hphk.

4차 산업혁명
소프트웨어와 프로그래밍

김재석
(주식회사 해피해킹)

Happy Hacking

교육일정

	내용	키워드	실습
Session 1	Hello, World!	소개 및 교육 안내	
Session 2	Hack Your Life	Life hacking 오픈소스	Python 챗봇 맛보기
Session 3	Python 기초	프로그래밍 언어 Python 기초 & API	챗봇 업그레이드
Session 4	Python 심화	Work hacking	웹스크래핑
Session 5	MarkDown	MarkDown	학습 내용 정리

Python
Web - HTML/CSS
Framework - flask/Django
DB - SQL - ORM
Git, Amazon Web Services

+프로그래밍, SW 학습방법

+프로젝트, 포트폴리오 관리

1 WEEK.

1 WEEK.
한 학기 미리 보기

Intro to Python/Git/HTML, CSS

Keyword based Chatbot
Telegram Chatbot
인생의 첫 배포.

마지막으로,
한 가지 부탁드리고 싶은건

이번 한주는 ‘재미’를 위한 강의지,
여러분들에게 ‘좌절’을 선물(?)하는
강의가 아니라는 점!

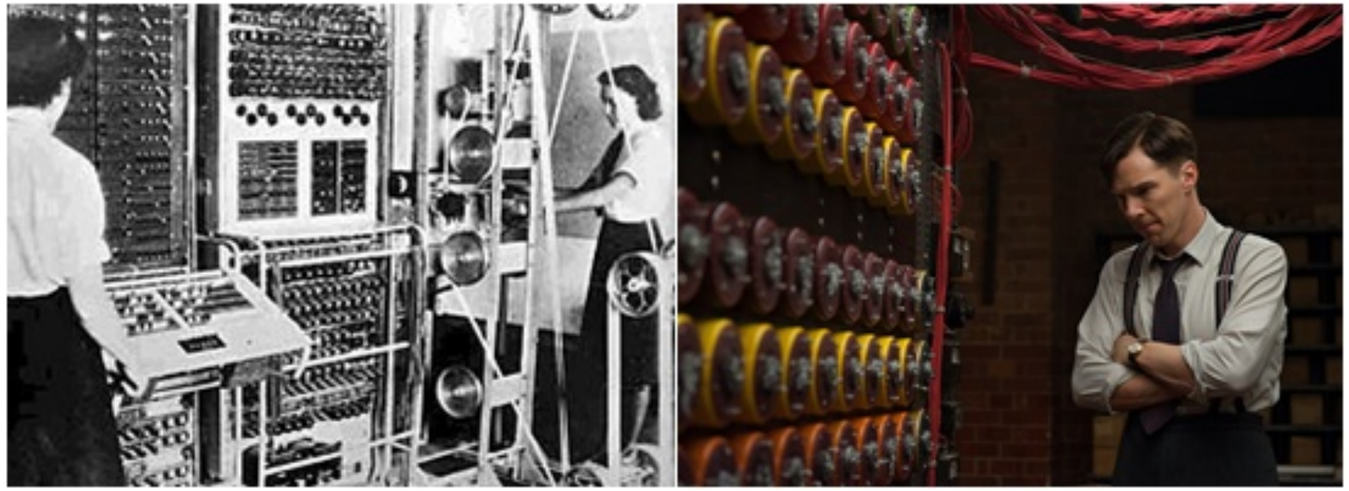
그냥 ‘잼잼’ 하게 즐겼으면 좋겠어요!

Session 0.
Ice breaking

Computational Thinking?

Computer Programming?

Computer?



계산

저장

프로그램?

프로그램?

컴퓨터 프로그램(영국 영어: computer programme, 미국 영어: computer program, 문화어: 콤퓨터프로그람)은 컴퓨터에 의해 실행되는 지시사항의 모음인 컴퓨터 소프트웨어의 한 예이다. 대부분의 프로그램은 실행 중(즉, 명령어를 '불러들일' 때)에 사용자의 입력에 반응하도록 구현된 명령어의 집합으로 구성되어 있다. 대부분의 프로그램들은 하드디스크 등의 매체에 바이너리 형식의 파일로 저장되어 있다가 사용자가 실행시키면 메모리로 적재되어 실행된다.

출처 : 위키피디아

프로그램?

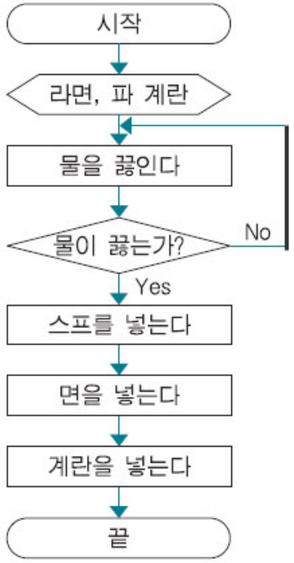
컴퓨터 프로그램(영국 영어: computer programme, 미국 영어: computer program, 문화어: 콤퓨터프로그람)은 컴퓨터에 의해 실행되는 지시사항의 모음인 컴퓨터 소프트웨어의 한 예이다. 대부분의 프로그램은 실행 중(즉, 명령어를 '불러들일' 때)에 사용자의 입력에 반응하도록 구현된 명령어의 집합으로 구성되어 있다. 대부분의 프로그램들은 하드디스크 등의 매체에 바이너리 형식의 파일로 저장되어 있다가 사용자가 실행시키면 메모리로 적재되어 실행된다.

출처 : 위키피디아

Programming
== 지시, 명령
== 시키는 것

컴퓨터 프로그래밍

== 컴퓨터에게 뭘 시키는 것



프로그래밍의 모든 것

이미 배웠음

But, 생각보다 쉽진 않다

Why?

Output - compiler (run)

```
run:
Success: true
java.lang.ClassNotFoundException: HelloWorld
        at java.net.URLClassLoader$1.run(URLClassLoader.java:372)
        at java.net.URLClassLoader$1.run(URLClassLoader.java:361)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:360)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:260)
        at CompileSourceInMemory.main(CompileSourceInMemory.java:50)
BUILD SUCCESSFUL (total time: 2 seconds)
```

Python Shell

File Edit Shell Debug Options Windows Help

```
>>> ============= RESTART ===============
>>>
Traceback (most recent call last):
  File "C:/MyCode/runtime.py", line 2, in <module>
    print( titel )
NameError: name 'titel' is not defined
>>>
```

컴퓨터가 멍청해서

‘내가 명령을 못해서’

Declarative Knowledge

vs

Imperative Knowledge

Declarative Knowledge

“What”
사실에 대한 정보

“서울은 한국의 수도다”

“삼성은
국내 최고의 기업이다”

“나는 SSAFY에 왔다”

Imperative Knowledge

“How to”
방법에 대한 정보

“코끼리를 냉장고에 넣는 법”

“코끼리 클래스와 냉장고 클래스를 만들어 코끼리를 Private Inheritance 한다. 그렇다면 has-a 관계가 성립되어 코끼리가 냉장고에 들어간다.”

- 컴퓨터공학과

“인턴한테 시켜라.”

- 의예과

“냉장고에 코끼리를 넣지 않았지만 우리는 이미 넣은 것이나 다름없다.”

- 철학과

“내가 강의실에 오는 법”

모든 단계가
잘 정의되어야 한다.

Activity 1.

