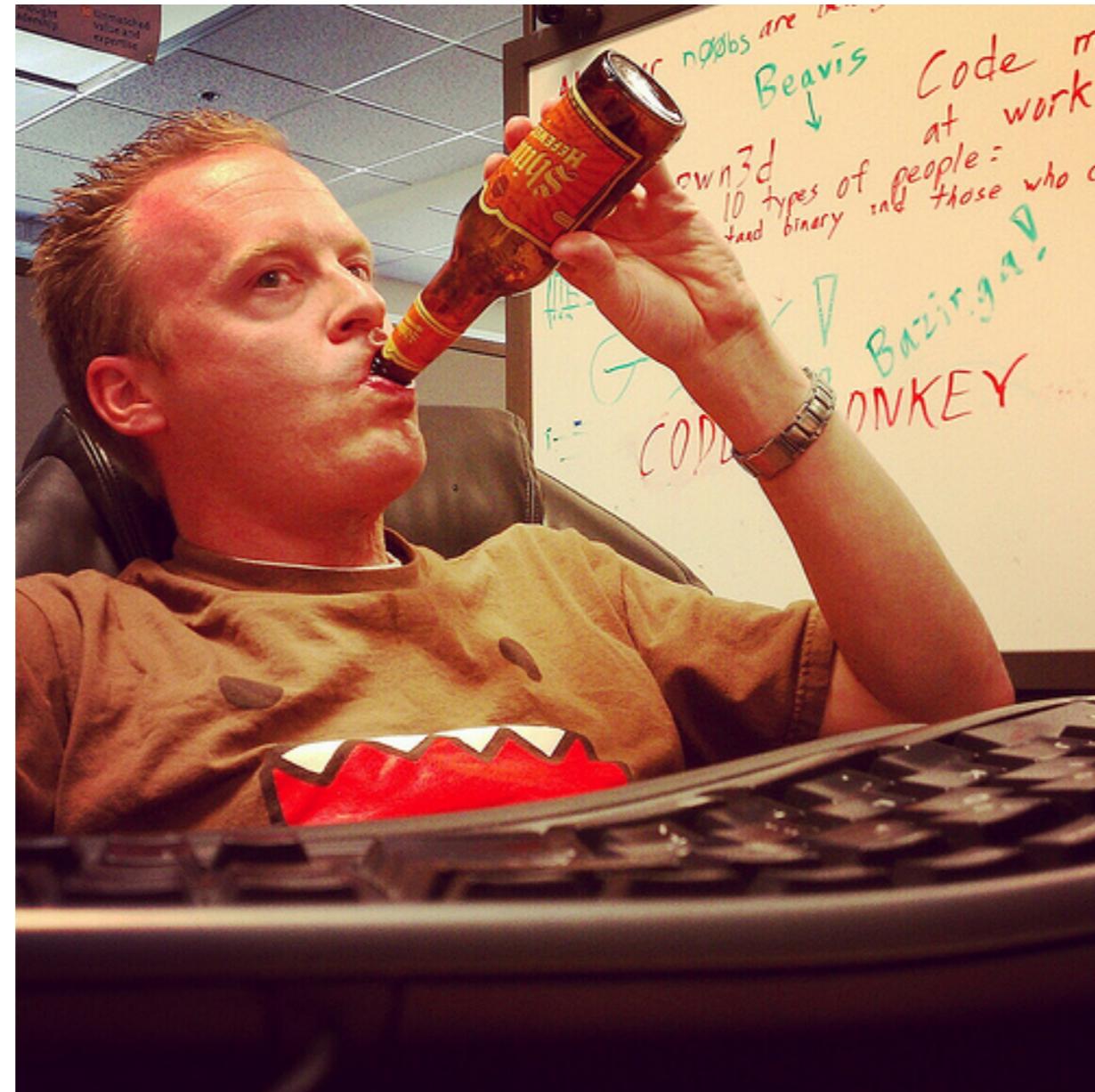


Source: Black Duck KnowledgeBase

# OSS - 라이선스

## 라이선스 = 사용허가권

- 이용자는 소프트웨어 개발자가 부여한 라이선스 조건에 따라야 함 (위반 시 법적 제재)
- OSS라이선스는 독점SW 라이선스와 동일하게 저작권 등의 법적 보호를 받으나 큰 차이가 있음
- 기본적으로 자유로운 사용, 수정, 배포를 보장



뭐가 이리 복잡해! 그냥 나중에 만나면 맥주나 한 잔 사줘!

