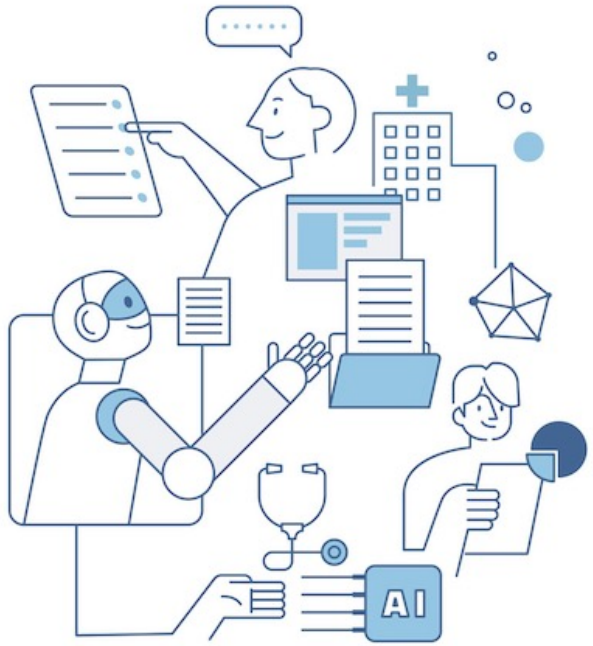
아주대 의료인공지능 융합인재양성 사업단 – 인실리코젠
산학협동 프로그램

의료인공지능 부트캠프



일시 10월 7일(금), 10월 14일(금) 오후 6시

장소 아주대학교 의과대학 송재관 2층 여유당

주 제 의료인공지능 입문자들을 위한 파이썬/인공지능 실습 부트캠프

대상자 ①전공과정 이수생 ②일반인(아주대 소속 학생 우선 선발)

모집인원 50명 이내

접수기한 2022.10.05(수), 12:00PM(정오)까지

신청방법 구글폼 신청서 제출

비 용 무료

- **목표**: 파이썬 기본 활용법을 익히고 인공지능 분석 사례를 확인하며, 이를 실습함으로써 실무 활용 역량 학습
- **실습환경**: 쉘 프롬프트+IDLE, Jupyter, Google Colab

구글폼 신청서

일정	강사	내용
10.07(금) 18:00 ~ 19:20	 김형용	파이썬 기본활용과 객체지향 프로그래밍
10.07(금) 19:30 ~ 20:50	 박주희	Jupyter와 pandas를 이용한 데이터분석 기초
10.14(금) 18:00 ~ 19:20	 권영인	머신러닝 이해와 의료데이터 기반 머신러닝 실습
10.14(금) 19:30 ~ 20:50	 이주연	딥러닝 기초 이론과 의료데이터 기반 딥러닝 실습

AIMED Artificial Intelligence in Medicine 아주대학교 의료인공지능 융합인재양성사업단

INSILICOGEN www.insilicogen.com

아주대학교 대학원 의생명과학과

교육 참고자료 제공:

https://incodom.kr/ISG-Edu22-10

의료인공지능 부트캠프 1일차 첫째시간

파이썬 기본 활용과 객체지향 프로그래밍

2022. 10. 07. 김 형 용

INSILICOGEN

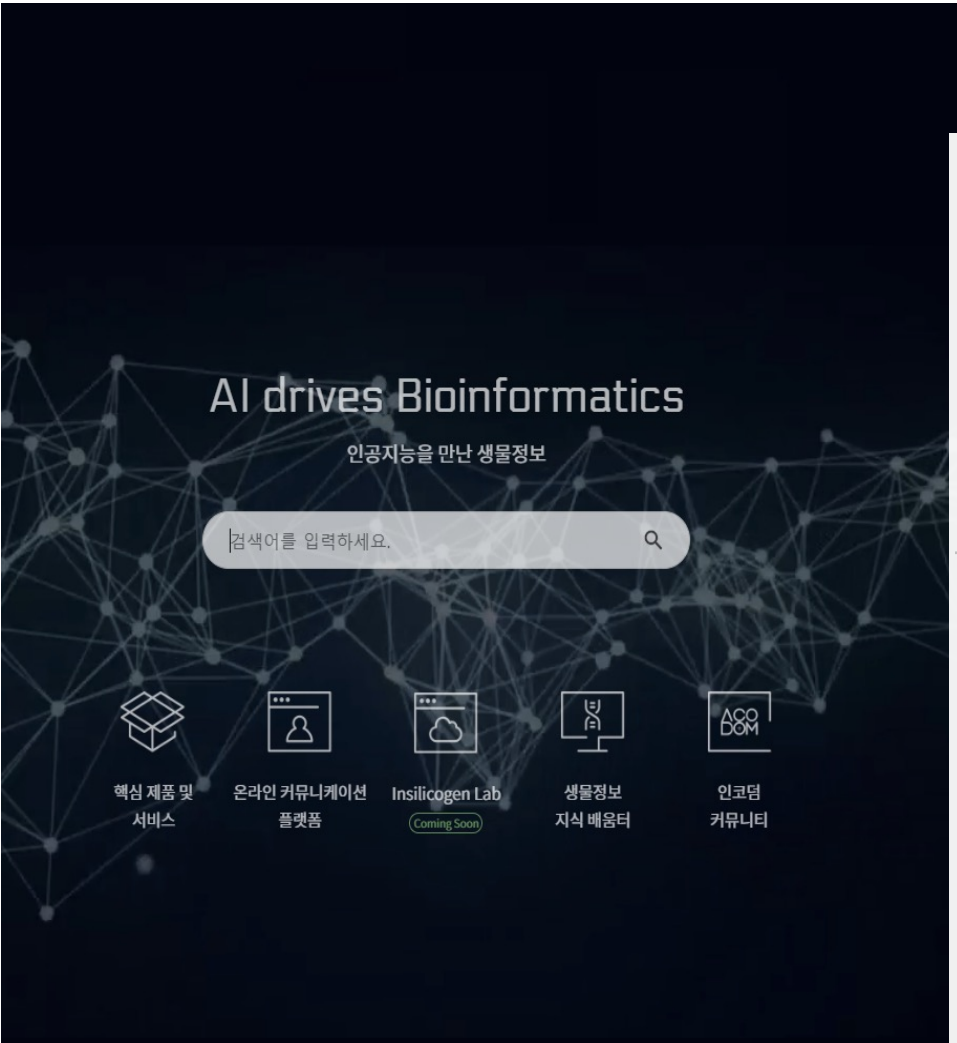
www.insilicogen.com

INDEX

01

인실리코젠 FLEX실 소개

㈜인실리코젠, 생물정보 전문기업



생물학적 데이터 분석 및 플랫폼 구축이 가능한 생물정보 전문기업

(주)인실리코젠

정밀의학, 디지털육종, 맞춤식품 등 첨단과학에서 일상까지 빅데이터의 시대가 현실로 다가왔습니다.
(주)인실리코젠은 Bioinformatics 전문기업으로 바이오 빅데이터의 바다를 함께 할 든든한 테크놀로지의 동반자가 되겠습니다.

비전	생물정보 전문 기업
핵심가치	人Co는 사람을 중심(Core)으로, 사람과 컴퓨터(Computer)에 의해, 배려(Consideration)와 소통(Communication)을 통한 새로운 문화 창조
미션	생물정보의 공유와 소통을 통한 새로운 가치사슬의 창조
대표자	최남우

INSILICOGEN
www.insilicogen.com
㈜ 인실리코젠

- 2004 **인실리코젠 설립**
- 총 52 인력 (2021. 65억 매출)
- 8개 부처 정부과제 **124**개 중 **56**과제 수행
  (생물정보 분석 및 정보화 DB 구축 분야, 11'~18')
- 분석컨설팅, **60**종의 유전체 프로젝트 완료
  (년간 40 개 이상의 분석 프로젝트 수행)
- 2021 우수 기업연구소 지정 (과학기술정보통신부)
  기계학습 관련 특허 3건
  최근 5년간 40 편의 연구 논문 출판 (22'~17')

과학기술정보통신부
2021. 06. 22

인공지능과 빅데이터 중심으로 미래를 선도하는 생물정보 전문기업

(주)인실리코젠은 Bioinfomatics 전문기업으로 바이오 빅데이터 서비스의 바다를 함께 할 든든한 테크놀로지의 동반자가 되겠습니다.

Business A

생물정보 분석 서비스

Analysis

36종 유전체 해독
600TB 원천 데이터 분석

(한우, 전복, 광어, 우럭, 고추
고구마, 오이, 참돔, 버섯…)

Business B

시스템 통합

SI

30개 기관
100개 시스템 구축

(CODA, NABIC, MAGIC,
치매빅데이터플랫폼 AIzNAVi
6.25 전사자 및 실종아동찾기
유정정보 검색시스템…)

Business C

인공지능

AI

다양한 연구과제 진행

(한우등심영상 기계학습
등급판정, 반려견퇴행성
유전질환바이오마커발굴,
멀티오믹스 데이터,
골다공증 영상 데이터 …)

Business D

솔루션

Solution

120개 기관
100,000명 고객확보

Business E

교육 서비스

Education

50개 기관
10,000명 인력 양성

다양한 교육 프로그램 보유

ACO SEMINAR

- 생물정보 기초 교육 프로그램
- 이론 및 생물정보 솔루션 활용 방법 교육

ACO WORKSHOP

- 생물정보 분석 역량 강화를 위한 맞춤형 교육
- 실무 예제 중심의 실습 교육 제공

ACO INTERNSHIP

- 생물정보 인재양성 프로그램
- 현장 실습을 통해 바이오 연구개발과 조직생활의 기초개념을 이해할 수 있는 기회 제공

ACO ACADEMY

- 온라인 교육센터
- 생물정보 솔루션을 활용한 다양한 교육 콘텐츠 제공 및 생물정보 분석을 수행할 수 있도록 지원

kobic 국가생명연구자원정보센터 Korean Bioinformation Center

- 국가생명연구자원정보센터(KOBIC)와 협업
- 생명정보 전반 내용으로 교육 진행

koreaBio 한국바이오협회 / 산업통상자원부

- 한국바이오협회(Koreabio)와 협업
- 산업체 현장 수요기반으로 유전체 데이터 분석 교육 진행

멀티오믹스 + 인공지능

데이터를 넘어 지식이 되기 위한 과정에 인공지능 기술이 함께합니다

생물정보학에서 인공지능 기술은 이전부터 중요하게 활용되고 있습니다.
알려진 정보들을 기계 학습하여, 알기 어려운 것을 알 수 있게 합니다.
최근의 빅데이터와 딥러닝의 지속적인 기술 발전은 그 가능성을 더 높여줍니다.

멀티오믹스(Multi-omics) 뿐 아니라, 문헌, 영상, 네트워크 등 복잡한 데이터의 홍수 속에서 고민이 많으십니까? 또는 소유하신 데이터의 적절한 통계적 방정식(규칙)을 찾기 힘드십니까?

인실리코젠의 인공지능 기술과 함께 귀하의 데이터에 숨겨진 의미와 가치를 찾고 고급 지식으로 만드십시오. 인실리코젠은 일반적인 문자 데이터부터 Sequential 데이터(시계열), 영상 이미지 및 의·약학 분야 대용량·비정형 빅데이터를 다룰 수 있으며 이들을 구조화 및 상호 연계, 기계학습, 특징선택/추출 방법으로 숨은 지식의 발견과 통합적 이해를 지원합니다.

인간

- 질병 연관 변이 분석(WES, Targeted)
  CLC Genomics Workbench
  QIAGEN Clinical Insight Interpret(QCII)
- 유전자 발현 및 네트워크 분석(RNA-seq)
  CLC Genomics Workbench Premium
  Ingenuity Pathway Analysis(IPA)
  OmicSoft Land Explorer

동·식물

- 유용 형질 관련 변이 분석
  CLC Genomics Workbench
- 유전자 발현 분석(RNA-seq)
  CLC Genomics Workbench
  OmicsBox Transcriptomics Module

미생물

- 유전자 구조 및 기능 분석
  OmicsBox Genomics Module
  OmicsBox Functional Analysis Module
- 메타게놈 분석
  CLC Microbial Genomics Module
  OmicsBox Metagenomics Module

전사체 분석

- 표준 전사체 어셈블리
- RNA-seq based 유전자 발현치 추정
- DEG/Pattern 분석
- GSEA 분석
- Pathway 분석
- 프로모터/발현 연관 분석
- 네트워크 분석

유전체 분석

- 유전체 사이즈 예측
- 전장 유전체 어셈블리
- Repeat element 분석
- 유전자의 구조 예측
- 단백질 기능 구조 예측
- 단백질 기능 예측
- Long non-coding RNA 분석
- Promoter 분석
- Gene family 분석
- Phylogenetic tree 분석
- GBS 활용 linkage map 작성

변이 분석

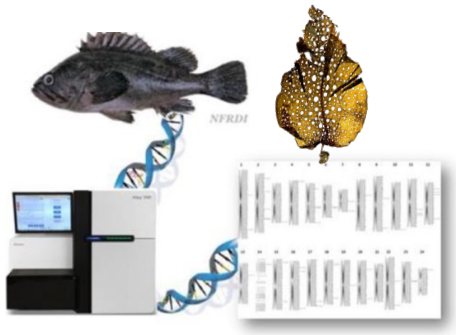
- Potential SNPs 분석
- Potential SSRs 분석
- 종/원산지 판별 마커 발굴
- 유용형질/육종 관련 마커 발굴
- 유전병/질환 연관 SNP 분석
- Population structure 분석
- Phylogenetic tree 분석
- Selective sweep 분석
- Genomic selection 분석
- 유효집단 개수 추정
- GWAS 분석
- 기계학습을 이용한 변이마커 개발

후성유전체 분석

- 어레이/시퀀싱 기반 후성 유전체 분석
- Methylation/Histone mark·miRNA 분석
- DMR & Pattern 분석
- GSEA & Pathway 분석
- 발현/조절 통합분석

연구 성과 – 유전체 분석

해양 생물 20종 유전체 해독



[출처] 연합뉴스(16.07.22)