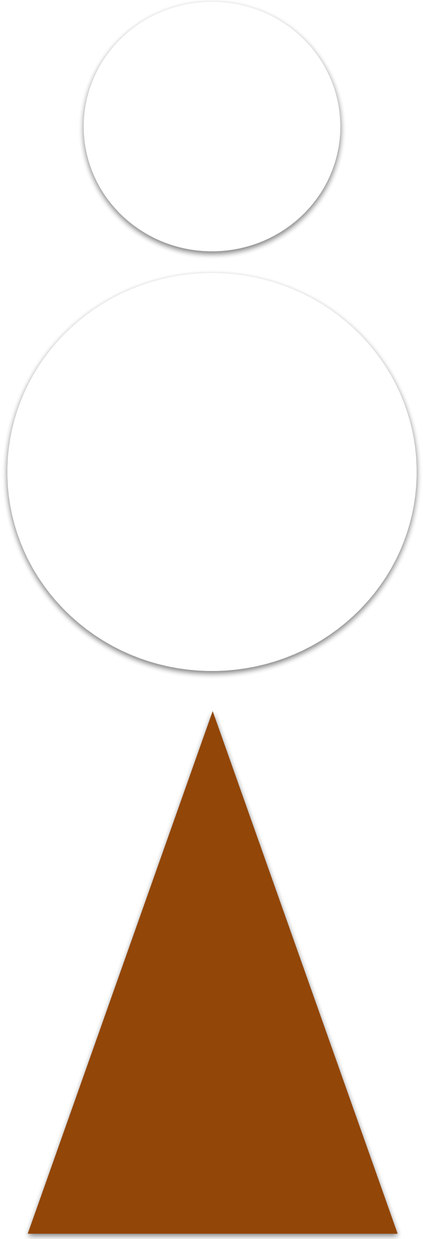
그림그리기

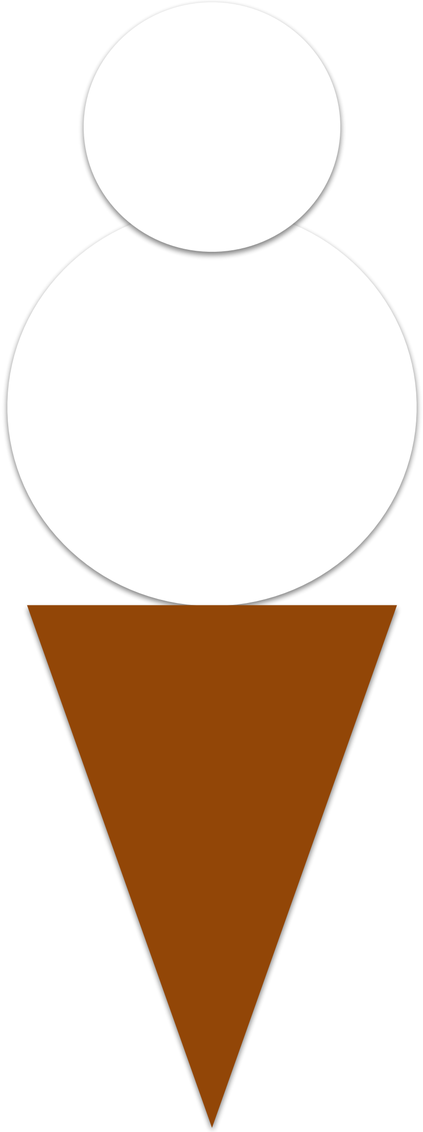
1. 종이에 동그라미를 그리세요

2. 그 아래에 또 하나의 동그라미를 그리세요

3. 세모를 그리세요

?





Activity 2.

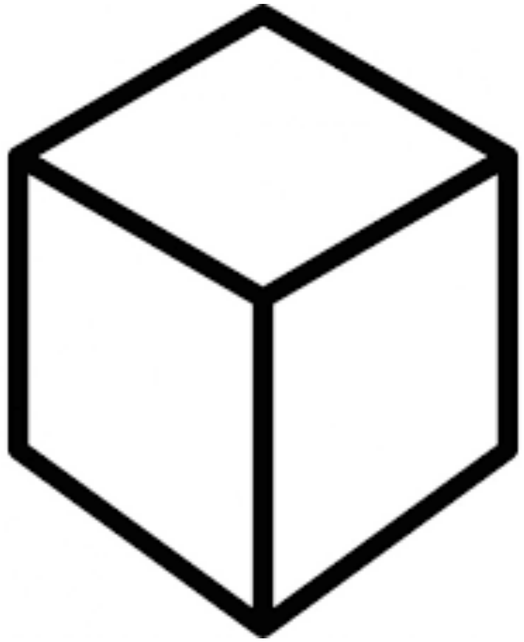
지원자

지원자에게는
스타벅스 기프트 카드 줄게여 :)

?