Beerware License

# OSS - 수익모델

## 라이선스가 아닌 서비스로 수익 창출

- 오픈소스 설치, 커스터마이징, 교육제공 등으로

## 프리미엄(*Freemium, Free + Premium*) 전략

- 기본 기능은 무료, 고급 기능은 유료 (듀얼 라이선스)

## 액세서리, 출판

- 스티커 판매, 오픈소스 전문 출판사



오픈소스 스티커

<https://opensource.com/business/15/11/open-source-stickers>

The screenshot shows the IBM Watson Analytics pricing page. At the top, it says "Watson Analytics pricing plans" and describes the service as "data visualization and discovery in the cloud" that allows users to "analyze data on your own, without downloading software". Below this, there are three main pricing options:

- Free**: \$0/mo. Includes complimentary access to Watson Analytics for 30 days. A "TRY IT NOW" button is available.
- Plus**: \$30/mo. Includes all the analytics of the free version plus the ability to upload larger data assets. A "BUY NOW" button is available.
- Professional**: \$80/mo. Designed for enterprises, includes a multi-user environment to collaborate and more data connectors. A "BUY NOW" button is available.

A "Talk to an expert" button is located on the right side of the page.

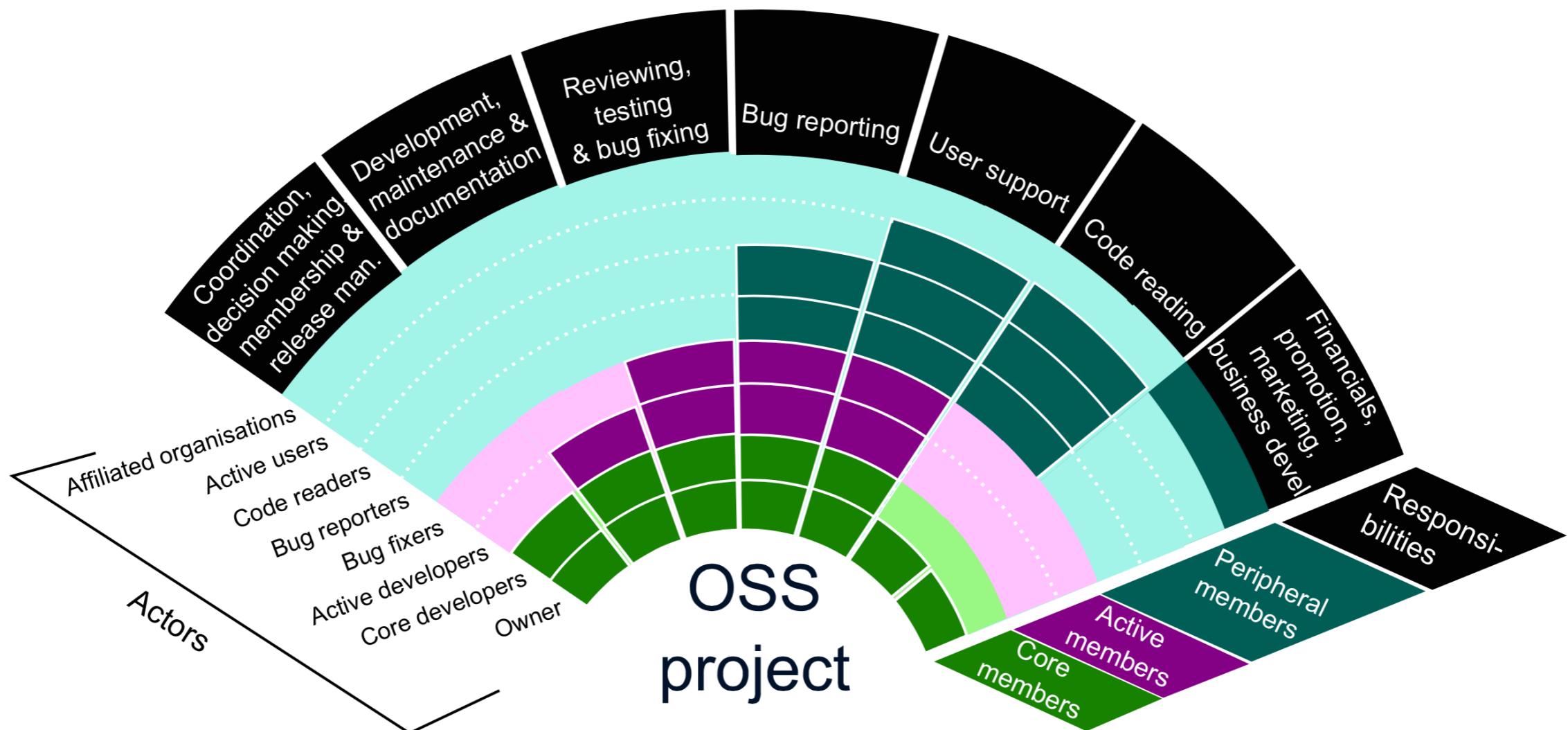
IBM 왓슨 가격정책

<https://www.ibm.com/watson-analytics/pricing>

# OSS - 커뮤니티

각 OSS 프로젝트는 다양한 역할, 능력, 참여 정도, 책임감을 가진 멤버들로 구성

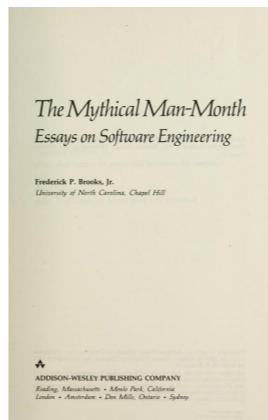
- 코어멤버: 시스템 설계 및 전략적 의사결정
- 액티브 개발자: 코드를 직접 수정
- 버그 리포터: 버그를 발견하고 리포팅
- 문서화하는 사람
- 적극적 사용자



## 맨먼스 미신

“ 지체된 프로젝트에 인력을 추가하는 것은 일정을 더욱 늦출 뿐이다 ”

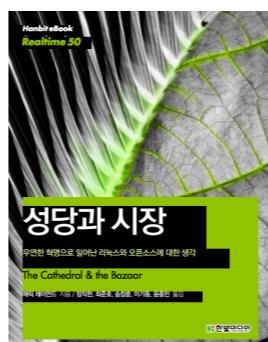
- Frederick P. Brooks Jr (1975)



## 성당과 시장

“ 보고 있는 눈이 충분히 많으면 찾지 못할 버그는 없다 ”