포스트게놈유전체 사업을 통한 수산생물 12종, 해조류 8종 유전체 해독에 성공

해양수산부 제주대학교 JEJU NATIONAL UNIVERSITY 성균관대학교 SUNG KYUN KWAN UNIVERSITY(SKKU) 경상대학교 GNU Gyeongsang National University

한국 참전복 유전체 완전 해독

[출처] 연합뉴스(16.04.04)

우리 연안 얕은 바다에 사는 참전복의 유전체를 완전 해독하는데 성공

국립수산과학원 National Institute of Fisheries Science C&K genomics

고추 유전체 해독


nature genetics

Menu ▸ Advanced search Search Go

home ▸ archive ▸ issue ▸ article ▸ full text

NATURE GENETICS | ARTICLE OPEN

日本語要約

Genome sequence of the hot pepper provides insights into the evolution of pungency in *Capsicum* species

Seungill Kim, Minkyu Park, Seon-In Yeom, Yong-Min Kim, Je Min Lee, Hyun-Ah Lee, Eunyoung Seo, Jaeyoung Choi, Kyeongchae Cheong, Ki-Tae Kim, Kyongyong Jung, Gir-Won Lee, Sang-Keun Oh, Chungyun Bae, Saet-Byul Kim, Hye-Young Lee, Shin-Young Kim, Myung-Shin Kim, Byoung-Cheorl Kang, Yeong Deuk Jo, Hee-Bum Yang, Hee-Jin Jeong, Won-Hee Kang, Jin-Kyung Kwon, Chanseok Shin *et al.*

Affiliations | Contributions | Corresponding author

Nature Genetics **46**, 270–278 (2014) | doi:10.1038/ng.2877

Received 23 October 2013 | Accepted 30 December 2013 | Published online 19 January 2014

[출처] Nature Genetics 46. (14.01.19)

고추 유전체 서열 국내 독자 기술로 완성
순수 우리나라 기술로 완성한 첫 번째 식물 유전체 표준 염기 서열

농촌진흥청 서울대학교 SEOUL NATIONAL UNIVERSITY

한우 유전체 서열 해독

[출처] 연합뉴스(16.04.04)

미 국립생물정보센터(NCBI)에 등록된 소의 표준 서열과 비교할 때 92%에 해당하는 한우 유전체 서열 해독 성공

농림수산식품부 Ministry for Food, Agriculture, Forestry and Fisheries 영남대학교

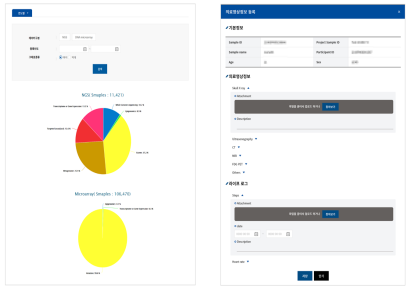
연구 성과 – SI

임상생명유전체시스템 (CODA)

포스트 다부처 유전체 사업에서 생산된 다양한 질병의
임상 정보와 오믹스 데이터를 등록, 보존, 공유하는 시스템

질병 임상 및 오믹스 데이터의 대규모 데이터 분석 가능하도록
아카이브 구축을 수행

< CODA 메인 화면 >



< CODA 데이터 등록 및 통계 현황>

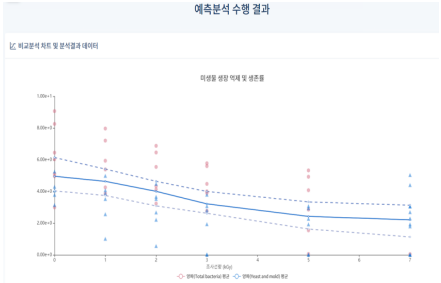
발주처 : 질병관리청

방사선 반응 분석 플랫폼 및 DB 구축 (RRM)

유무기 소재 및 생명체 방사선 반응 정보의 체계적 수집, 관리,
통합, 예측을 위한 국가 방사선 반응지도 모델링 플랫폼 구축

방사선 반응 빅데이터(RRM) 구축 및 방사선 반응 예측(모델링)
시스템(RAD-NET) 개발을 수행

< RRM 메인 화면 >



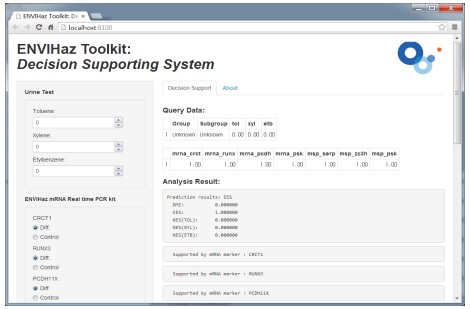
< 양파 내 미생물 생장 억제 및 생존율 분석 예측 >

발주처 : 한국원자력연구원

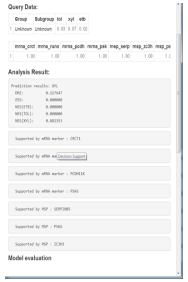
환경유해화학물질 위해성 예측 의사결정 통합 툴킷 개발

환경유해화학물질 위해성 예측 가능 통합 관리를 위한
예측 시스템 구축

환경유해화학물질 노출군을 평가하여 사전 예측 평가를 위한
의사결정지원시스템 개발을 수행



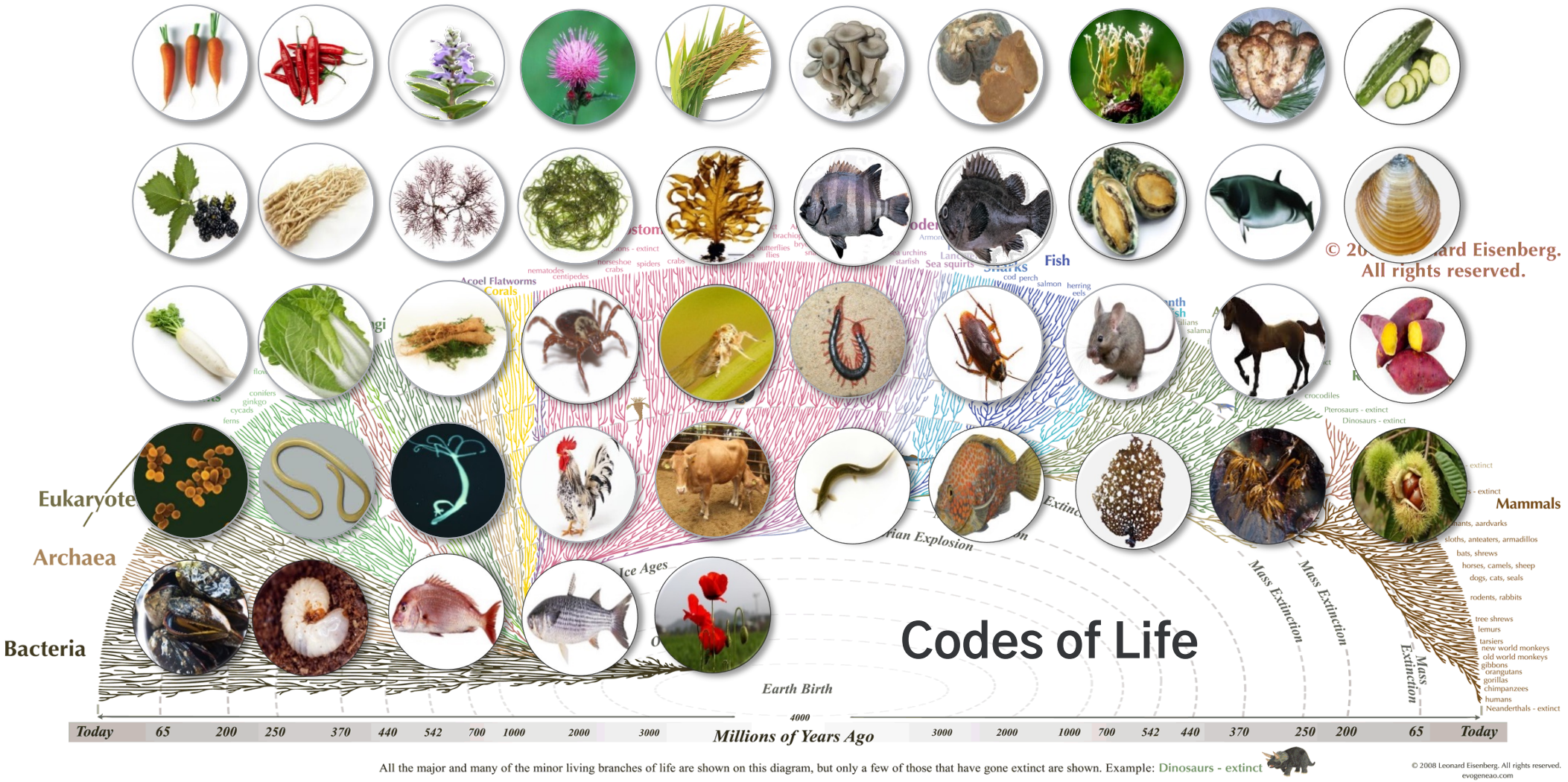
< 환경유해화학물질의 위해성 예측을 위한 의사결정 지원 시스템 화면 >



< 위해성 예측 모델을 통한 사전 예측 평가 결과 화면 >

발주처 : 환경부

연구 성과 – 50 New Genomes Finished by ISG



국내 유일 생물정보 플랫폼

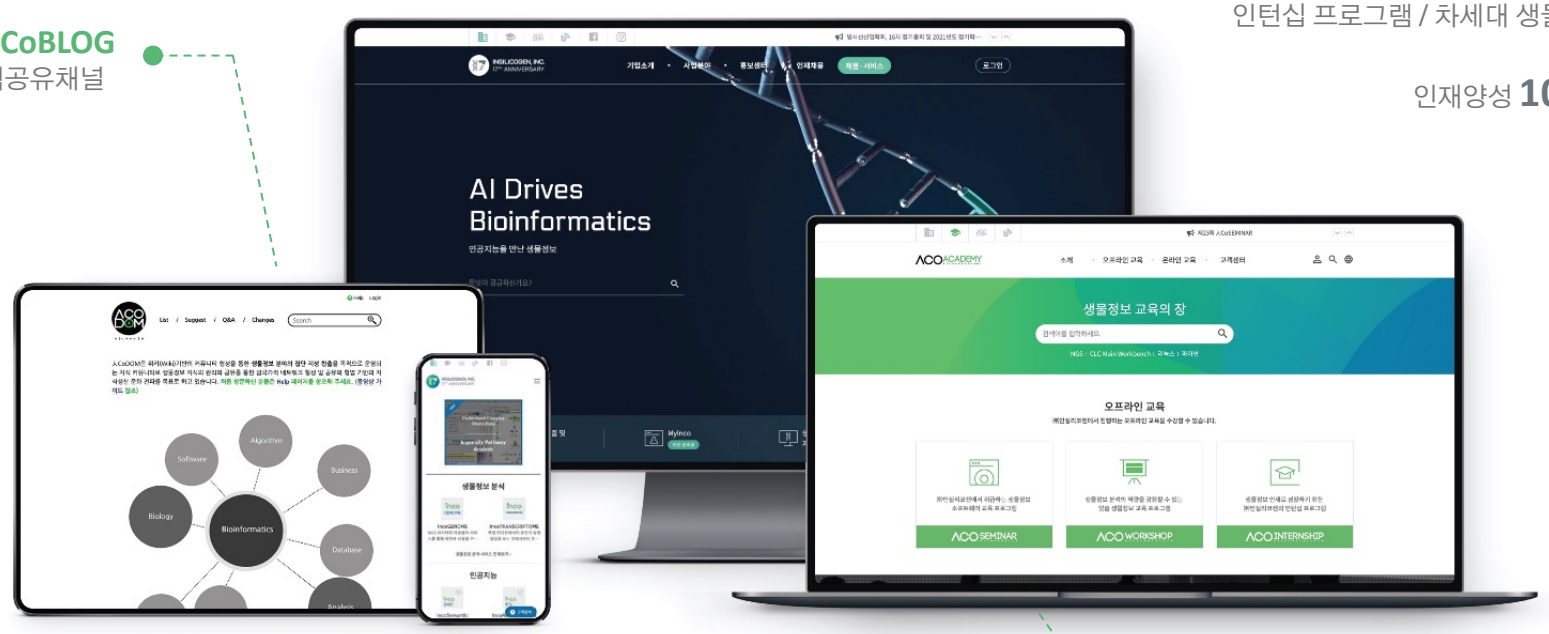
생물정보 플랫폼 이용자 : 월평균 7만명 이상

빅데이터 검색 기반 생물정보 분석
생물정보 지식연계 플랫폼

人CoDOM / 人CoBLOG
생물정보분야 지식공유채널

Off-line 생물정보 교육
인턴십 프로그램 / 차세대 생물정보 교육 운영

인재양성 **10,000**명



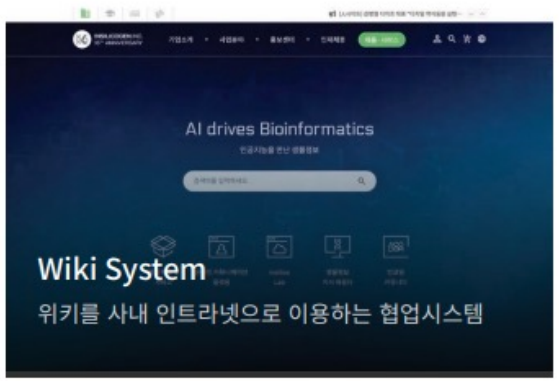
- **Wrote 2,000 articles**
- **Google search engine**
- **Route for advertisement**
- **Make sales opportunities**

Bioinformatics Platform
월평균 이용자 **70,000**명
ref. google analytics (2020.12)

人CoACADEMY / MyInco
온라인 교육채널 / 온라인 구매채널

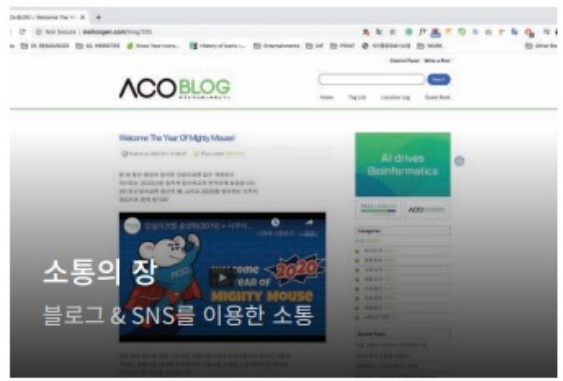
창의적이고 긍정적인 마인드의 토대 위에
새로운 가치를 발굴하고 새로운 문화를 전파합니다.