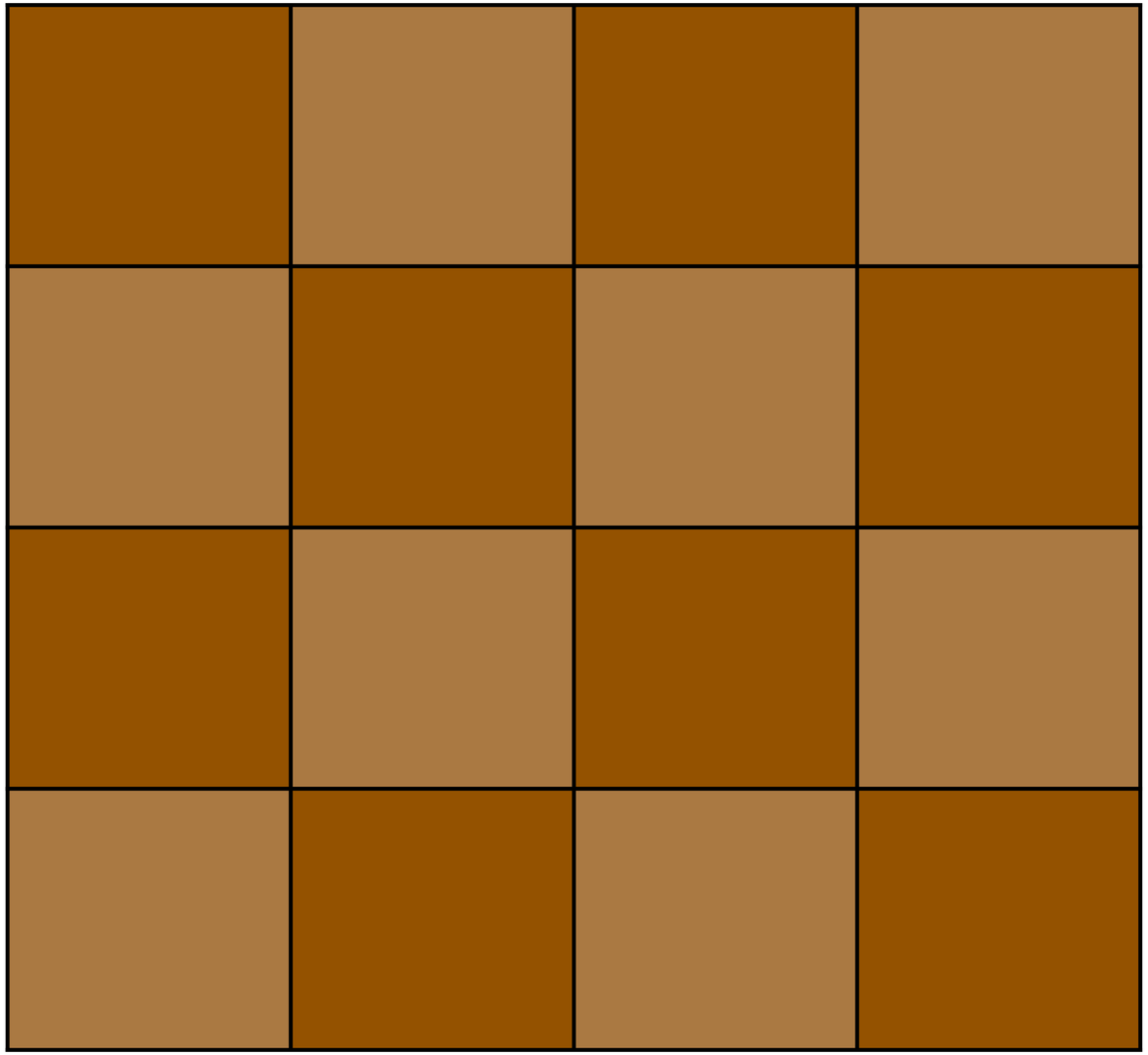
점, 선, 방향으로만 지시

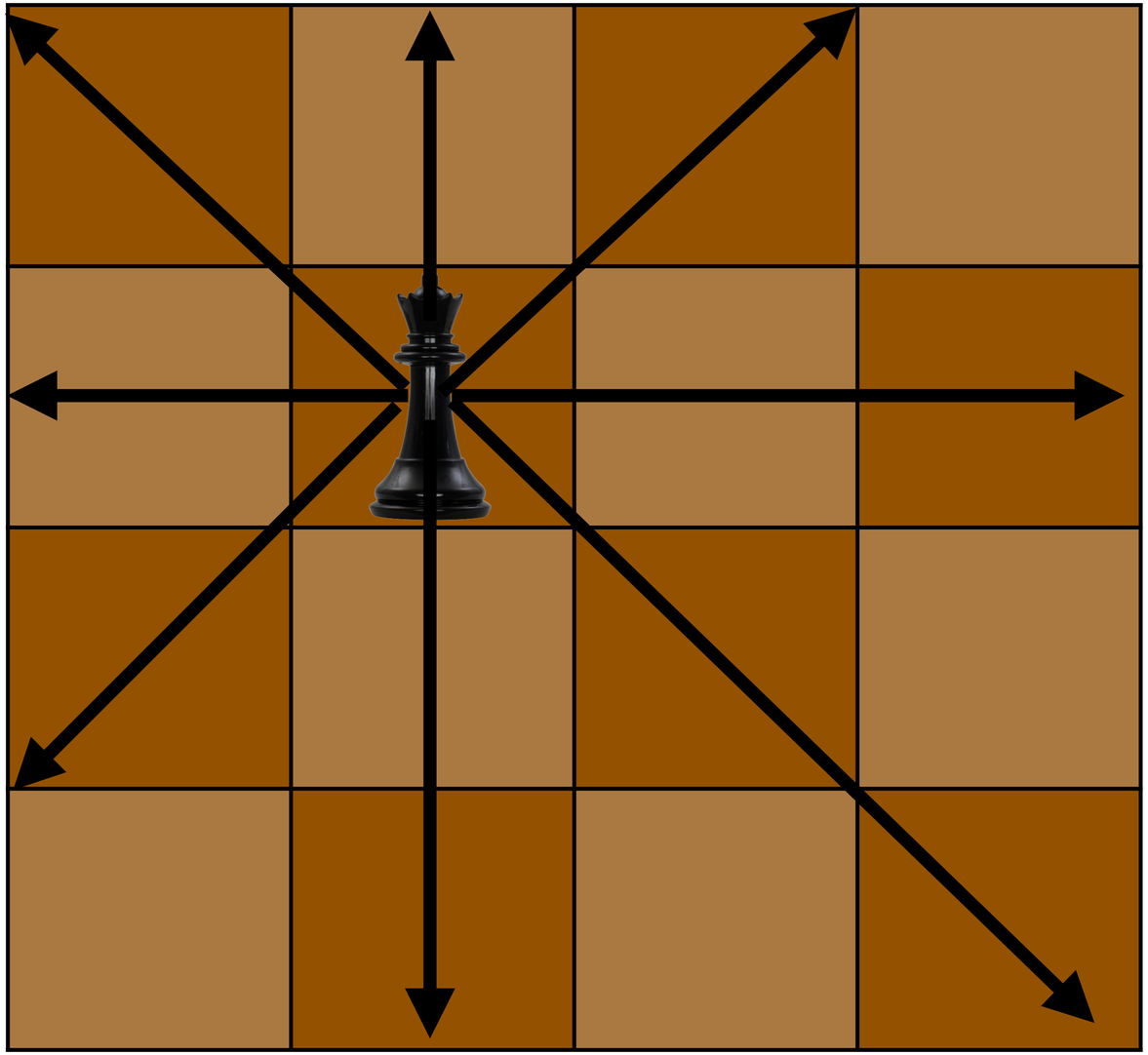
정육면체



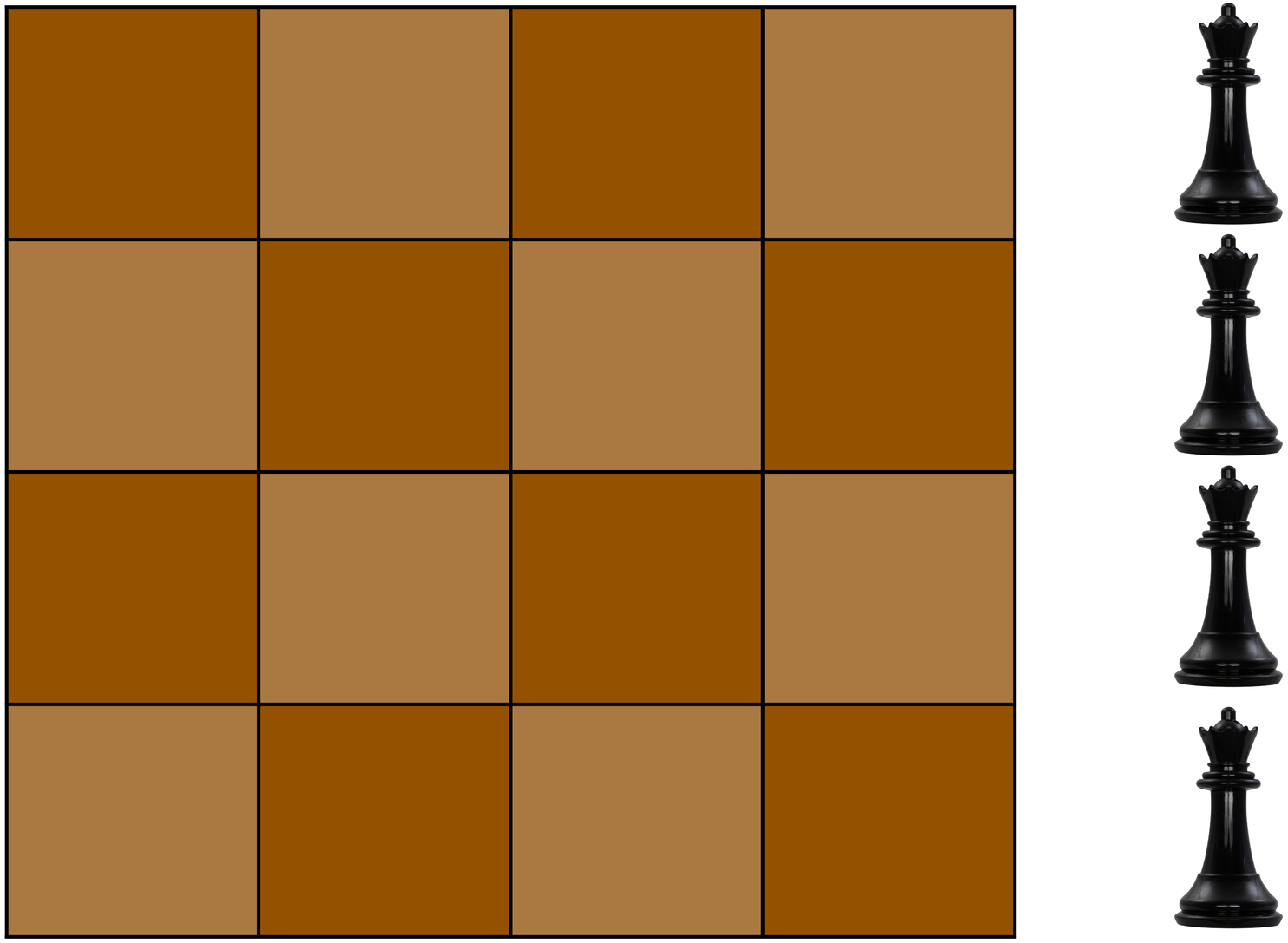
아주 상세한
set of instructions

Activity 3.