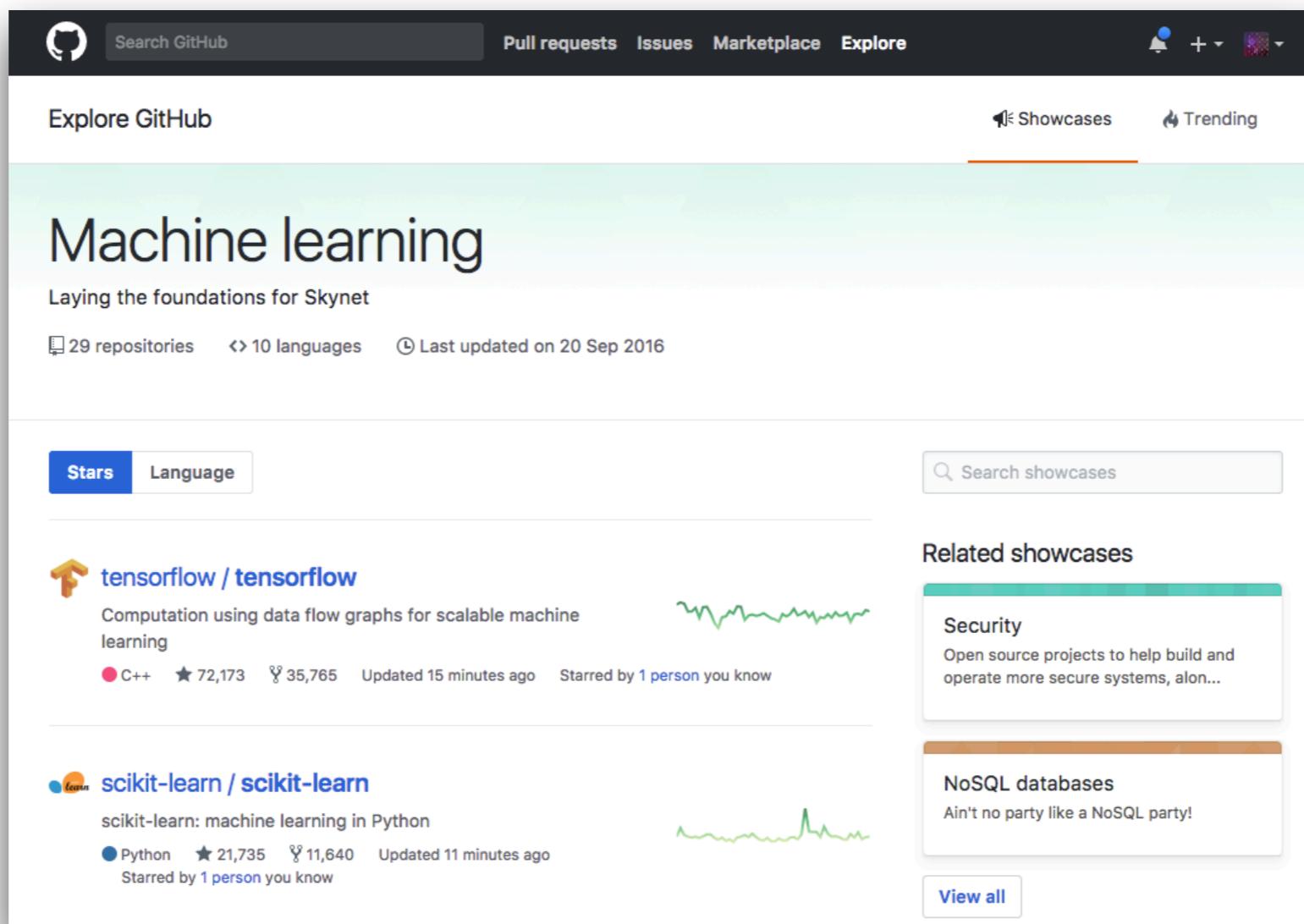
- Eric Raymond



## 깃허브(GitHub)

가장 인기있는 오픈소스 코드 저장소  
Git을 사용하는 협업형 버전관리 서비스

<https://github.com>



출처: <https://github.com/showcases/machine-learning>

# 인공지능 - 개요

## 개념

인공지능(AI: Artificial Intelligence)은 기계가 사람처럼 생각하고 행동하게 만드는 기술

## 철학적 관점의 인공지능

인간의 일을 얼마나 수행하는지에 따라 강인공지능(Strong AI)과 약인공지능(Weak AI)으로 구분 (출처: 위키피디아)

강인공지능	약인공지능