사내 지식관리시스템을 통한 아이디어 및 업무 공유, 환경 캠페인, 독서경영, 컬처데이 등 다양한 문화를 발굴하고 발전시켜
최고의 회사가 되기 위한 경쟁력을 만들어 갑니다.



Wiki System

위키를 사내 인트라넷으로 이용하는 협업시스템

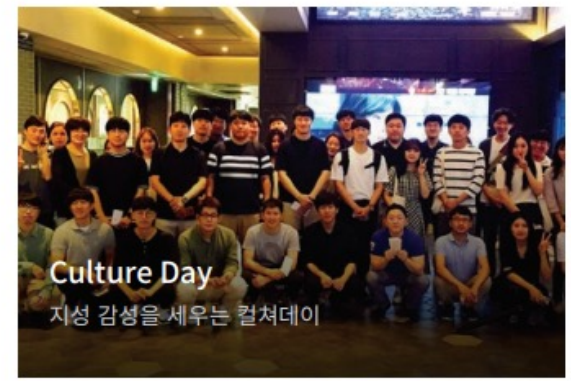


소통의 장

블로그 & SNS를 이용한 소통

환경 캠페인

식물과 함께 동고동락



Culture Day

지성 감성을 세우는 컬쳐데이

지속 성장

세미나·블로그·연재·독서경영을 통한 지속 성장

건강 증진

건강한 몸과 마음을 위한 건강지킴이

한마음 대회

人CoPLAY와 단합대회를 통한 임직원 활동

집단 지성 창출

생물정보 공유의 장을 만들어가는 人CoDOM

#FLEX

[사전적 의미] 구부리다, 힘을 주다, 근육을 수축시키다

FLEX는 유연한 사고(Free)로, 숨은 잠재력을 끌어올려(Lift),
구체적인 계획(Earn)에 의해, 뛰어넘는다는(eXcel) 의미로

새롭게 출범하는 인실리코젠의
통합 서비스 개발 조직


서비스 기획


UX/UI


마케팅


프론트엔드 개발


404
백엔드 개발

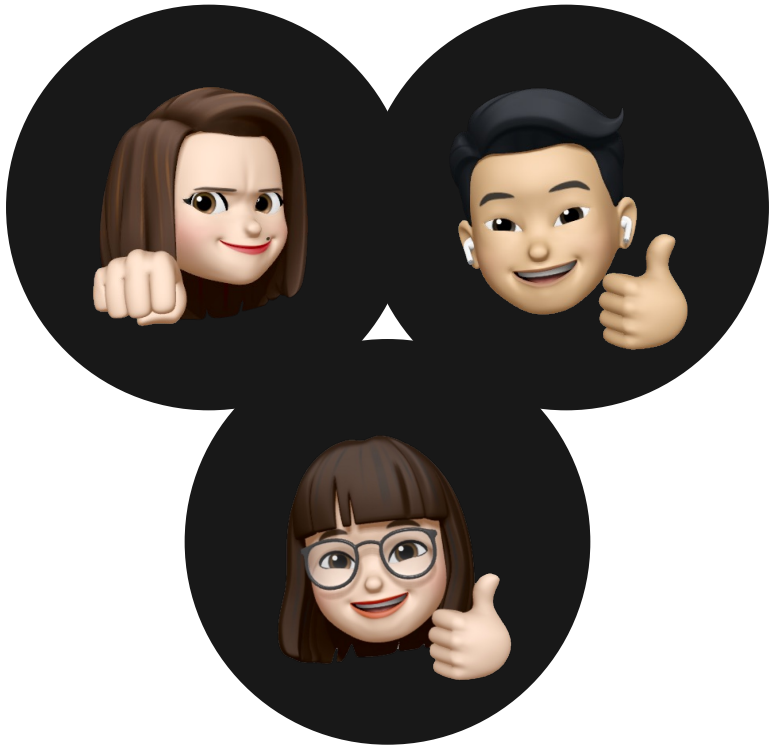

AI 모델 개발

AI Part

Data Analysis — AI Modeling

이주연 | Researcher

권영인 | Researcher



박주희 | Researcher

직관보다는 통계, 수학, 프로그래밍 등
복합 지식 데이터 기반의 의사 결정

주요 업무

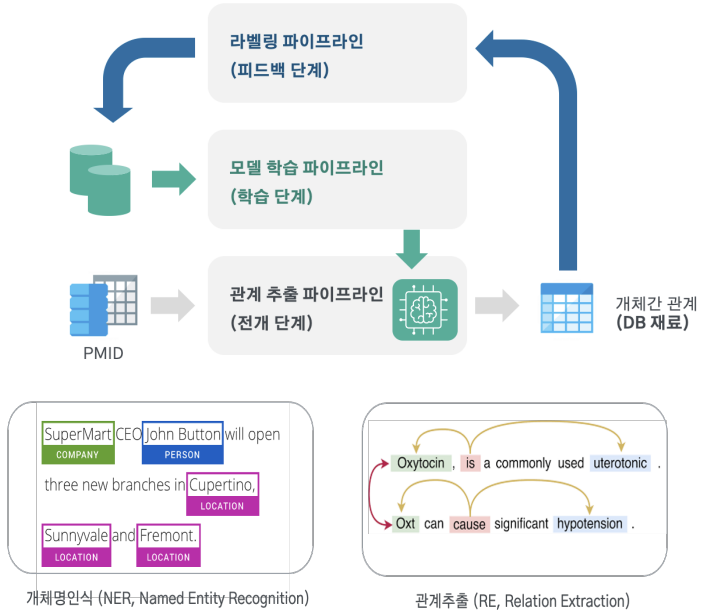
- 통계 기반 빅데이터 분석
- 머신러닝 기반 데이터 분석 및 예측
- 딥러닝 기반 Image Detection, Classification and Segmentation
- 딥러닝 기반 자연어 처리 및 Entity Recognition and Relation Extraction
- 머신러닝 및 딥러닝 기반 추천 시스템

인상깊은 프로젝트

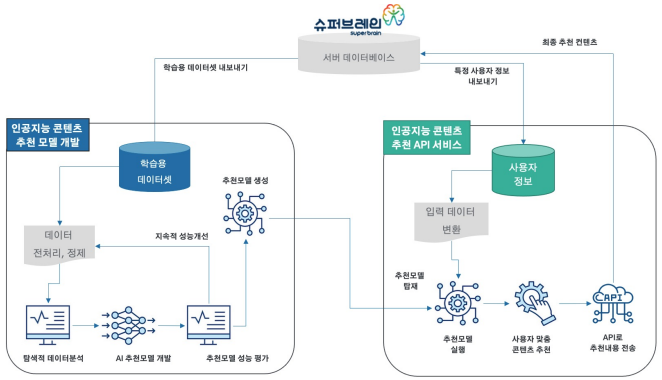
- [2020.02 ~ 2020.12] 소도체 기계화 품질 평가 시스템 구축
- [2021.08 ~ 2022.02] 양식 넙치 양성 예측 프로그램 구축
- [2021.01 ~ 2022.06] 인공지능 기반 골다공증 유병 여부 예측
- [2022.04 ~ 현재] 질병관리청 헬스케어 AI파이프라인 개발 과제

업무 사례

골다공증 위험 예측 논문 및 특허



NLP 분석 파이프라인



AI기반 콘텐츠 추천 시스템

02

파이썬 개요, 운영 환경

Excel, R, **Python,…**

컴퓨터 프로그래밍을 통한 생물학 데이터 분석

Bio-sequence, **NGS**, AI, ⋯

방대한 데이터를 내 입맛대로 분석해 주는 프로그램은 없다.

→ 파이썬으로 직접 분석하는 프로그램 만들기

프로그래밍 언어

컴파일 언어

인터프리터 언어

특수목적 언어

General, Interpreter language

- 보통 스크립트를 만든다고도 함.
- 머리속의 아이디어를 쉽고 빠르게 구현
- Battery included. 이미 만들어진 라이브러리 이용
- Prototype → Product

- **가장 많은 배터리**
- **꾸준한 성능 개선**
- **손쉬운 C 확장**
- **벡터기반 연산**

python™

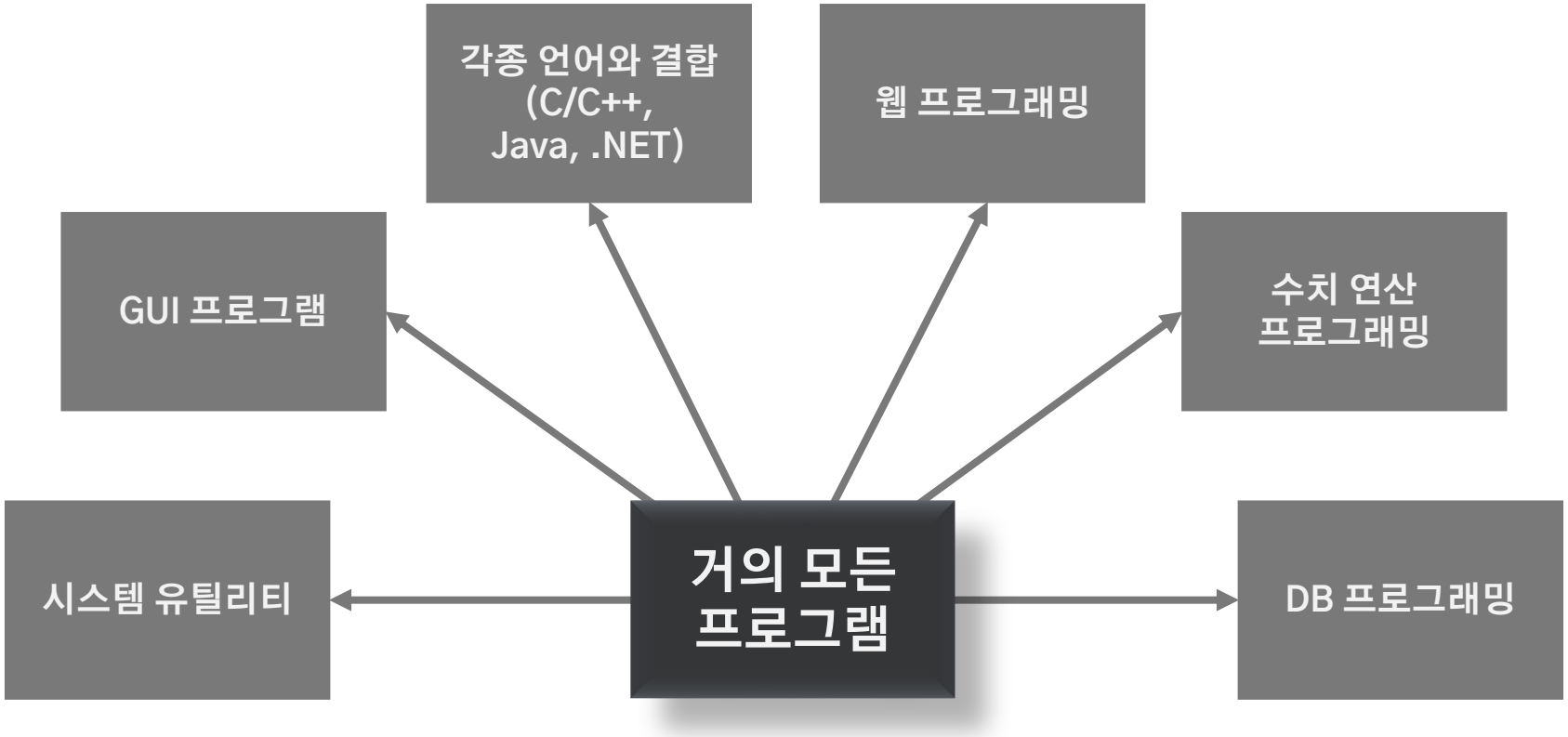
파이썬(Python)은 1991년 프로그래머인 귀도 반 로섬(Guido van Rossum)이 발표한 고급 프로그래밍 언어로, 플랫폼 독립적이며 인터프리터식, 객체지향적, 동적 타이핑, 함수형 언어이다.

파이썬이라는 이름은 귀도가 좋아하는 코미디〈Monty Python's Flying Circus〉에서 따온 것이다.

파이썬의 특징

- 다중패러다임 – 좋은 것을 모두 모음 – **“절차적”, “명령형”, “함수형”, “객체지향형”**
- 플랫폼 독립 – 윈도우, 유닉스, 맥 등 다른 운영체제에서도 동일하게 사용
- 인터프리터식 – 컴파일 필요 없음. 상호 반응형 환경 제공
- 객체지향적 – 모든 것이 객체! 더욱더 현실에 가까운 모델링 – 유지관리 용이
- 동적타이핑 – 데이터형은 실행되면서 결정됨. – 보다 더 쉬운 프로그래밍
- 함수형 – 입력과 출력으로 일관화함. 함수를 데이터처럼~
- **대규모 커뮤니티에서 컴퓨터 과학의 좋은 것들을 빠르게 도입함.**
  – 최신의 개념, 기술들이 다 있음.