4 Queens

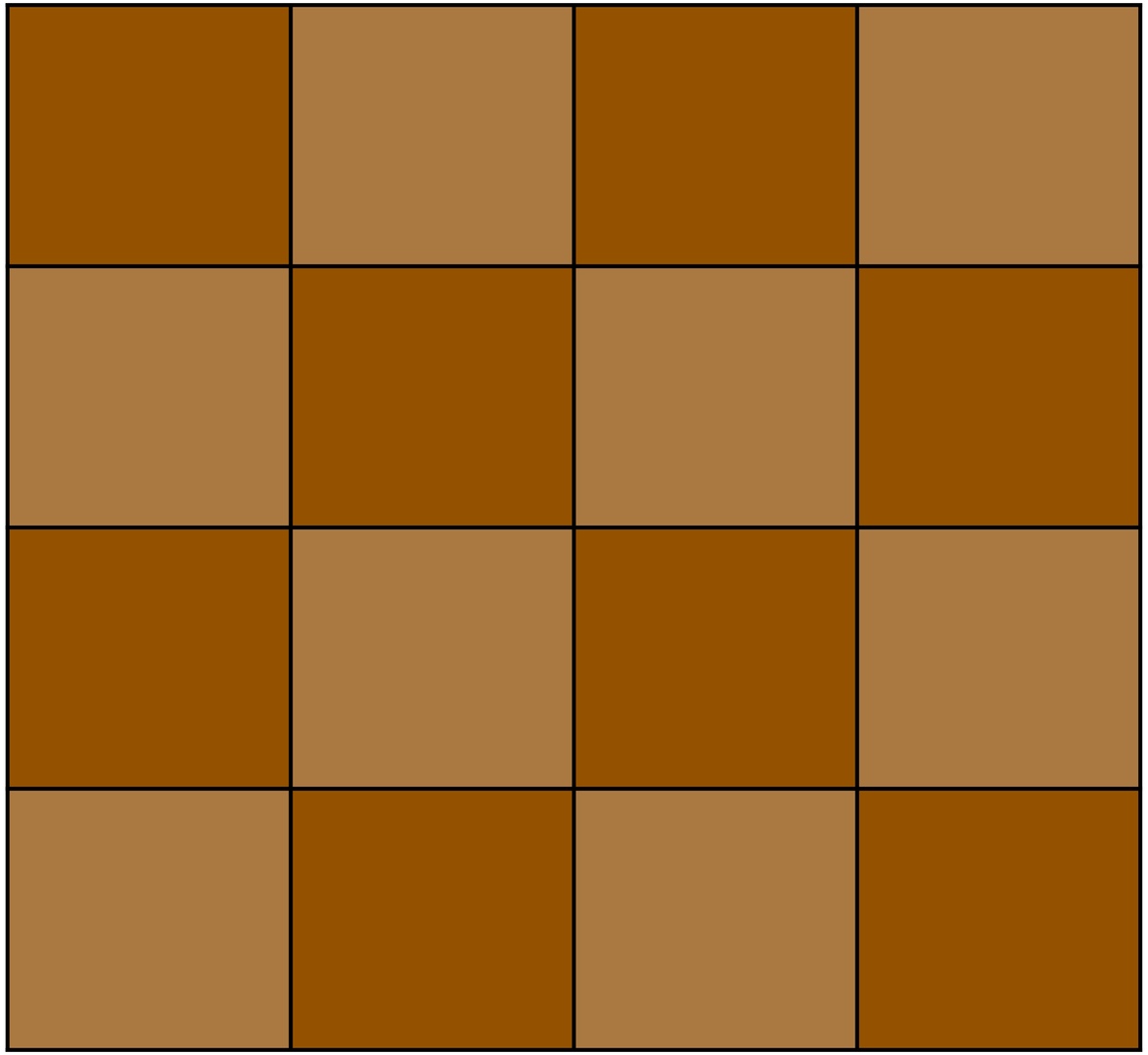


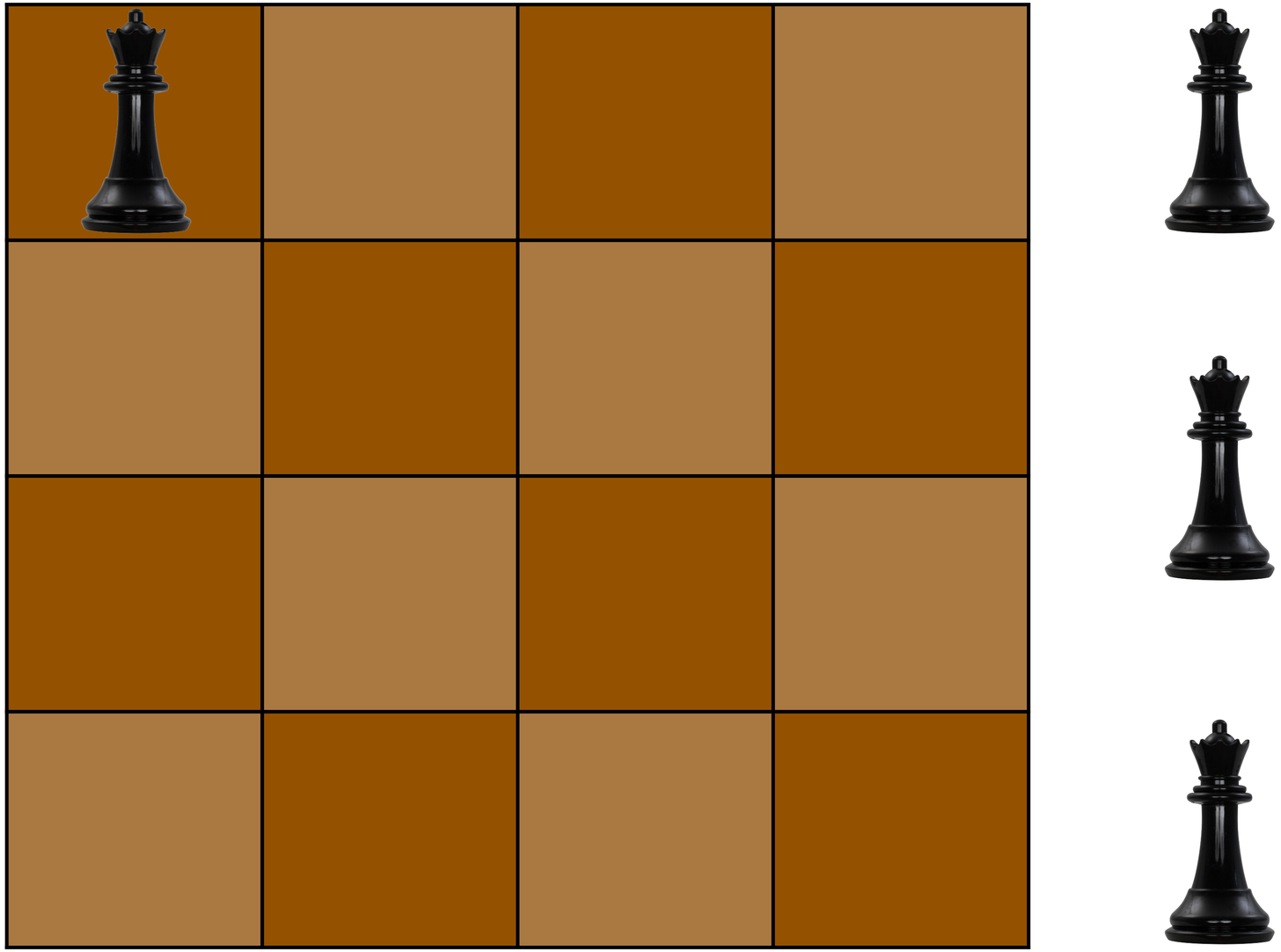
서로가 서로를