<ul style="list-style-type: none"><li>- 스스로 데이터를 찾아 학습하여 어떤 문제를 실제로 사고하거나 해결할 수 있는 인공지능</li><li>- 다양한 분야에 보편적으로 적용 가능</li></ul>	<ul style="list-style-type: none"><li>- 미리 정의된 규칙을 이용해서 지능을 흉내내는 컴퓨터 프로그램</li><li>- 특정 분야에서만 활용(챗봇 등)</li></ul>

## 싱글래리티 (Singularity)

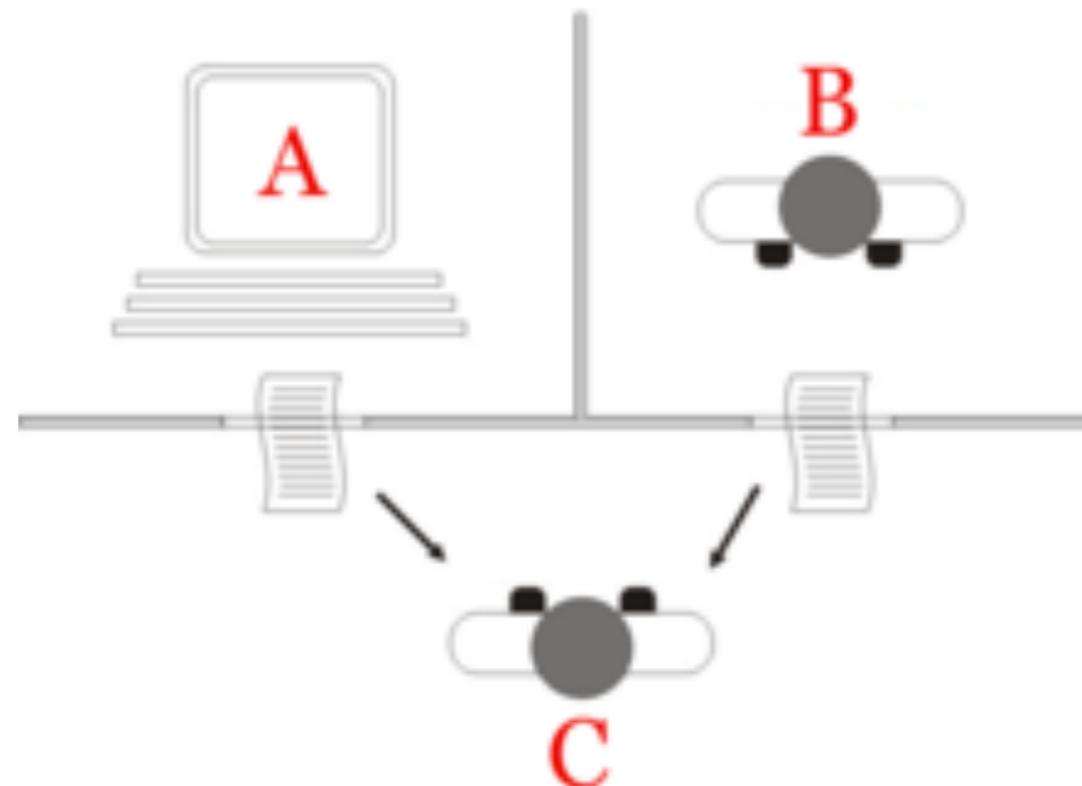
- 인공지능(AI)의 발전이 가속화되어 인간의 지성보다 더 뛰어난 초인공지능이 출현하는 시점
- 미래학자 레이 커즈와일은 이런 강인공지능을 ‘싱글래리티(Singularity)’라는 용어로 설명
- 당초 2045년이면 싱글래리티가 도래할 것이라고 주장했다가 2015년에 이를 2030년으로 앞당김