파이썬으로 할 수 있는 일

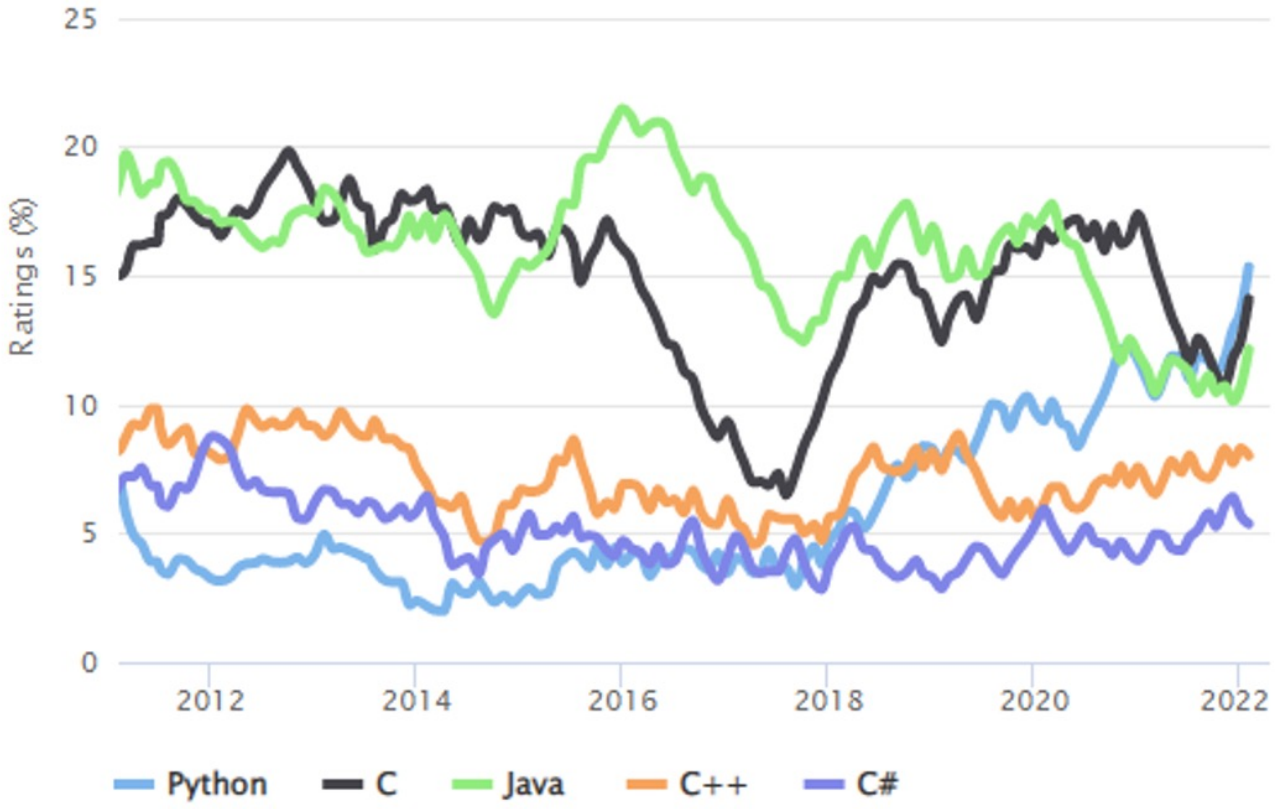


But, 대단히 빠른 속도나
하드웨어를 직접 제어하는
프로그램에는 어울리지 않음.

→ 필요하다면 C 확장!

Feb 2022	Feb 2021	Change	Programming Language	Ratings	Change
1	3	▲	Python	15.33%	+4.47%
2	1	▼	C	14.08%	-2.26%
3	2	▼	Java	12.13%	+0.84%
4	4		C++	8.01%	+1.13%
5	5		C#	5.37%	+0.93%
6	6		Visual Basic	5.23%	+0.90%
7	7		JavaScript	1.83%	-0.45%
8	8		PHP	1.79%	+0.04%
9	10	▲	Assembly language	1.60%	-0.06%
10	9	▼	SQL	1.55%	-0.18%

https://www.tiobe.com/tiobe-index/



Zen of Python

```
>>> import this
```

Beautiful is better than ugly.
못생긴 것보다 아름다운 것이 낫다.

Explicit is better than implicit.
암시적인 것보다 명시적인 것이 낫다.

Simple is better than complex.
복잡한 것보다 단순한 것이 더 낫다.

Complex is better than complicated.
난해한 것보다 복잡한 것이 더 낫다.

Readability counts.
프로그램은 읽기 쉬워야 한다.

Special cases aren't special enough to break the rules. Although practicality beats purity.
규칙을 깨야 할 정도로 특별한 경우는 없다. 비록 실용성이 이상을 능가한다 하더라도.

Errors should never pass silently. Unless explicitly silenced.
오류는 결코 조용히 지나가지 않는다. 알고도 침묵하지 않는 한.

In the face of ambiguity, refuse the temptation to guess..
모호함을 마주하고 추측하려는 유혹을 거절하라.

There should be one-- and preferably only one --obvious way to do it.
문제를 해결할 하나의 – 바람직하고 유일한 – 명백한 방법이 있을 것이다.

If the implementation is hard to explain, it's a bad idea.
설명하기 어려운 구현이라면 좋은 아이디어가 아니다.

Indentation (들여쓰기)

C 언어

```c
int factorial(int x)
{
    if(x == 0) {
        return 1;
    } else {
        return x * factorial(x - 1);
    }
}
```

파이썬

```python
def factorial(x):
    if x == 0:
        return 1
    else:
        return x * factorial(x - 1)
```

들여쓰기가 문법요소임!

({} 대신 들여쓰기)

파이썬에서 지원하는 다양한 자료형 (Data/Variable Types)

자료형	설명	비고
Boolean (불린)	True 혹은 False로 정의되는 데이터 유형	a = True
Numbers (숫자형)	정수, 실수, 분수, 복소수 등의 수로 표현되는 데이터 유형	a = 1.0
String (문자열)	문자로 이루어진 데이터 유형	a = 'Hello'
List (목록)	숫자, 문자, 목록 등을 순서대로 나열한 것	a = ['my', 1, 'one']
Tuple (튜플)	숫자, 문자, 목록 등을 순서대로 나열한 것이며 변경할 수 없음. (immutable)	a = ('my', 1, 'one')
Set (세트)	숫자, 문자, 목록 등을 순서 없이 모아 놓은 것	a = {1, 2, 3}
Dictionary (사전)	키-값의 형태로 순서 없이 모아 놓은 것	a = {'one': 1, 'two': 2}

Coding style

PEP 8 - Style Guide for Python Code

PEP:	8
Title:	Style Guide for Python Code
Author:	Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>
Status:	Active
Type:	Process
Created:	05-Jul-2001
Post-History:	05-Jul-2001, 01-Aug-2013

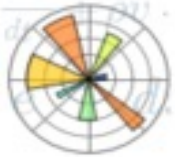
파이썬 코딩 스타일

- 일반변수는 lowercase_with_underbar
- 클래스는 CamelCase
- 들여쓰기는 스페이스 4칸

데이터 분석을 위한 파이썬 – Basic packages

NumPy
Base N-dimensional array package

SciPy library
Fundamental library for scientific computing



Matplotlib
Comprehensive 2D Plotting

IP[y]:
IPython

IPython
Enhanced Interactive Console

Sympy
Symbolic mathematics

pandas
Data structures & analysis

Basic packages for data analysis and visualization

파이썬으로 데이터분석한다 하면,

NumPy(C, Fortran으로 만들어진 고속자료구조) 위에 **pandas**(R data.frame 같은 편리한 인터페이스)를 올리고, 그 위에서 **SciPy**(C, Fortran으로 만들어진 과학 함수 모음)로 과학연산하고, **matplotlib**(MatLab 같은 차트 모듈)으로 가시화하며, **sckit-learn**
(기계학습 알고리즘 모음) 으로 예측 알고리즘 만들고, **statsmodels**(통계모델 모음)로 통계분석하고, 데이터를 **PyTables**(HDF5 지원 자료구조)에 저장하고 검색함.

여기에, 관계형 데이터베이스(Oracle, MySQL, SQLite)나 NoSQL 데이터베이스(MongoDB)를 쓰고, **Django**(웹프레임워크) 로 웹사이트 만들어 서비스

파이썬 실행

상호반응형 모드 (Interactive mode)

```
$ python3
Python 3.10.6 (main, Aug 30 2022, 05:12:36) [Clang 13.1.6 (clang-1316.0.21.2.5)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print(“Hello world”)
Hello world
```

프로그램 소스 코드를 .py 파일에 저장

```
$ cat mycode.py
print(“Hello world”)
$ python mycode.py
Hello world
```