잡을 수 없게 만들기

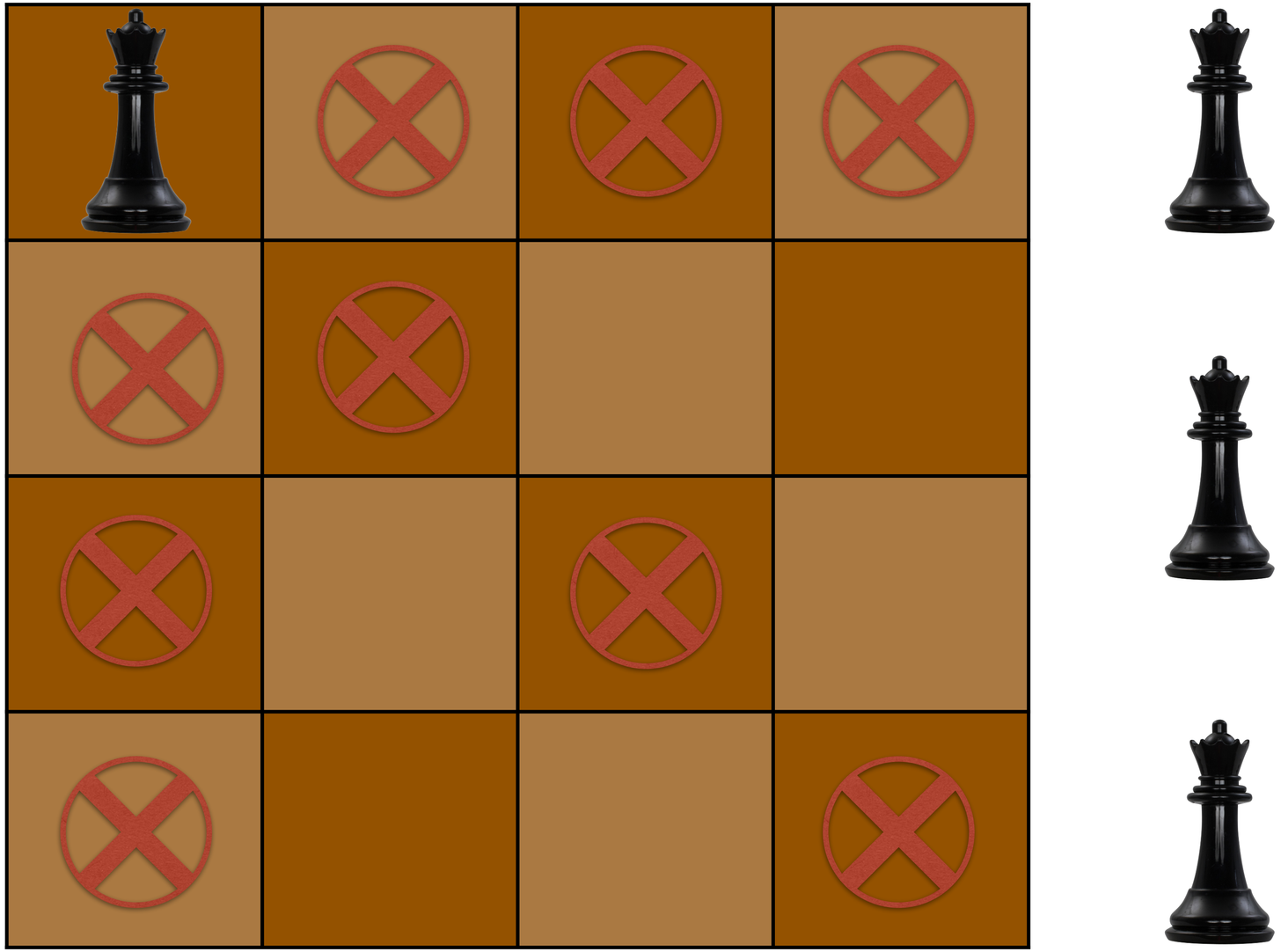
풀어봅시다 :)

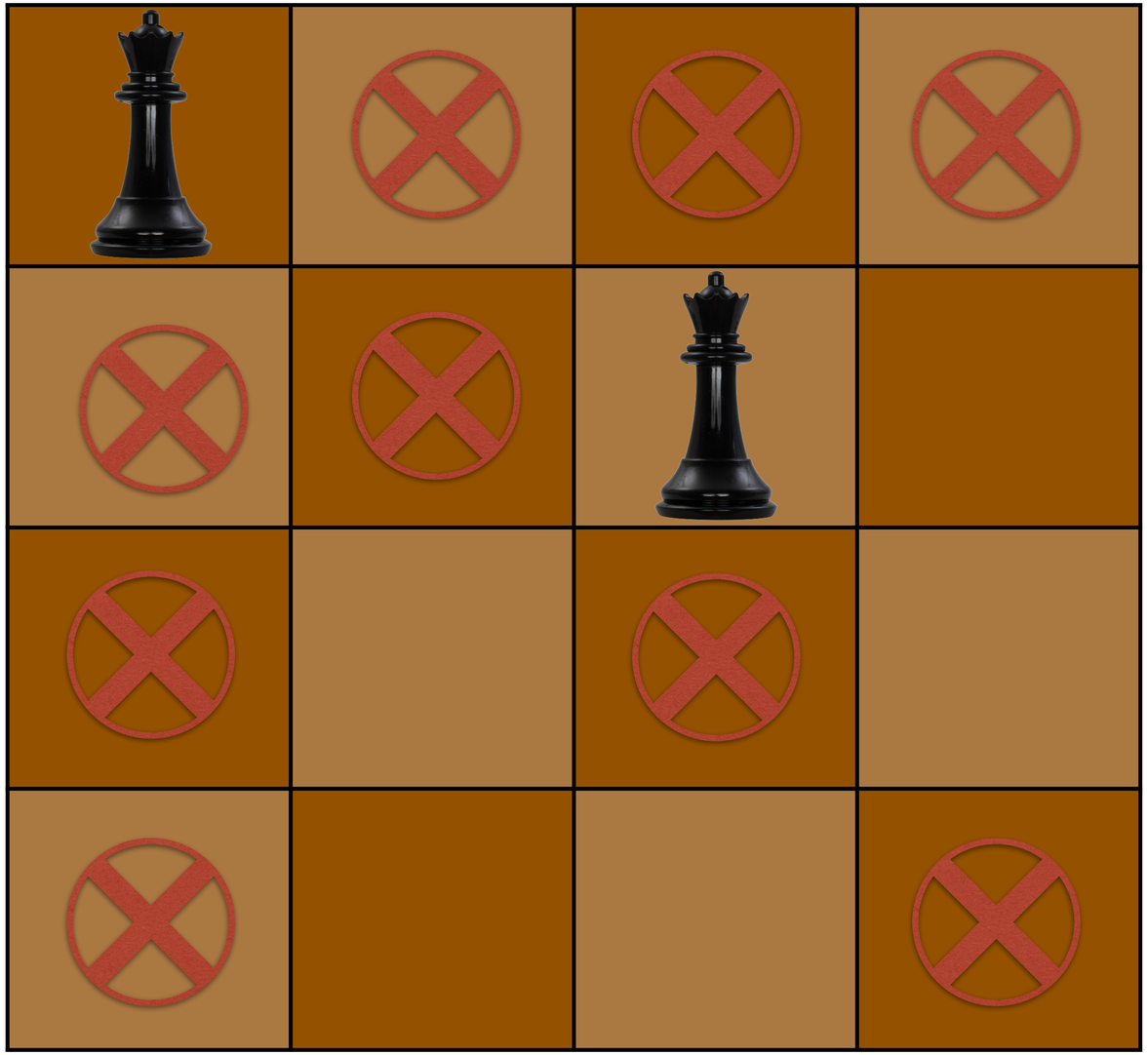
시간은 5분!