(출처: 위키피디아, [https://ko.wikipedia.org/wiki/%EA%B8%BC%88%A0%EC%A0%81\\_%ED%8A%B9%EC%9D%B4%EC%A0%90](https://ko.wikipedia.org/wiki/%EA%B8%BC%88%A0%EC%A0%81_%ED%8A%B9%EC%9D%B4%EC%A0%90))

# 인공지능 - 개요

## 튜링 테스트 (Turing test)

- 기계에 지능이 있는지를 판별하는 테스트(앨런 튜링이 1950년에 제안)
- 앨런 튜링은 1950년에 철학 저널 Mind에서, 기계가 지능적이라고 간주할 수 있는 조건을 언급
- 그는 "컴퓨터로부터의 반응을 인간과 구별할 수 없다면 컴퓨터는 생각(사고, thinking)할 수 있는 것"이라고 주장
- 만일 지성 있는 사람이 관찰하여 기계가 진짜 인간처럼 보이게 하는 데 성공한다면 확실히 그것은 지능적이라고 간주해야 한다는 주장 (출처: 위키피디아, [https://ko.wikipedia.org/wiki/튜링\\_테스트](https://ko.wikipedia.org/wiki/튜링_테스트))



[https://en.wikipedia.org/wiki/Turing\\_test](https://en.wikipedia.org/wiki/Turing_test)