Jupyter notebook

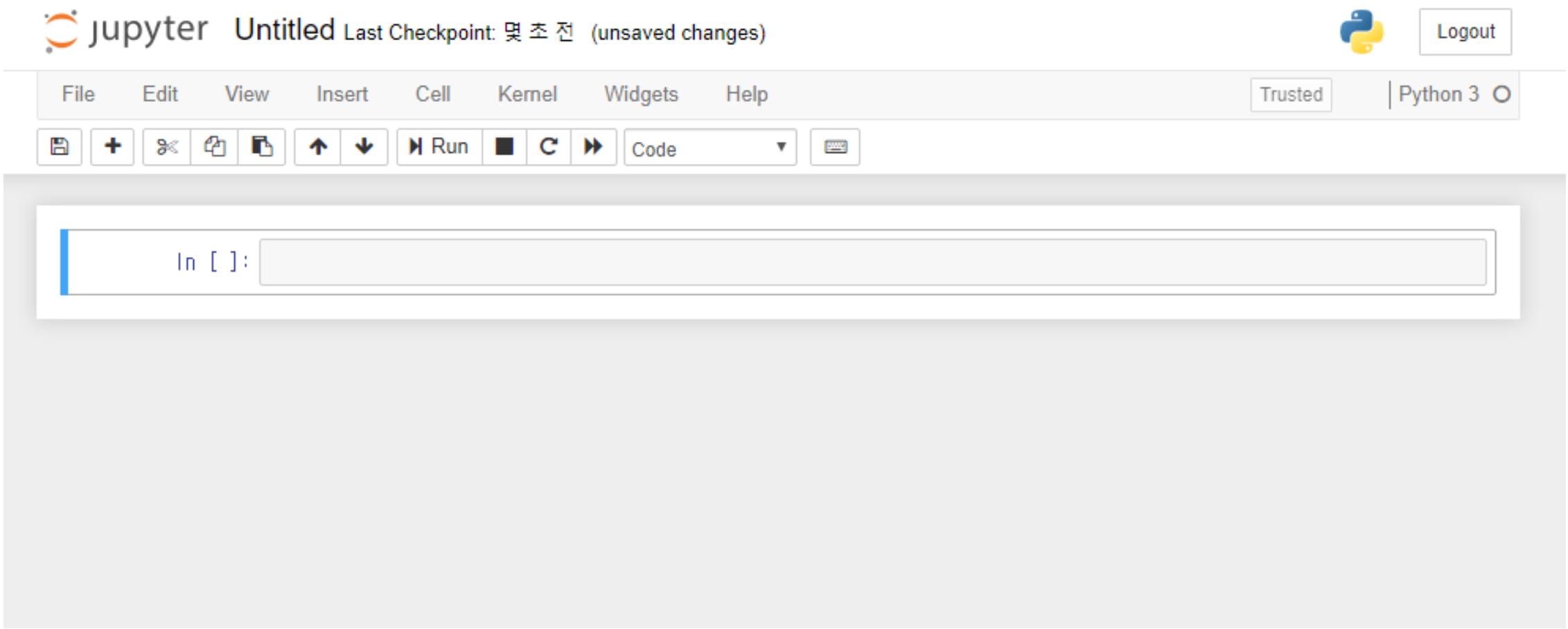
```
(inco) C:₩Users₩Eunchul jang>jupyter notebook
[I 19:42:11.311 NotebookApp] Serving notebooks from local directory: C:₩Users₩Eunchul jang
[I 19:42:11.311 NotebookApp] The Jupyter Notebook is running at:
[I 19:42:11.311 NotebookApp] http://localhost:8888/?token=ea987a30834c437cc6f84c7edd6cfbeb6271170ecfaf2481
[I 19:42:11.311 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 19:42:11.326 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=ea987a30834c437cc6f84c7edd6cfbeb6271170ecfaf2481
[I 19:42:11.658 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

Jupyter notebook

Python 3 선택

Jupyter notebook



Jupyter notebook 실행

파이썬 프로그래밍 준비 완료! Happy Python!

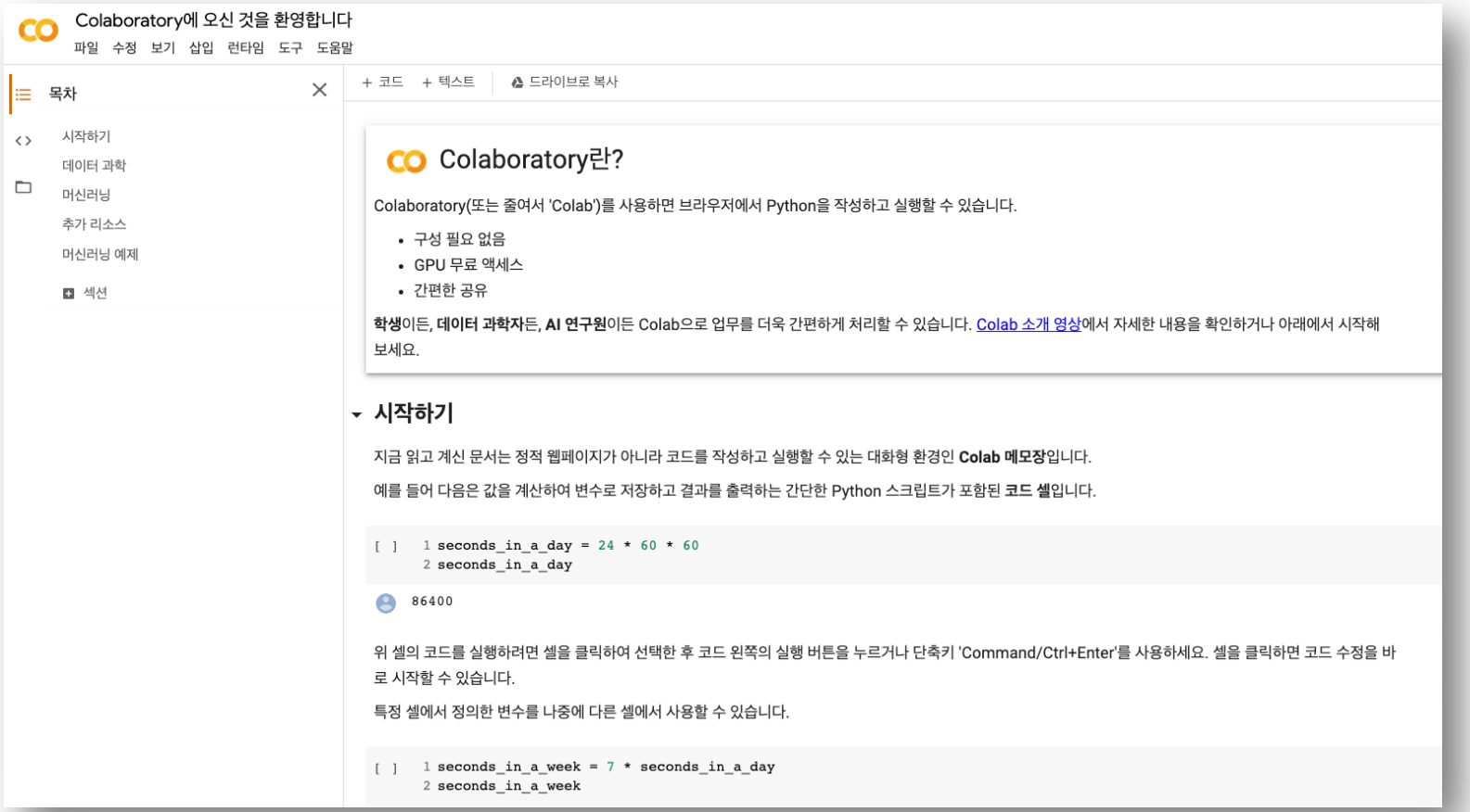
Google Colab!

인터넷환경이면 어디든 Jupyter 하세요~(장점)

- 소개 영상(https://youtu.be/inN8seMm7UI)
- 클라우드 기반의 무료 Jupyter
- 테블릿 PC, mobile에서도 코딩 가능
- 구글드라이브와 연동가능
- GPU와 TPU 이용 무료
  - GPU: Nvidia Tesla K80 혹은 Nvidia Tesla P100
  - GPU Memory: 12GB (K80) / 16GB (P100)

단점은..

- 세션 유지 **최대 12시간.**

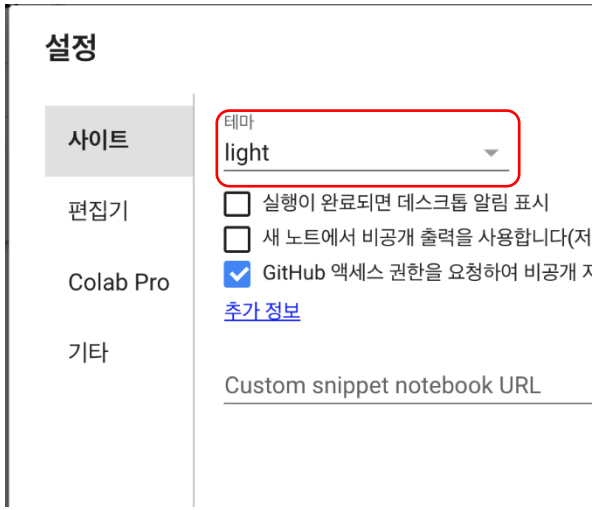


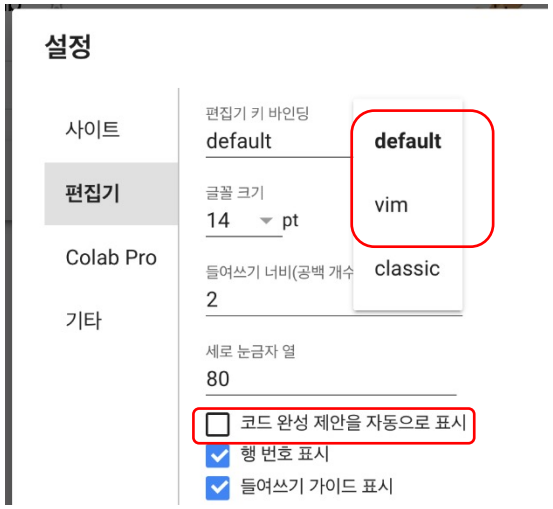
Colab 환경 설정

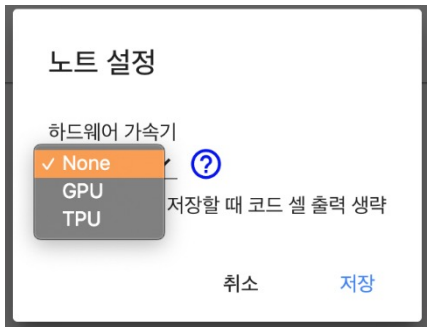
1. Colab 접속
https://colab.research.google.com/ 혹은 Google에서 'Colab'검색

2. [파일] - [새노트]
3. [도구] - [설정]
- 테마 변경 가능 : light / dark
- 편집기 설정 가능 : 기본 / vim
- ' 코드 완성 제안을 자동으로 표시' 해제

4. [런타임] - [런타임 유형 변경] - 'GPU / TPU'







03

소프트웨어 개발 방법론과 TDD (Test driven development)

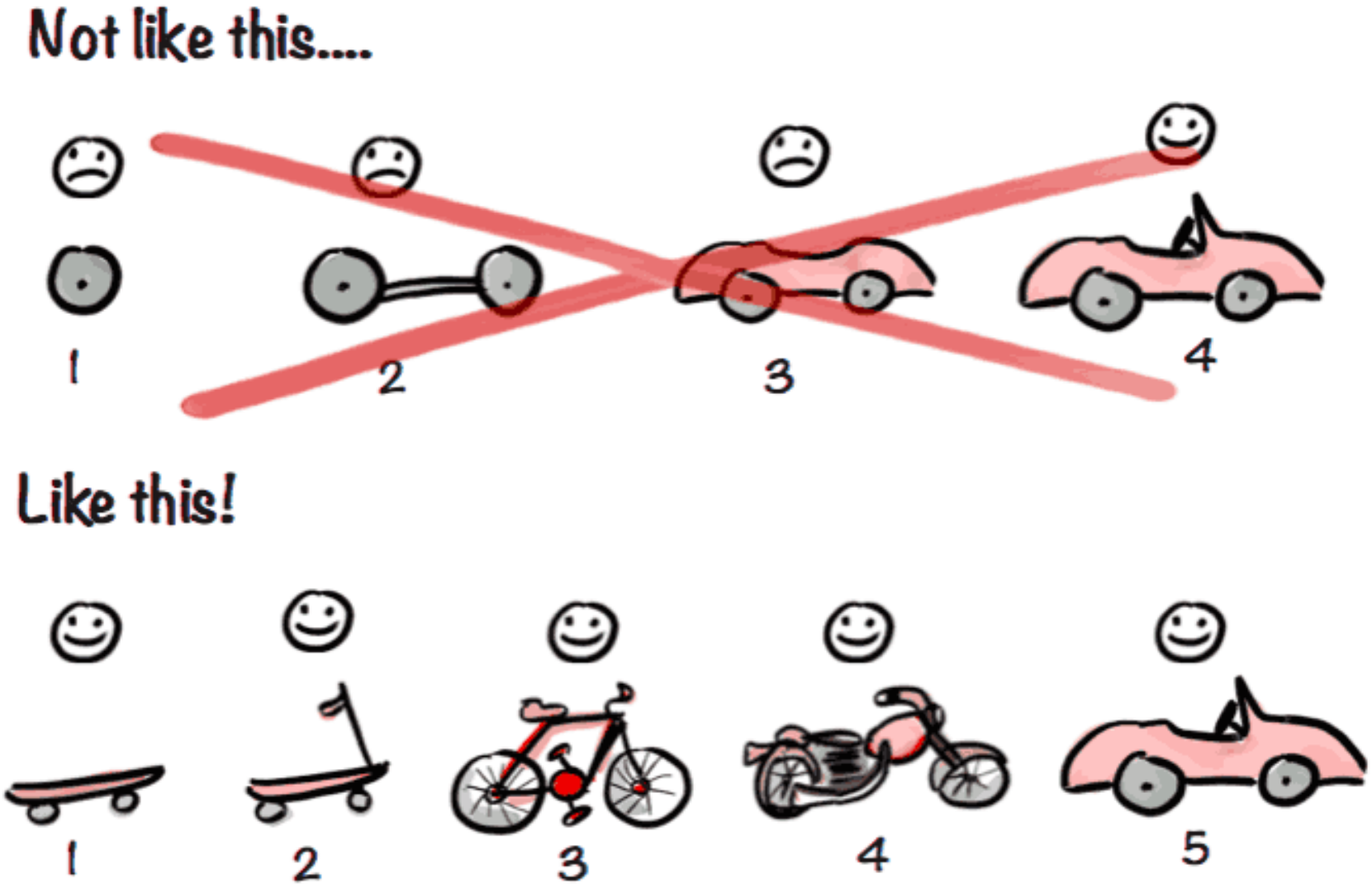

SECURITY
QUIET
HTML
php
git
LARACON
NODE
node

2 kinds of method

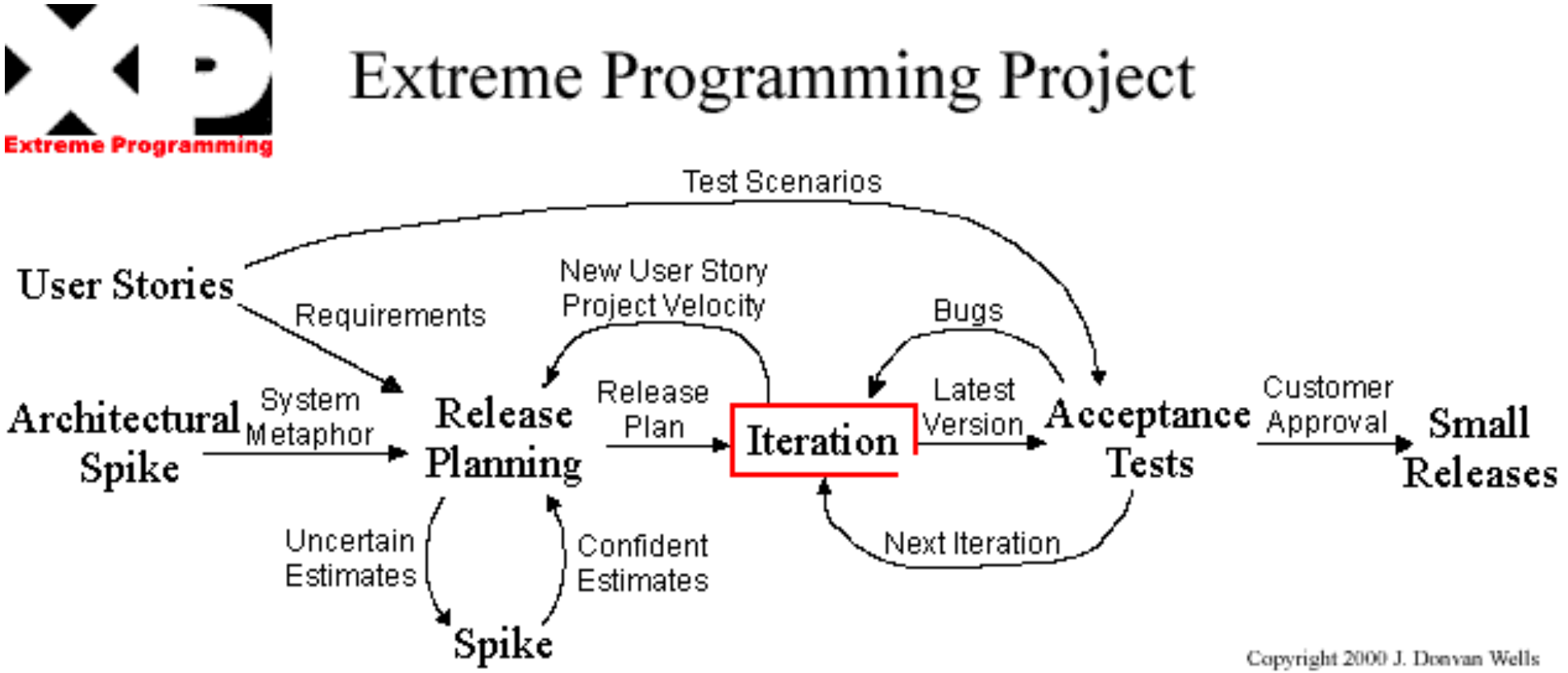
Big design up front

Small design

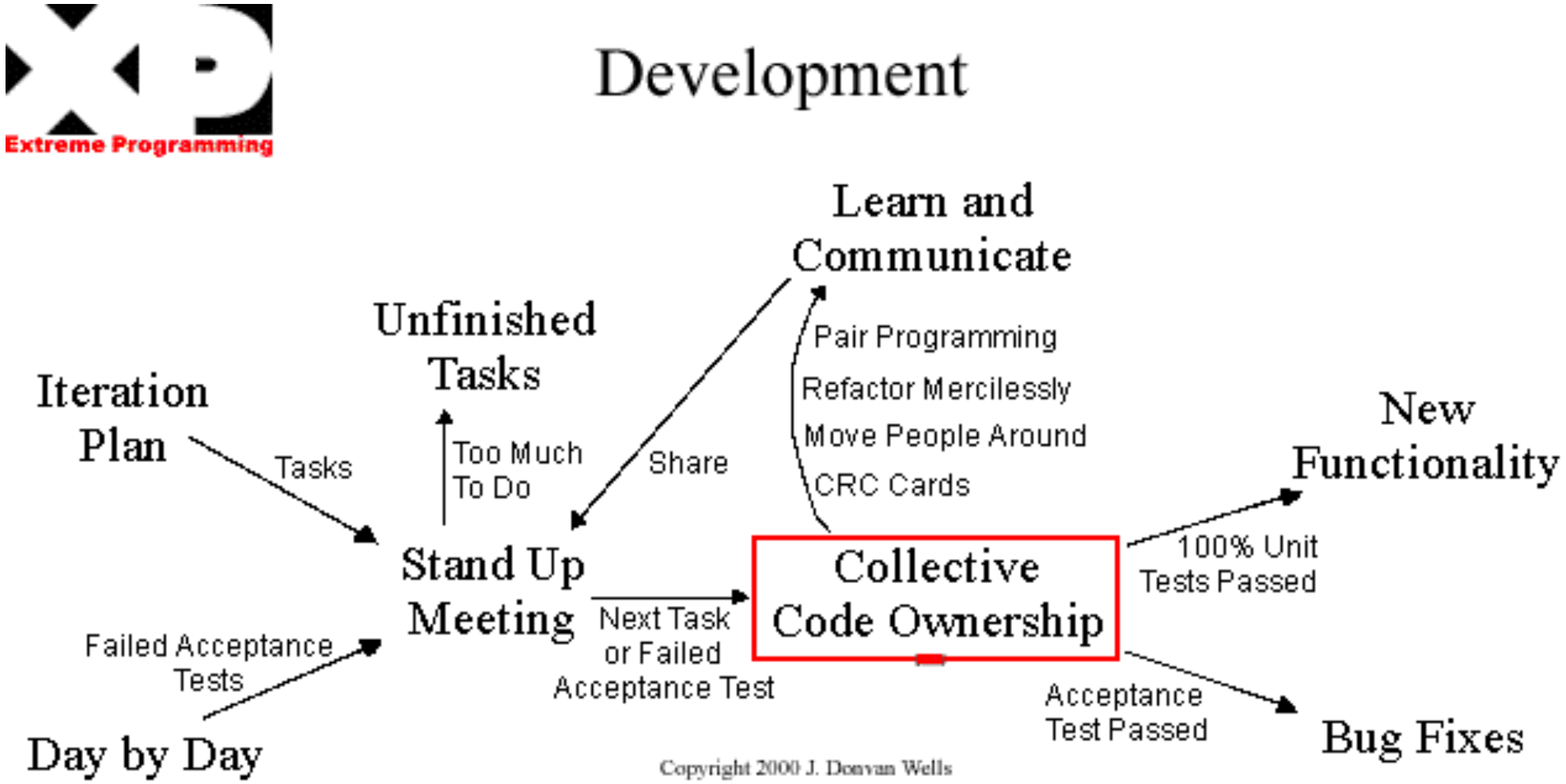
Agile software development



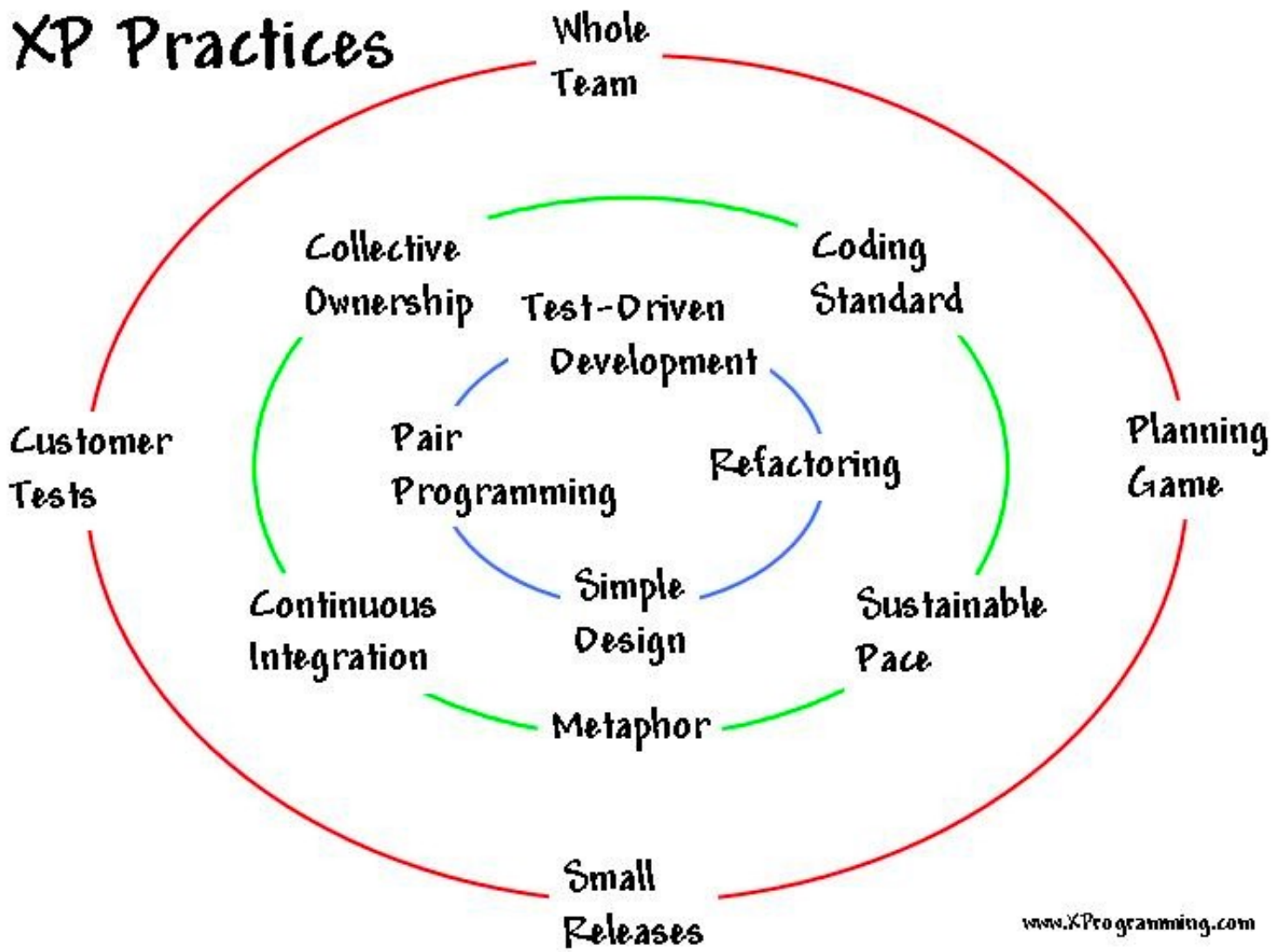
XP (eXtreme Programming)



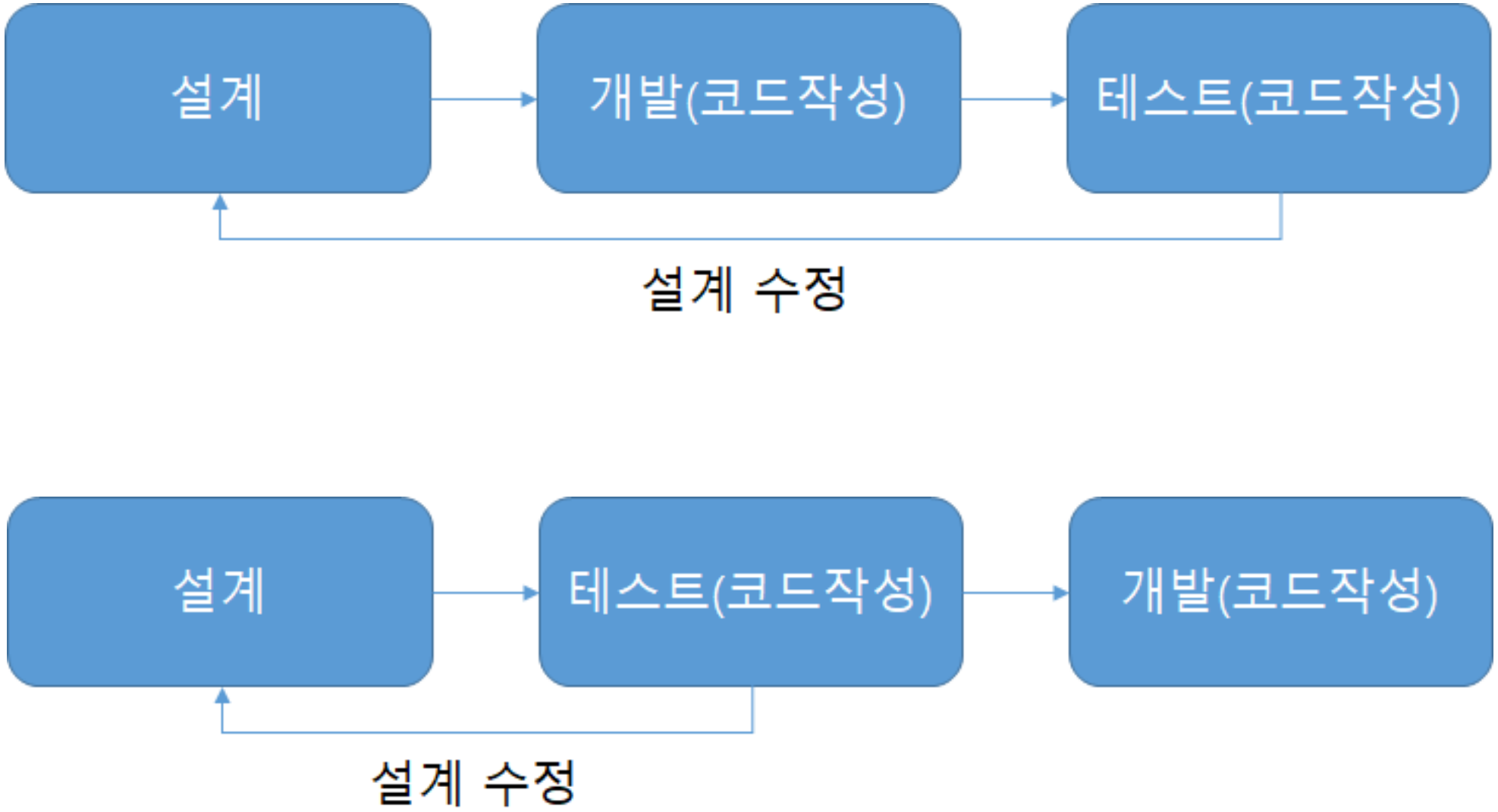
XP (eXtreme Programming)



XP Practice

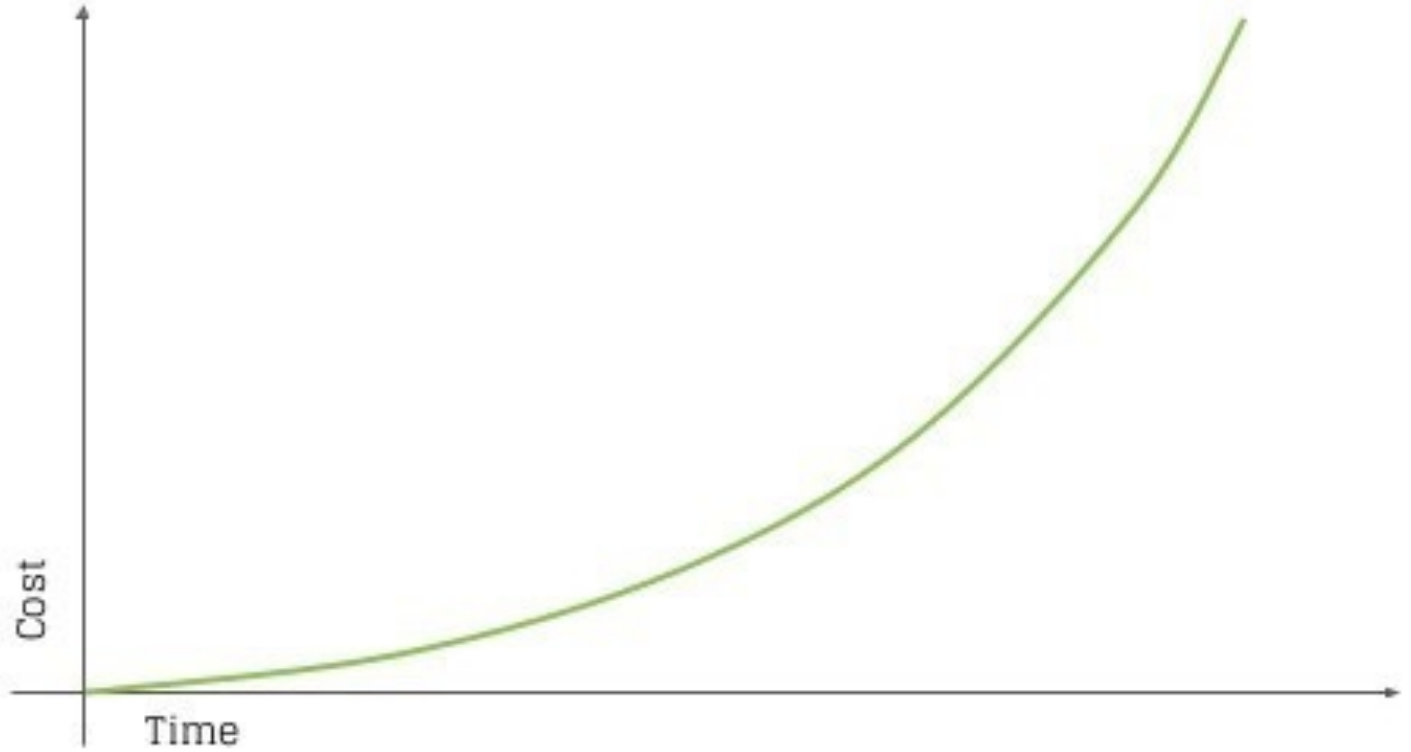


테스트 주도 개발

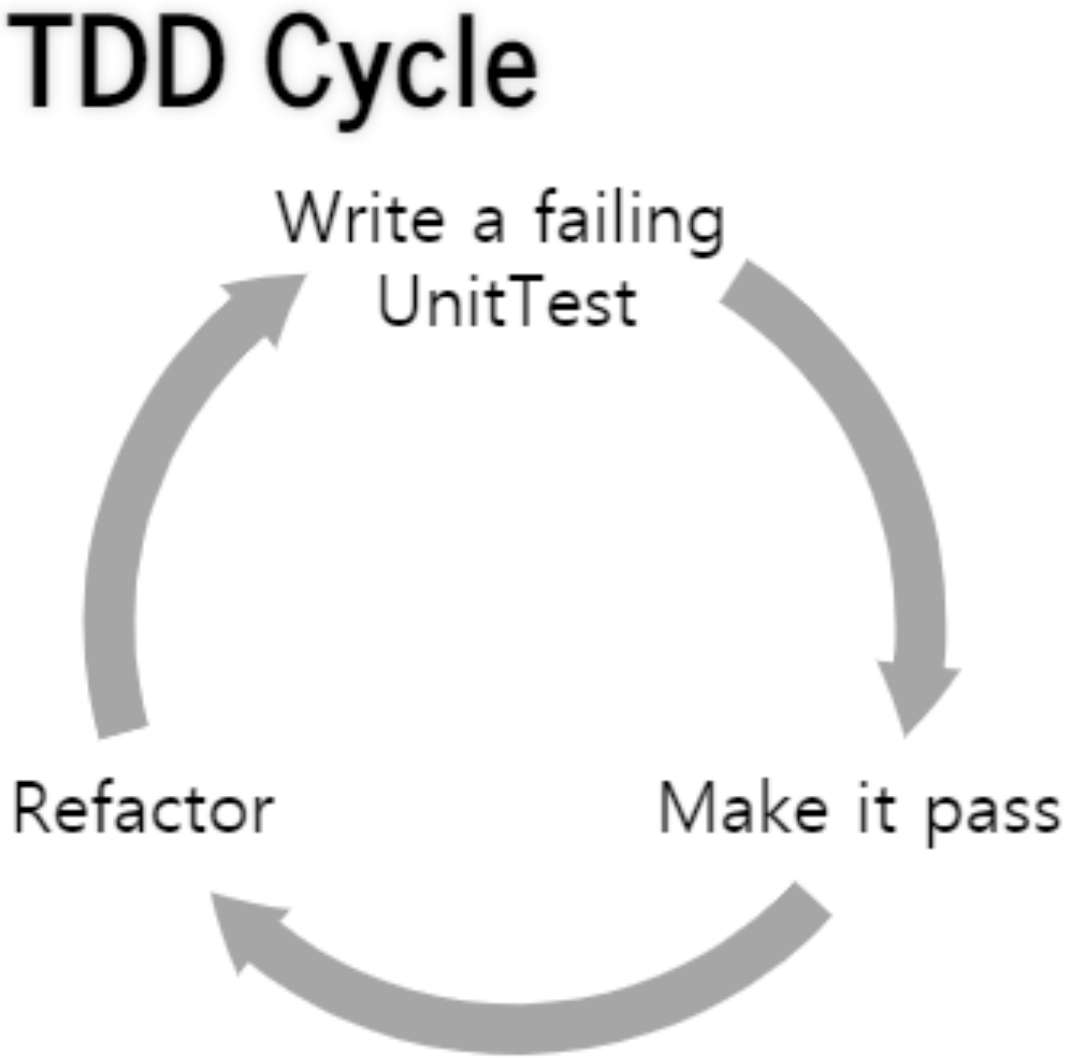


잘못 만들면 나중에 고치는 비용이 더…

결함은 일찍 찾을 수록 고치는 비용이 적게 든다



TDD 주기



TDD의 장점

- 테스트에는 실제 코드를 어떻게 사용하는지에 대해 작동하는 설명이 들어있다. (인터페이스가 정의된다.)
- 따로 테스트를 할 필요가 없다.
- 코드 수정 시 기존의 테스트 코드를 통과하는지 체크되기 때문에 통합적인 테스트가 유지된다.
- 테스트가 용이한 코드가 유지보수관리가 용이하다.
- 프로그램이 잘못되었는지를 "빨리" 알 수 있다(혹은 그럴 확률이 높다). (Fail early, often)
- 어떤 기능을 구현할 때, 어떻게 사용할지를 먼저 생각하도록 이끄는 역할을 한다. (Programming by intention)