나와서 설명해주신 분께도 소소한 스타벅스☕

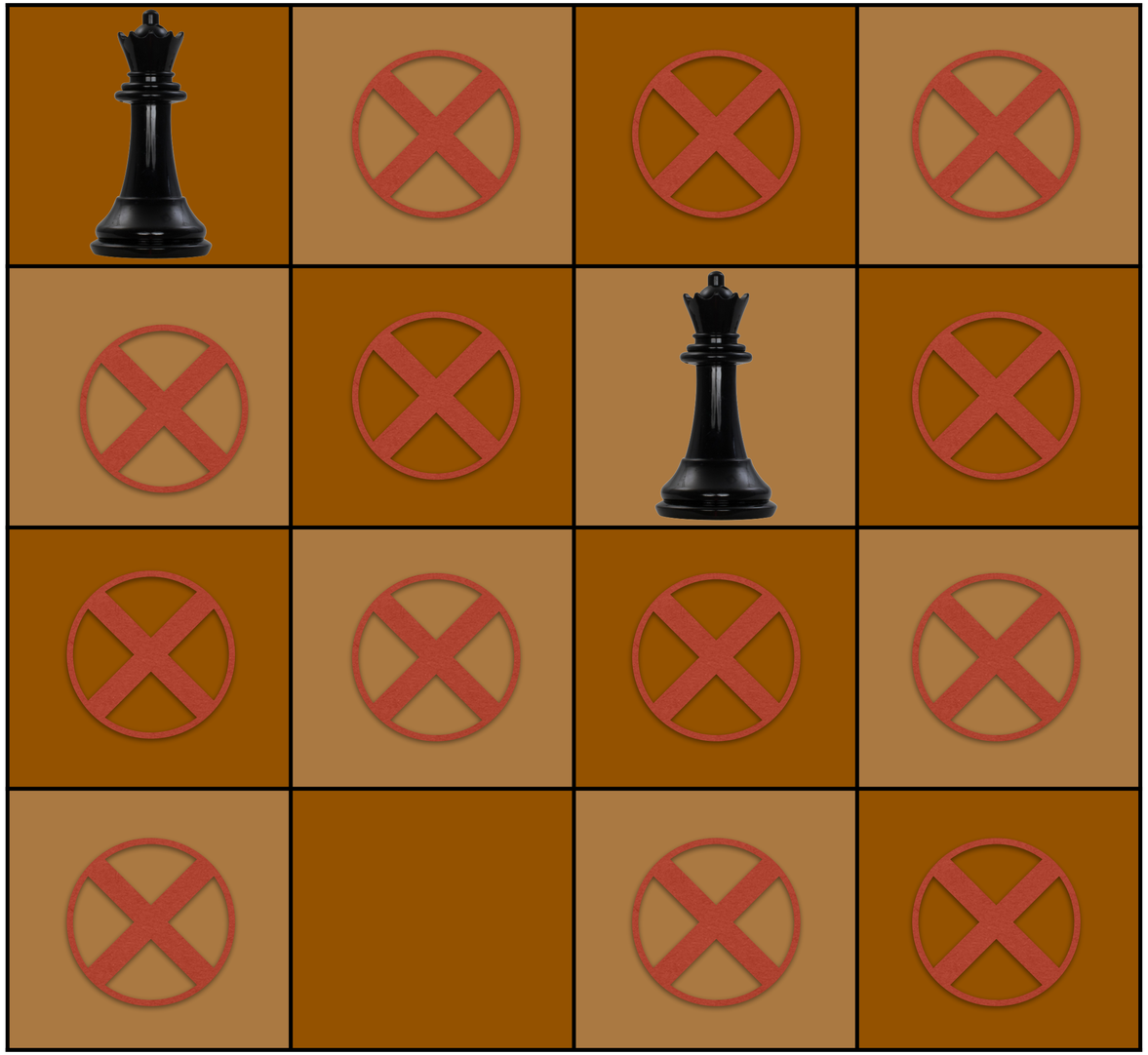






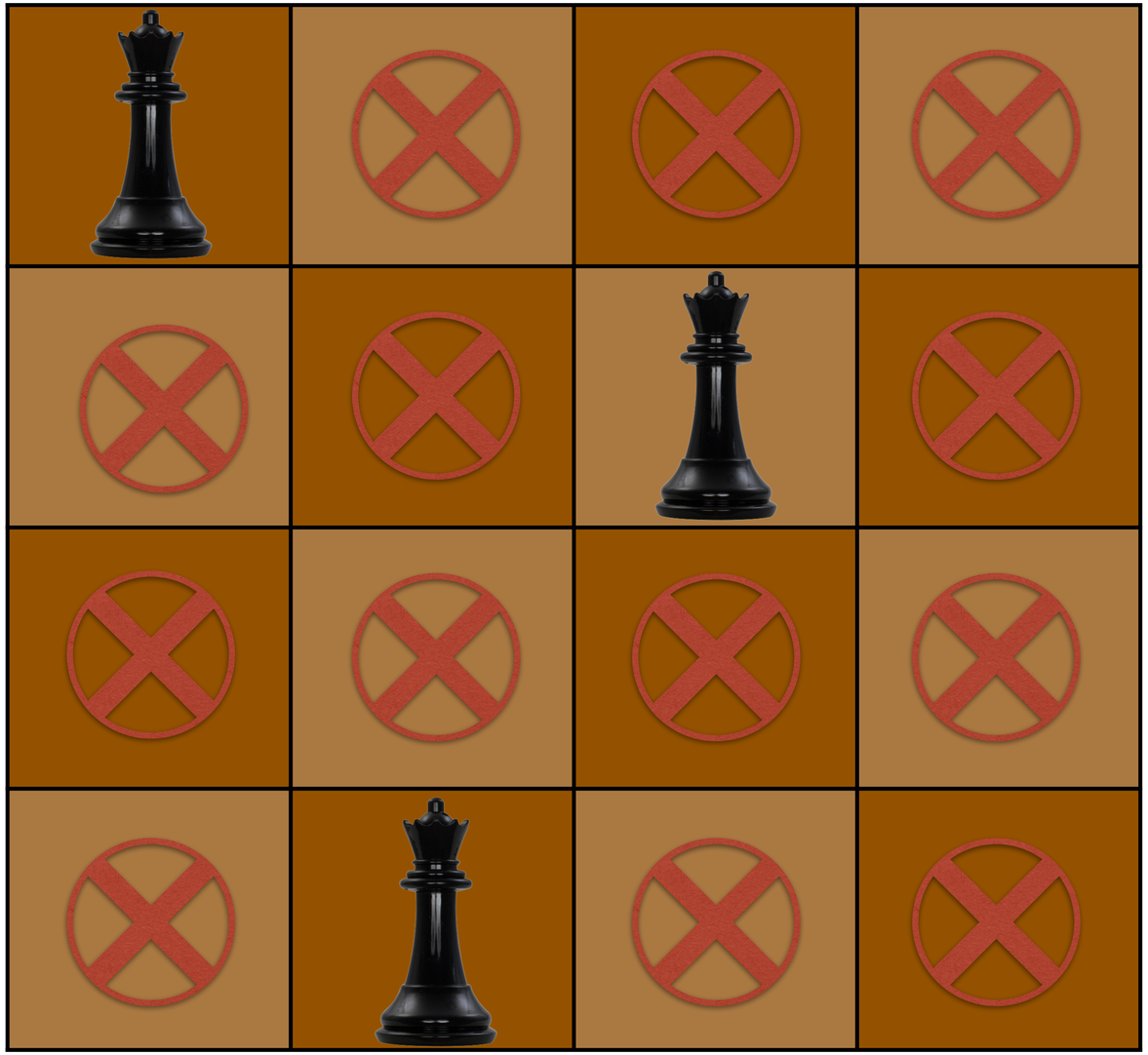


hphk.