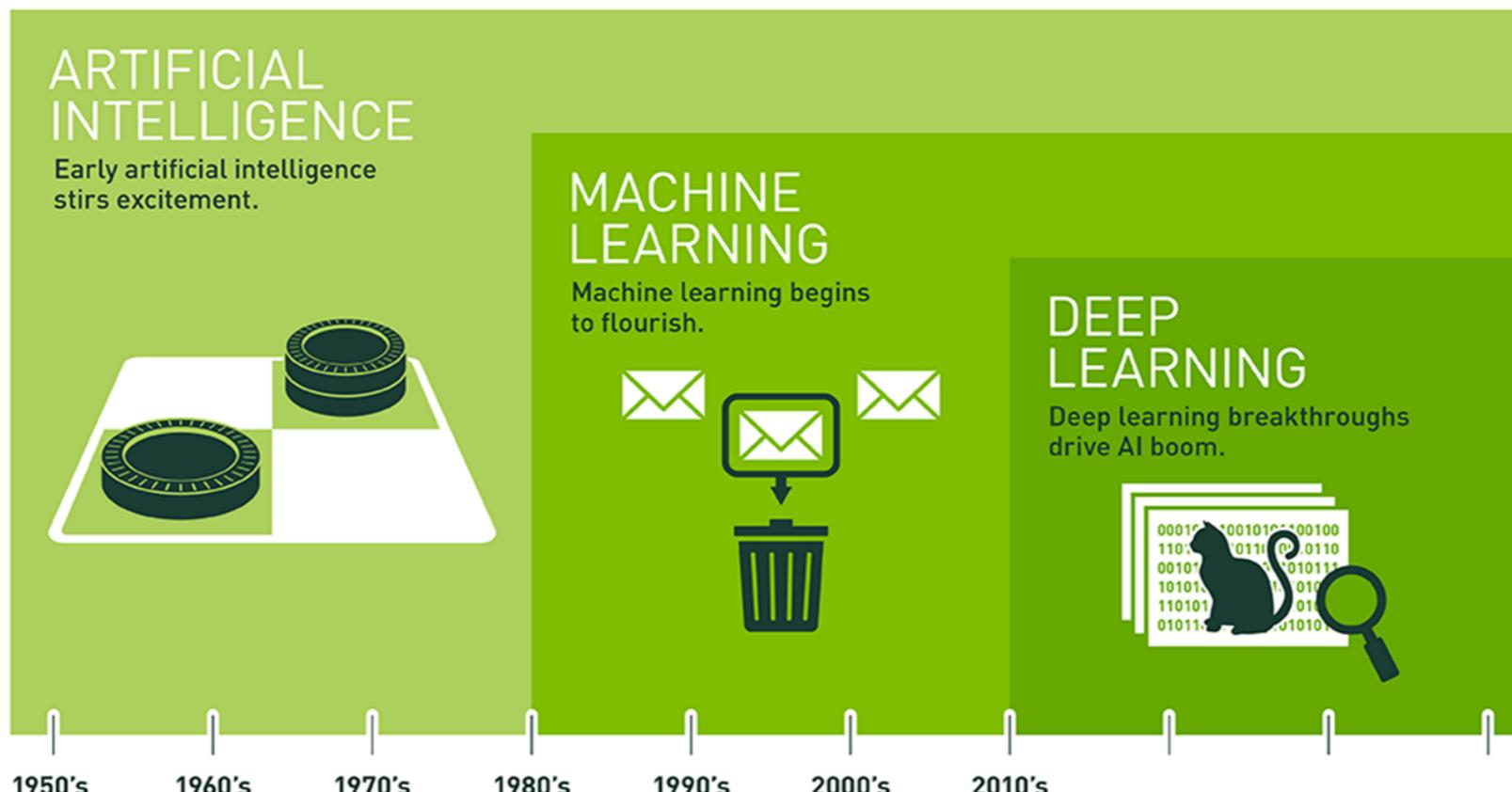
# 인공지능 - 머신러닝, 딥러닝

## 머신러닝 (기계학습)

- 인공 지능의 한 분야로, 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 말한다. 가령, 기계 학습을 통해서 수신한 이메일이 스팸인지 아닌지를 구분할 수 있도록 훈련할 수 있다 (학습한 것만 해결)

## 딥러닝 (심층학습)

- 딥러닝(Deep Learning)은 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야
- 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화를 시도하는 머신러닝 알고리즘의 집합