- 오랜 시간이 지난 후에 다시 그 코드를 개선해야 할 일이 생길 때(혹은 어쨌던 봐야 할 일이 있을 때), 빨리 접근할 수 있도록 도와준다. (Documentation)

04

파이썬 객체지향 프로그래밍 (OOP, Object oriented programming)


AWN

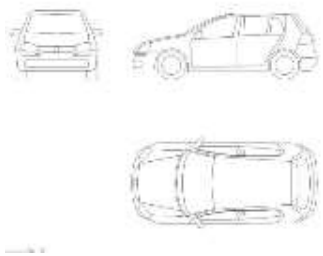
Object Oriented

- Programming paradigms
  - ✓Procedural programming
  - ✓Functional programming
  - ✓Object oriented programming
- Its origin is the modeling of cell
- Modeling of real world → Easy maintain
- **The keys** : remove duplication, easy management

Object oriented programming

- Object in Python is a representation of a person, a place, a bank account, a car, or any item that the program should handle.

- Object Oriented Programming Recipe:
  - (1) define **classes** (these are descriptions)
  - (2) **make object instances** out of classes.

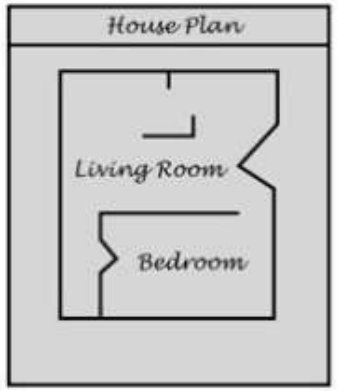


Class Car

Car instances

Class

Blueprint that describes a house



Instances of the house described by the blueprint

3 objects /
instances /
individuals

OOP with a Taxi Example

- To learn OOP, we will use an example of a Taxi.

Example Object - Taxi

Every object has two main components:

Data (the attributes about it)

Behavior (the methods)

DATA

- DriverName
- OnDuty
- NumPassenger
- Cities

BEHAVIOR

- PickUpPassenger
- DropOffPassenger
- SetDriverName
- GetDriverName

Taxi
• DriverName: **string** • OnDuty: **Boolean** • NumPassenger: **int** • Cities:**list**
• PickUpPassenger():**int** • DropOffPassenger()**: int** • SetDriverName(**string**) • GetDriverName:**string**

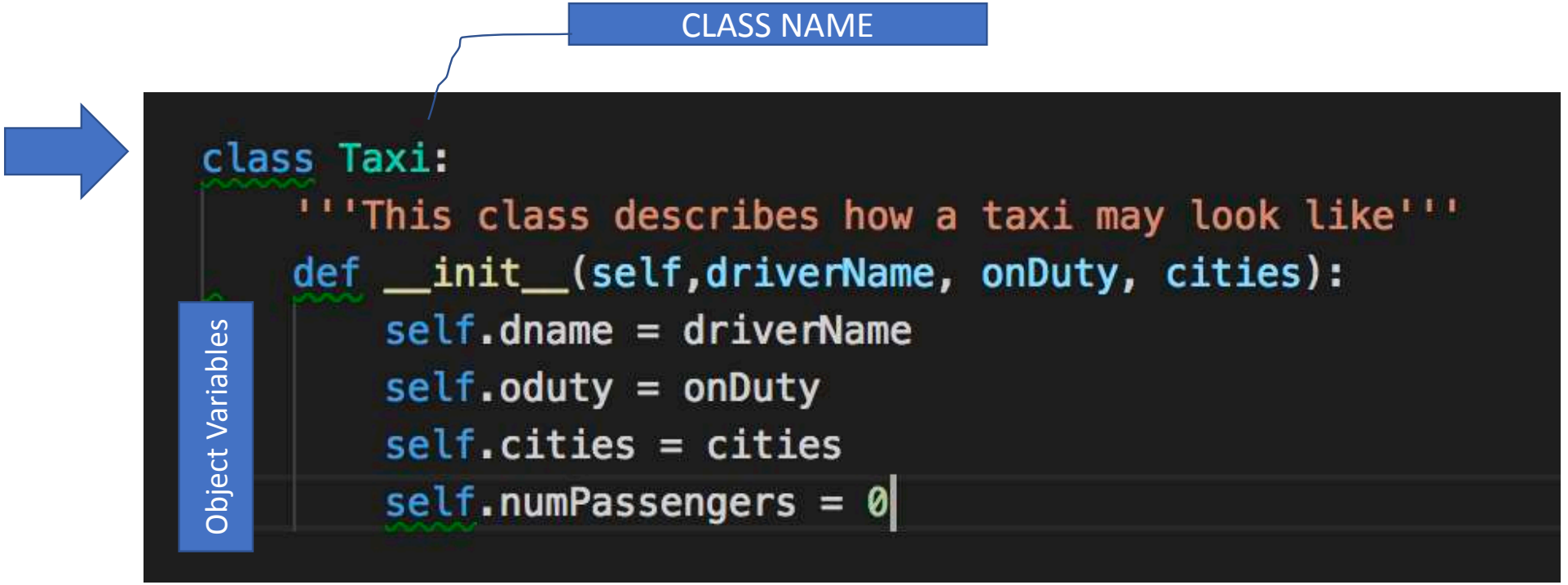
Creating a simple class in Python

- In a class, we describe (1) how the object will look like, and (2) what behavior it will have.

- Classes are the blueprints for a new data type in your program!

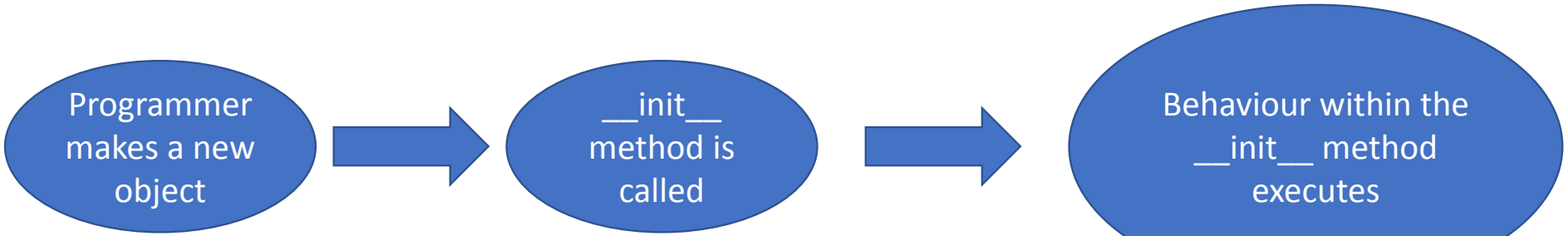
- A class should have at minimum*:
  - A name (e.g. **Class Taxi**)
  - A constructor method (that describes how to create a new object, **__init__)**

Example Class



The __init__ method

- __init__ is a special method in Python classes,
- The __init__ method is the constructor method for a class
- __init__ is called when ever an object of the class is constructed.

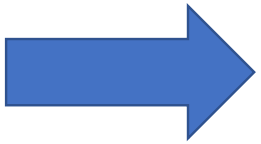


myTaxi = Taxi(“Aleks”)

…
Self.taxiDriver = “Aleks”
…

Creating an object from the class

- From one class, you make objects (instances).