Happy Hacking

hphk.



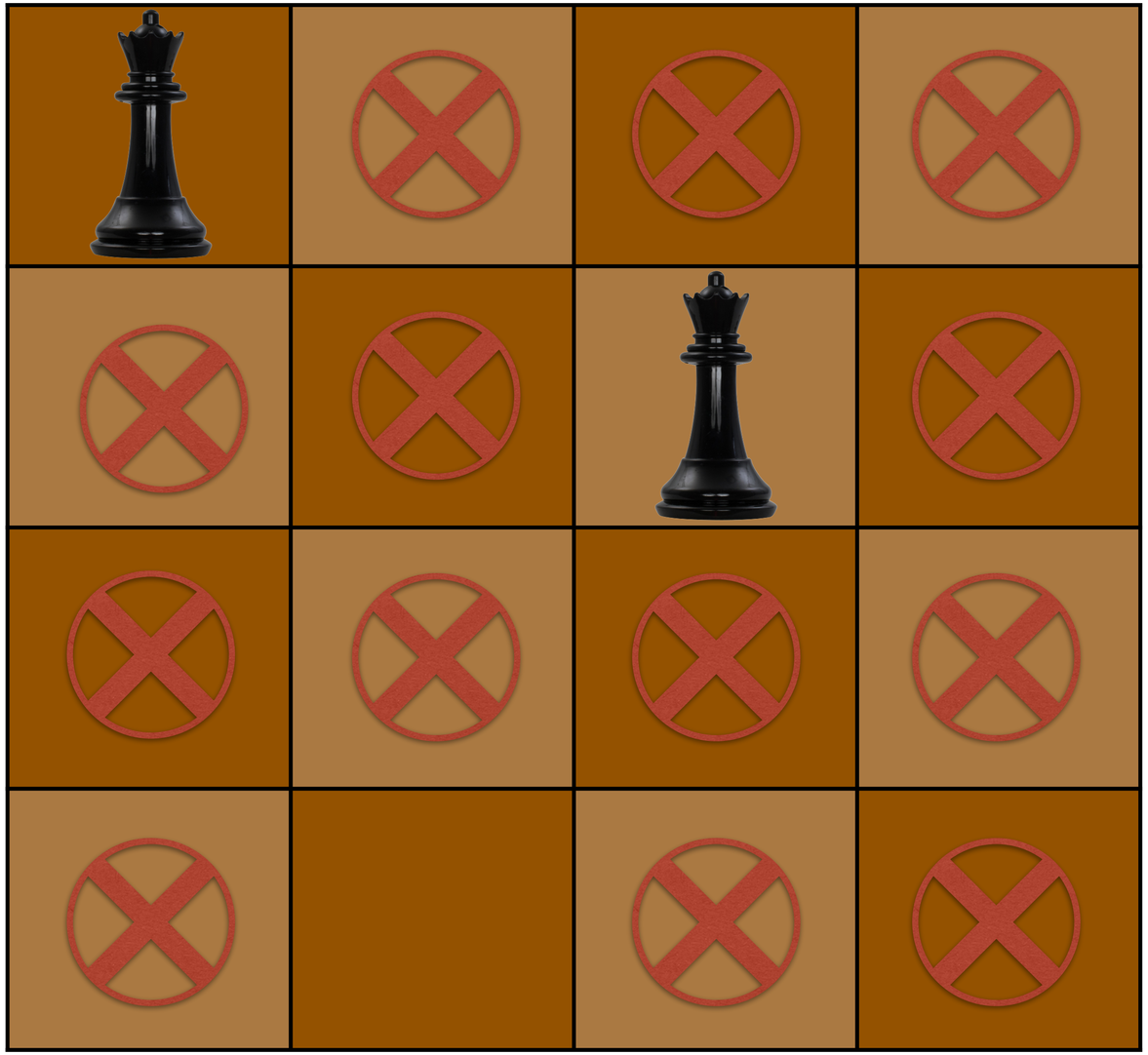
Happy Hacking

아 갈데가 없네

실패

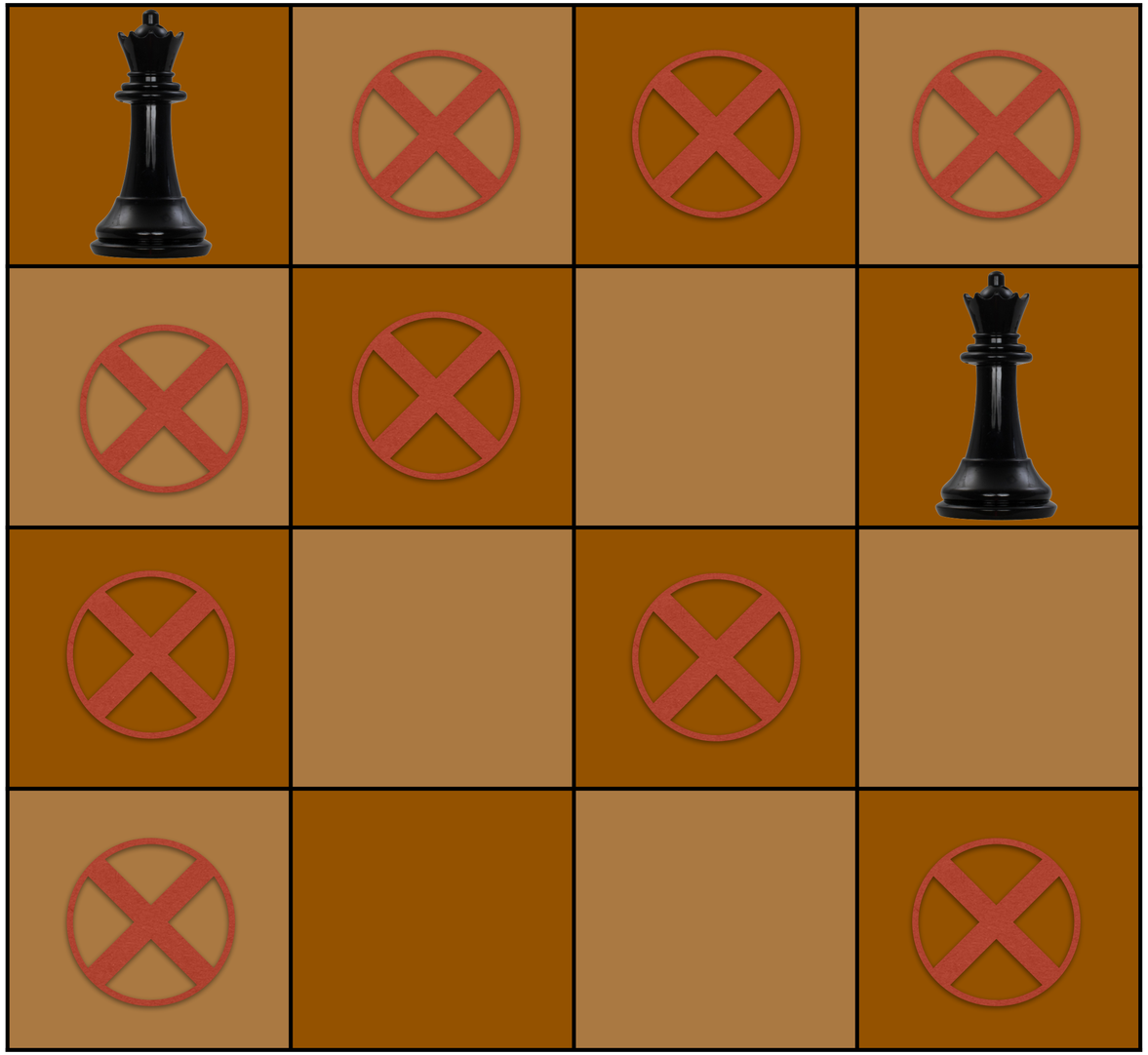