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

<https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>

# 인공지능 - 응용분야

**자연어 처리** : 텍스트를 이해하는 기술

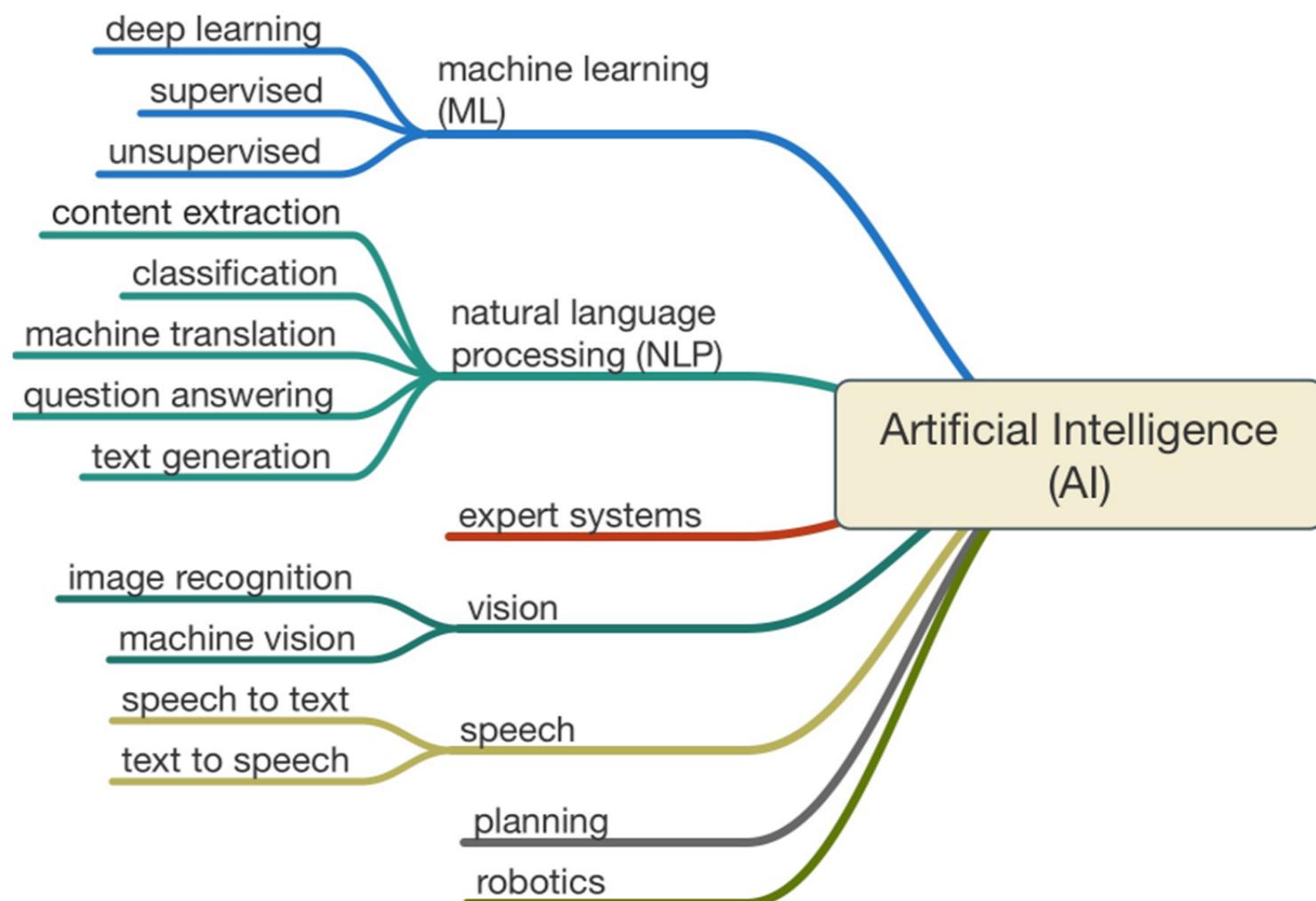
**전문가 시스템** : 조언을 하거나 의사결정을 하는 시스템

**비전** : 이미지나 비디오를 비롯한 시각 데이터 처리

**음성 인식** : 음성 언어를 듣고 이해하는 기술을 연구하는 분야

**계획** : 최소 비용으로 최대 효과를 내는 계획수립 방법을 연구하는 분야

**로보틱스** : 로봇의 설계, 제조 및 응용분야



<https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>

# 인공지능 - 플랫폼

## 클라우드 기반 AI 서비스 상용화

글로벌 IT기업들은 자연어처리, 음성인식, 시각인식 등 사람의 두뇌/입/귀/눈 등에 해당하는 AI 서비스 제공  
오픈소스 프리미엄(Freemium) 모델 적용

IBM 왓슨	Google	Microsoft Cognitive Toolkit	Facebook	Baidu	Amazon	OpenAI	ETRI 엑소브레인
							
<ul style="list-style-type: none"><li>Language 9종</li><li>Speech 2종</li><li>Vision 1종</li><li>Data Insight 3종</li><li>IoT 1종</li><li>총 14종</li></ul>	<ul style="list-style-type: none"><li>Language</li><li>Speech</li><li>Vision</li><li>Data Insight</li><li>IoT</li><li>총 31종</li></ul>	<ul style="list-style-type: none"><li>Language 6종</li><li>Speech 3종</li><li>Vision 6종</li><li>Knowledge 4종</li><li>Search 5종</li><li>총 23종</li></ul>	<ul style="list-style-type: none"><li>얼굴인식, 챗봇 등 다수 소프트웨어 오픈소스 및 API 제공</li><li>Caffe2의 경우 <b>총 47종</b></li></ul>	<ul style="list-style-type: none"><li>Language, 음성인식(Deep Speech) 등 <b>다수 API 제공</b></li></ul>	<ul style="list-style-type: none"><li>자동음성인식 (ASR), 자연어이해 (NLU), TTS, 이미 지분석 등 <b>다수 API 제공</b></li></ul>	<ul style="list-style-type: none"><li>기계학습 테스트 를 위한 <b>다수 API 제공</b></li></ul>	<ul style="list-style-type: none"><li>자연어처리, QA 등 <b>기능 제공</b></li></ul>

# 인공지능 - 체험하기

---

- ▶ 구글 자연어처리 : <https://cloud.google.com/natural-language/>
- ▶ 구글 OCR : <https://cloud.google.com/vision/>
- ▶ 구글 객체 감지 : <https://techcrunch.com/2017/06/16/object-detection-api/>
- ▶ 구글 음성인식 : <https://cloud.google.com/speech/>
- ▶ 구글 비디오인식 및 분석 : <https://cloud.google.com/video-intelligence/>
- ▶ 구글 동영상 온톨리지 : <https://research.google.com/audioset/>
  
- ▶ IBM Watson 음성인식 : <https://speech-to-text-demo.mybluemix.net/>
- ▶ IBM Watson Visual Recognition : <https://visual-recognition-demo.mybluemix.net/>
- ▶ IBM Watson 자연어처리 : <https://natural-language-understanding-demo.mybluemix.net/>
  
- ▶ Summarizer : <http://esapi.intellexer.com/Summarizer>
- ▶ TensorFlow Text summarization : <https://github.com/tensorflow/models/tree/master/textsum>

## 전문기술 직종의 작업과정

1. Gather data (데이터 수집)
2. Analyze the data (데이터 분석)
3. Interpret the results (결과 해석)
4. Determine a recommended course of action (실행계획 결정)
5. Implement the course of action (실행)