```
class Taxi:
    '''This class describes how a taxi may look like'''
    def __init__(self,driverName, onDuty, cities):
        self.dname = driverName
        self.oduty = onDuty
        self.cities = cities
        self.numPassengers = 0


ourFirstTaxi = Taxi("Aleks", True, ['Lund', 'Malmo'])
print ourFirstTaxi.cities
```



```
class Taxi:
    '''This class describes how a taxi may look like'''
    def __init__(self, driverName, onDuty, cities):
        self.dname = driverName
        self.oduty = onDuty
        self.cities = cities
        self.numPassengers = 0

    def setDriverName(self, driverName):
        self.dname = driverName

ourFirstTaxi = Taxi("Aleks", True, ['Lund', 'Malmo'])
print(ourFirstTaxi.cities)
ourFirstTaxi.setDriverName('Micheal')
print(ourFirstTaxi.dname)
```

```
class Taxi:
    '''This class describes how a taxi may look like'''
    def __init__(self, driver_name, on_duty, cities):
        self.driver_name = driver_name
        self.on_duty = on_duty
        self.cities = cities
        self.num_passengers = 0

    def set_driver_name(self, driver_name):
        self.driver_name = driver_name

our_first_taxi = Taxi("Aleks", True, ['Lund', 'Malmo'])
print(our_first_taxi.cities)
our_first_taxi.set_driver_name('Micheal')
print(our_first_taxi.driver_name)
```


Another example – calculate average

Name	Korean	English	Math	Science
smith	80	69	70	88
neo	92	66	80	72
trinity	82	73	91	90
oracle	80	42	100	92

Procedural example

```
smith_korean = 80
smith_english = 69
smith_math = 70
smith_science = 88

neo_korean = 92
neo_english = 66
neo_math = 80
neo_science = 88

smith_average = (
        smith_korean + smith_english + smith_math + smith_science) / 4.0
neo_average = (
        neo_korean + neo_english + neo_math + neo_science) / 4.0
```


Functional example