## 인공지능 시대 살아남기

1. 인공지능 도구와 친숙하게 지내기
2. 학습능력 및 문제해결 능력의 향상
3. Emotional Intelligence 높이기

(출처: 김인택, 인공지능과 제4차 산업혁명)

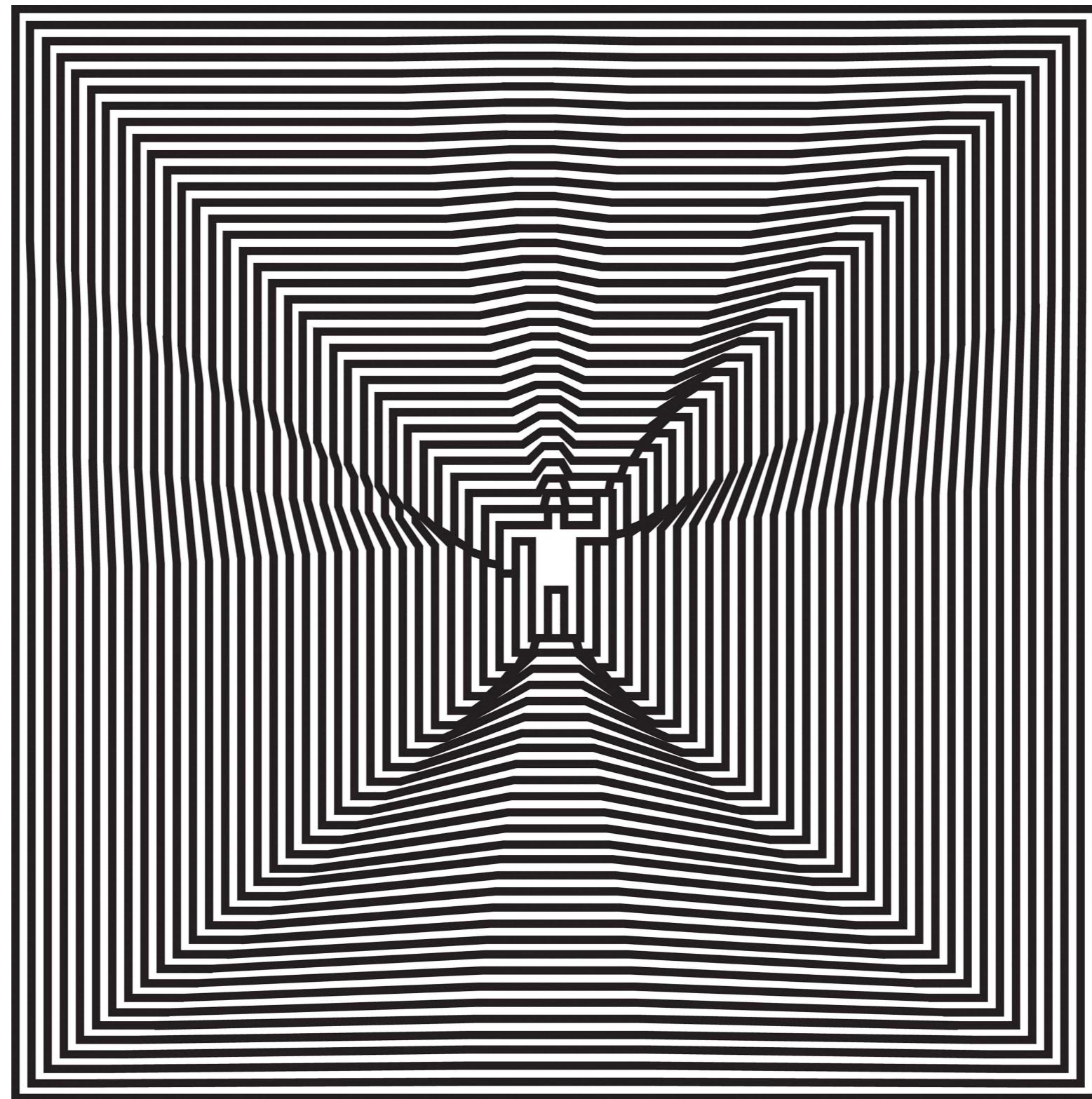
# *Hello, Robot.*

---

Christoph Niemann,  
*Robot Morph*, 2016 ©  
Christoph Niemann.

[http://www.mak.at/e/  
hello\\_robot](http://www.mak.at/e/hello_robot)

[http://www.viennabiennale.org/  
en/exhibitions/](http://www.viennabiennale.org/en/exhibitions/)



감사합니다

[daejin@archivelab.co.kr](mailto:daejin@archivelab.co.kr)