```
def average(alist):
    return sum(alist) / float(len(alist))

smith = {
    'korean': 80,
    'english': 69,
    'math': 70,
    'science': 88,
}
neo = {
    'korean': 92,
    'english': 66,
    'math': 80,
    'science': 88,
}

smith_average = average(smith.values())
neo_average = average(neo.values())
```

Object oriented example

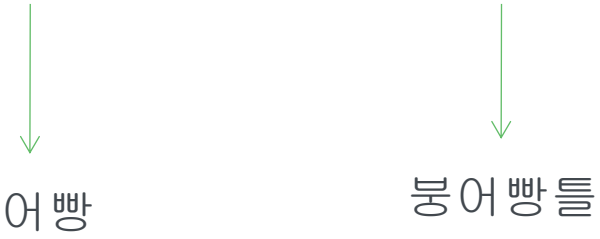
```
class Score:
    def __init__(self, korea, english, math, science):
        self.korea = korea
        self.english = english
        self.math = math
        self.science = science
    def get_average(self):
        return (self.korea + self.english + self.math + self.science) / 4.0

smith_score = Score(80, 69, 70, 88)
smith_average = smith_score.get_average()
neo_score = Score(92, 66, 80, 88)
neo_average = neo_score.get_average()
```

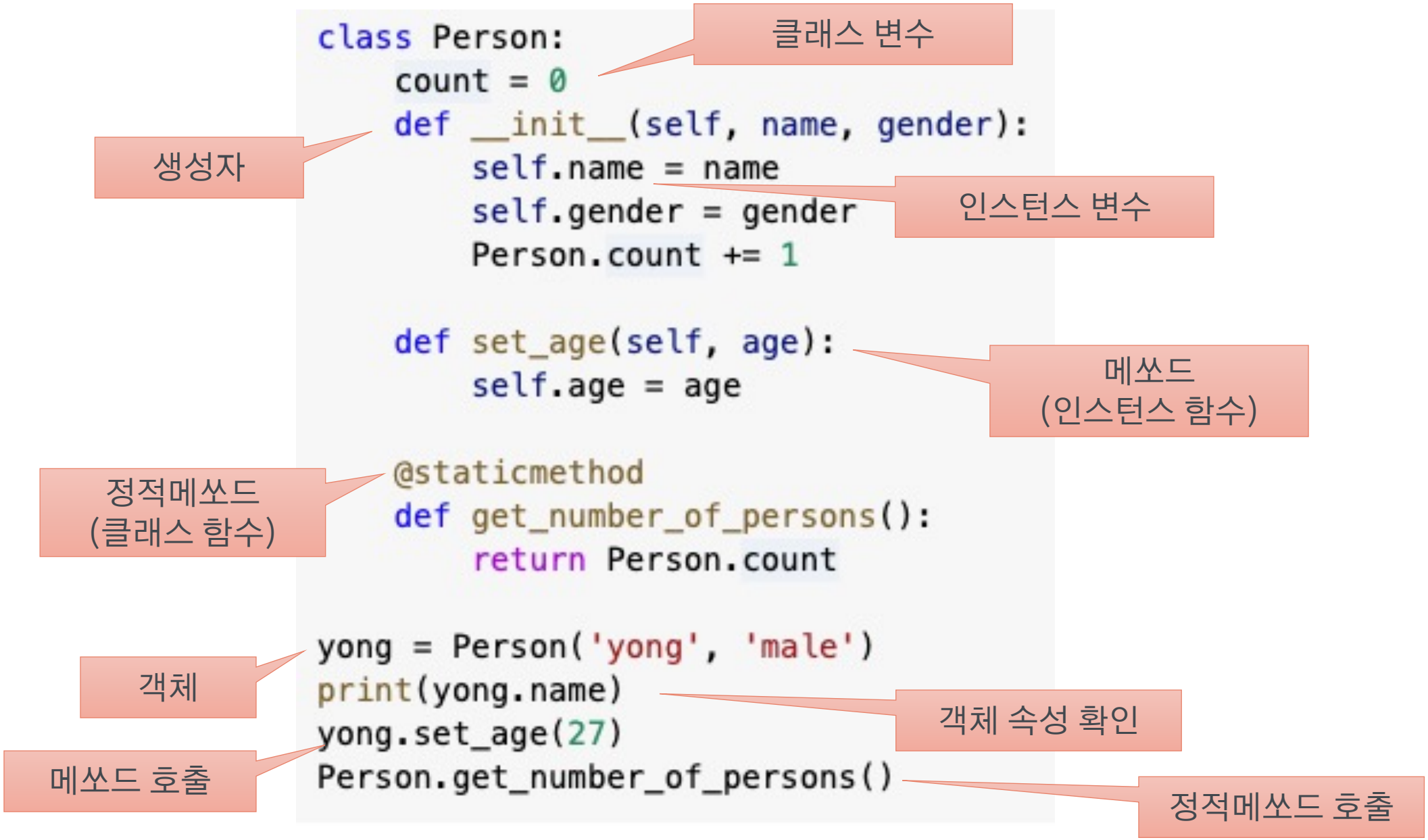
Instance(object) and Class

```
smith_score = Score(80, 69, 70, 88)
```

```
object = Class()
```



클래스의 구성



상속 예제 1

```python
class Person:
    count = 0
    def __init__(self, name, gender):
        self.name = name
        self.gender = gender
        Person.count += 1

    def set_age(self, age):
        self.age = age

    @staticmethod
    def get_number_of_persons():
        return Person.count

yong = Person('yong', 'male')
print(yong.name)
yong.set_age(27)
Person.get_number_of_persons()
```

```python
class Student(Person):
    def sing_a_song(self):
        print("숨 참고 Love dive~")

hyuna = Student('hyuna', 'female')
hyuna.set_age(11)
hyuna.sing_a_song()


class Childern(Person):
    def sing_a_song(self):
        print("송아지 송아지~")

minji = Childern('minji', 'female')
minji.set_age(6)
minji.sing_a_song()
```

상속 예제 2

택시 요금 함수 추가

```
class Taxi:
    '''This class describes how a taxi may look like'''
    def __init__(self, driver_name, on_duty, cities):
        self.driver_name = driver_name
        self.on_duty = on_duty
        self.cities = cities
        self.num_passengers = 0

    def set_driver_name(self, driver_name):
        self.driver_name = driver_name

    def get_the_fee(self, distance):
        return distance * 500   # km 당 500원
```

만일 모범택시 요금은 km 당 700원,
전기 택시 요금은 400원이라면?

상속 예제 2

```
class Taxi:
    '''This class describes how a taxi may look like'''
    def __init__(self, driver_name, on_duty, cities, type_='general'):
        self.driver_name = driver_name
        self.on_duty = on_duty
        self.cities = cities
        self.num_passengers = 0
        self.type_ = type_   # (general, mobum, electric)

    def set_driver_name(self, driver_name):
        self.driver_name = driver_name

    def get_the_fee(self, distance):
        if self.type_ == 'general':
            unit_price = 500
        elif self.type_ == 'mobum':
            unit_price = 700
        elif self.type_ == 'electric':
            unit_price = 400
        return distance * 500   # km 당 500원

taxi_a = Taxi('Yong', True, ['Seoul', 'Anyang'])
taxi_b = Taxi('Gunhee', True, ['Anyang'], type_='mobum')
taxi_c = Taxi('Choi', True, ['Suwon'], type_='electric')

taxi_a.get_the_fee(20)
taxi_b.get_the_fee(20)
taxi_c.get_the_fee(20)
```

```
class Taxi:
    '''This class describes how a taxi may look like'''
    def __init__(self, driver_name, on_duty, cities):
        self.driver_name = driver_name
        self.on_duty = on_duty
        self.cities = cities
        self.num_passengers = 0

    def set_driver_name(self, driver_name):
        self.driver_name = driver_name

    def get_the_fee(self, distance):
        return distance * 500

class MobumTaxi(Taxi):
    def get_the_fee(self, distance):
        return distance * 700

class ElectricTaxi(Taxi):
    def get_the_fee(self, distance):
        return distance * 400

taxi_a = Taxi('Yong', True, ['Seoul', 'Anyang'])
taxi_b = MobumTaxi('Gunhee', True, ['Anyang'])
taxi_c = ElectricTaxi('Choi', True, ['Suwon'])

taxi_a.get_the_fee(20)
taxi_b.get_the_fee(20)
taxi_c.get_the_fee(20)
```


05

TDD로 OOP하기 실습

FASTA 형식의 DNA 자료 변환 – fasta example

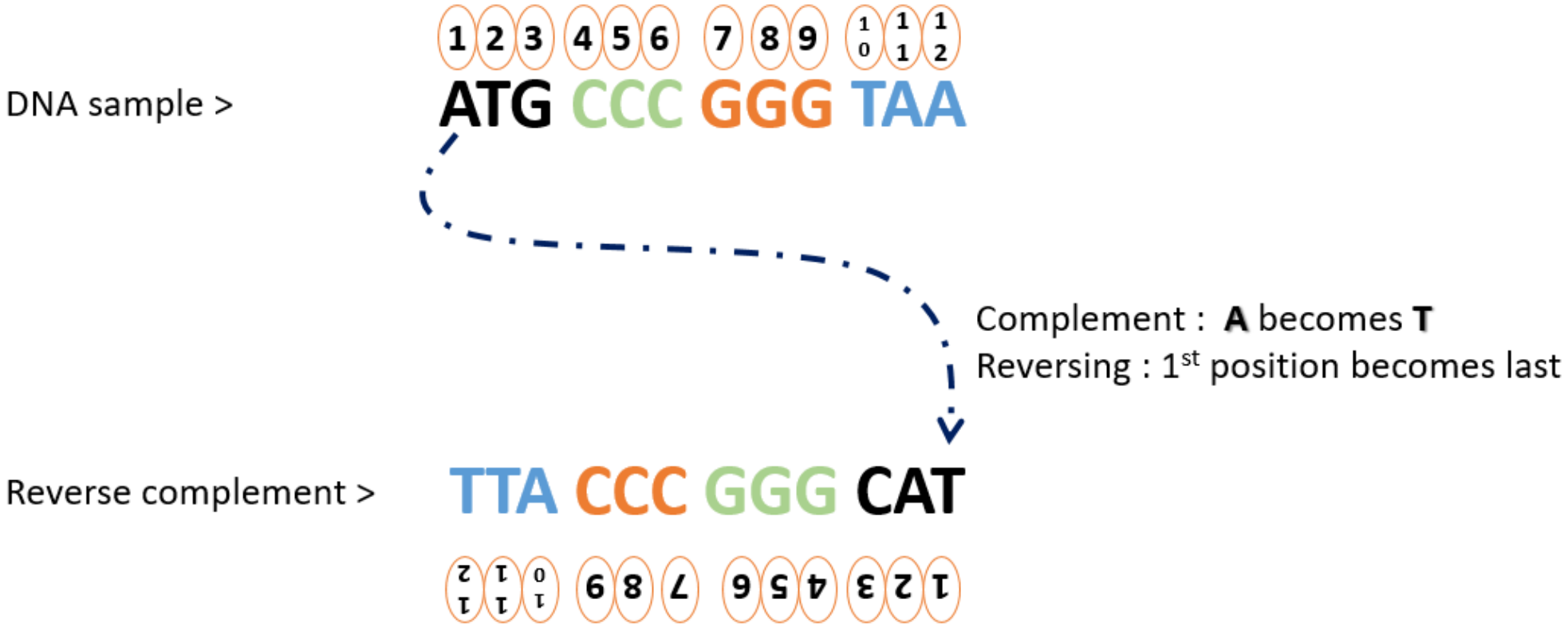
>AJ131283.1 Sabella spallanzanii mRNA for globin 1
GCTCGAAGAATCACCAAGGCACATCAGAAAGTGTACCATCCGACTGGAAGACCATGTTTCGTTTTGCTCT
TCTGTGTGCTTTTGTTGCCGATGCCAGTGCAGAAGGATGCTGCTCCATGGAGGACAGACAGGAGGTTCTC
AATGCCTGGGAGGCTCTGTGGAGTGCTGAGTACACTGGCAGGAGAGTCATGATCGCCCAGGCAGCGTTTC
AAAAACTGTTTGAGAAGGCCCCCGATTCCAAGGCATTGTTTACCCGCGTCAACGTTGATAACATCGGAAG
CCCCCAATTCCGAGCTCACTGTATCCGTGTTACCAATGGTTTCGACACCATCATCAACATGGCCTTTGAC
ACTGATGTTCTGGAAGAGCTACTGACCCATCTGGGCAACCAACACACGAAGTATCAAGGCATGAGAGCTG
CGTACTTGACGCATTTCCGTGAATCTTTCGCTGAGATCTTGCCCCAGGCTATTCCGTGCTTCAACACCGC

>M63452.1 Bovine gamma globin gene and globin (PSI-2) pseudogene, complete cds
GGAGAGTAGAGTTTCTGAGTTTAGACACACTGAATCAGCCAATCACAGATGAAGAGCACTGAGCAACAAG
AGTTCATCTTACATTCCCCCAAACCAATGAACTTGTATTATGCCCTGGGCTAATCTGCTCTCAGAAGCAG
GGAGGGCAGGAGGCTGGGTGGGGCTCACAAGGAAGACCAGGGCCCCTACTGCTTACACATGCTTTTGACA
CAACTTGCAGCTGCACAAACACACATCATGGTGTATCTGACTCTTGAGAAGAAGGCTACTGTCATTGACT
TGTGGAGTAAGATGAGGGTGGCTGAAGTTGGTCCGGATACTGTAGGCAGGCAGGTATTCAACTTACAAGG
CAGGCTGAAGGAGAGTGAATGTCAGTTGGGTGTGTGGGGACAGAGCCATTGCCTGAGATTCTGGCAGGCA
CTGACTCCCTCTGACCTTGTGCTGTTTTCACCCCCTTTGCTGCTGGTCGTCTACCCCTCGACTCAGAGGT
TCTTTGACTATTGTGGGGACTGTCCTTTGCTGATTTATGGGCAATGTTTTACCTTCTTTGTTTCCAGGCA
TAGTTTCCTCTTATTCATTCTTGTTTTTCTCTGTTTGTTCTCTGCAGTATCTTCTTTATATTTAAACATT

>M63453.1 Bovine Beta globin gene and globin (PSI-3) pseudogene, complete cds
GGAGAATAAAGTTTCTGAGTCTAGACACACTGGATCAGCCAATCACAGATGAAGGGCACTGAGGAACAGG
AGTGCATCTTACATTCCCCCAAACCAATGAACTTGTATTATGCCCTGGGCTAATCTGCTCAGAGCAGAGA
GGGCAGGGGGCTGGGTGGGGCTCACAAGCAAGACCAGGGCCCCTACTGCTTACACTTGCTTCTAACACAA
CTTGCAACTGCACAAACACACATCATGGTGCATCTGACTCTTGAGGGGAAGGCTACTTGTCACTGCCCTG
CAGACGAAAATGAGGGTGGCTGAAGTTGGTGTTGAAACCTTAGGCAGGCAGGTATTCAGCTTACAAGGCA
AGGAGAGTGAATGTCAGCTGGGTGTGTGGGGACAGAGCCATTGCCTGGGATTCTGGCAGGCATTGACTCC

이 파일을 읽어서
reverse complement
서열을 출력하기

FASTA 형식의 DNA 자료 변환 – reverse complement



실무에 쓸 일이 있으면 라이브러리를 찾아서…


```python
from Bio import SeqIO

with open('my.fasta') as afile:
    for record in SeqIO.parse(afile, 'fasta'):
        rc_record = record.reverse_complement()
        rc_record.id = record.id
        print(rc_record.format('fasta'))
```


Thank you!

Contact Info.

OFFICE	경기도 용인시 기흥구 흥덕1로 13 흥덕IT밸리 타워 A동 2901-2903호
EMAIL	info@insilicogen.com
PHONE #	031 278 0061
FAX	031 278 0062