돌아가 보자

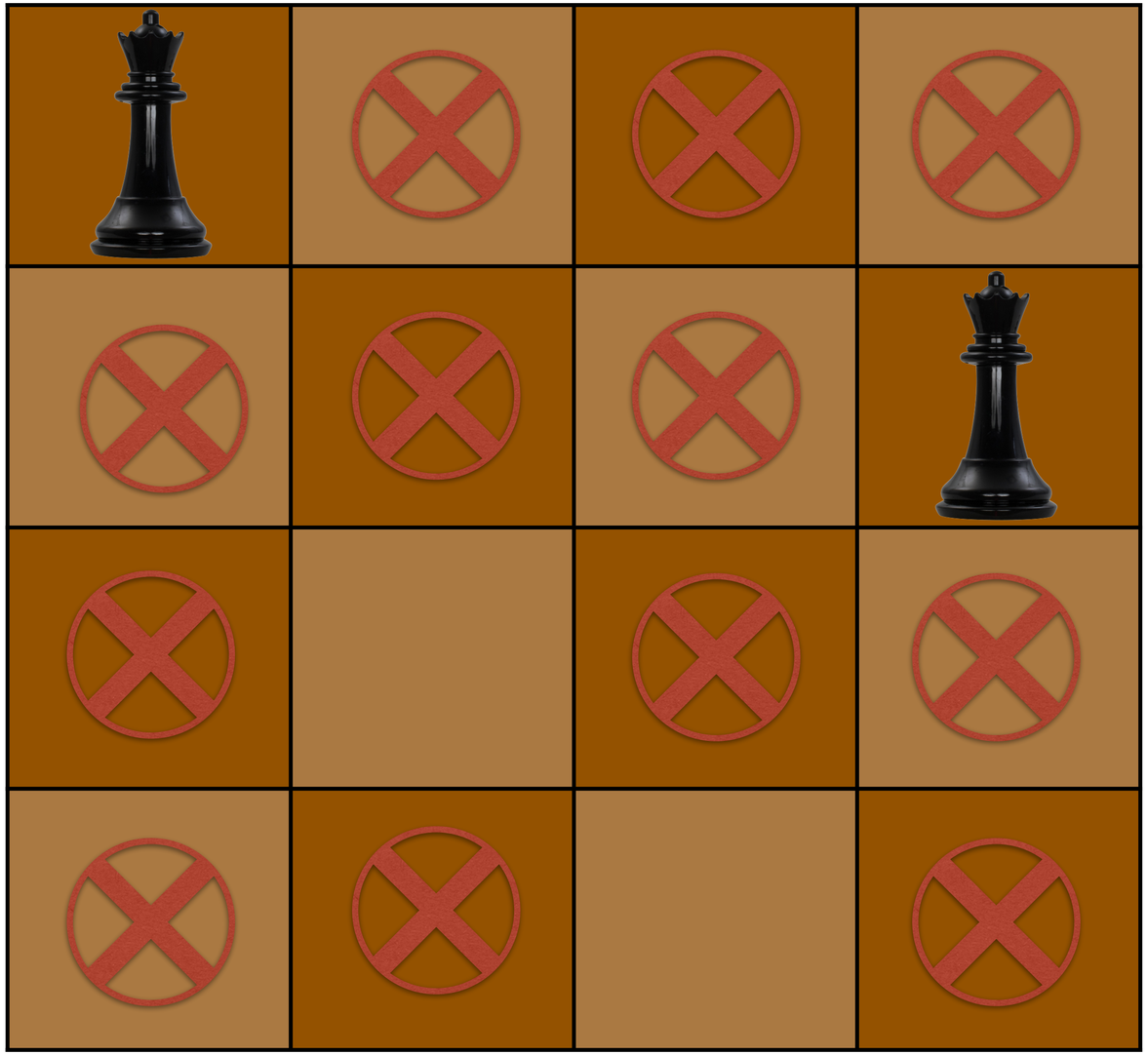
hphk.



Happy Hacking

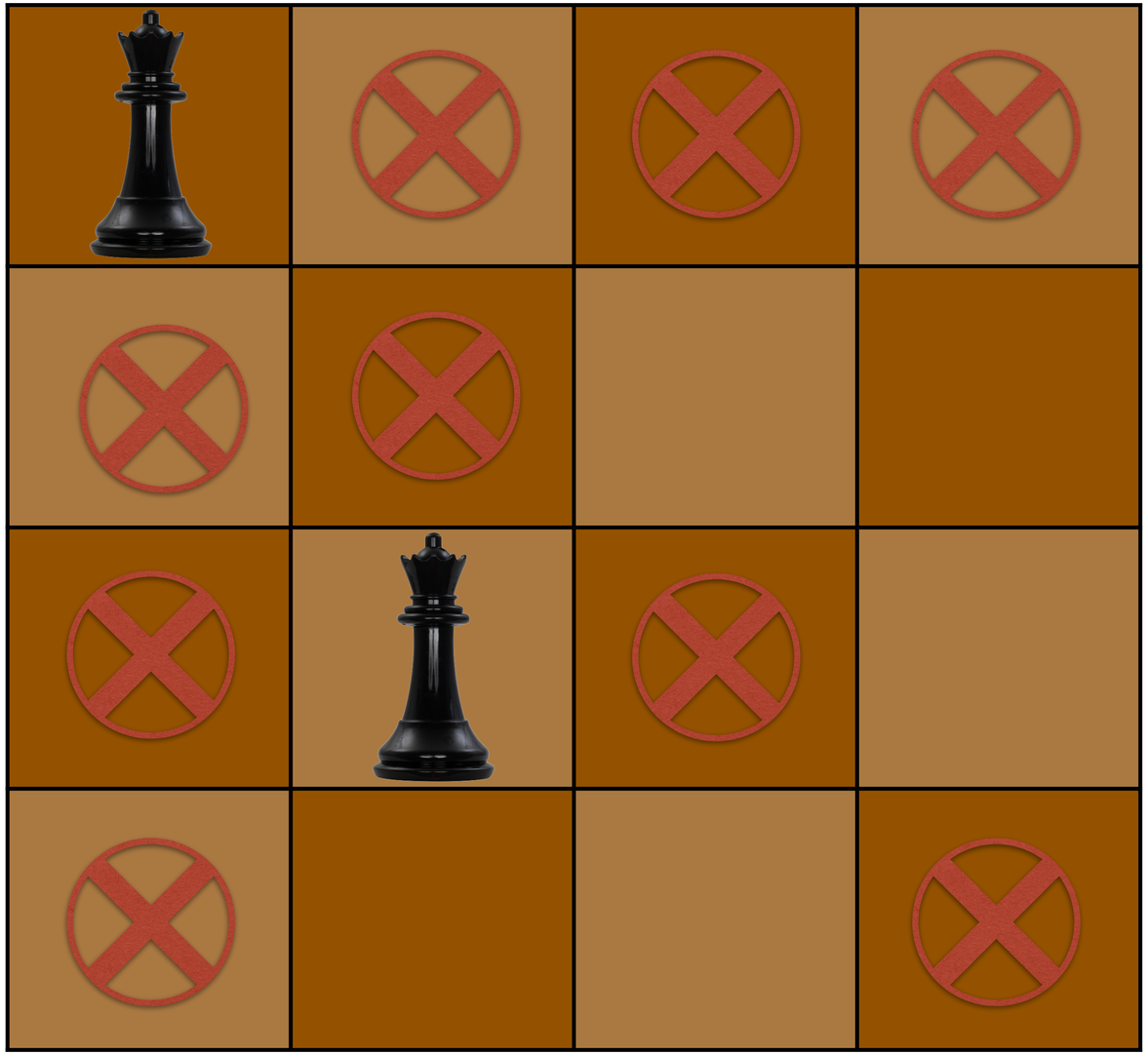
실패

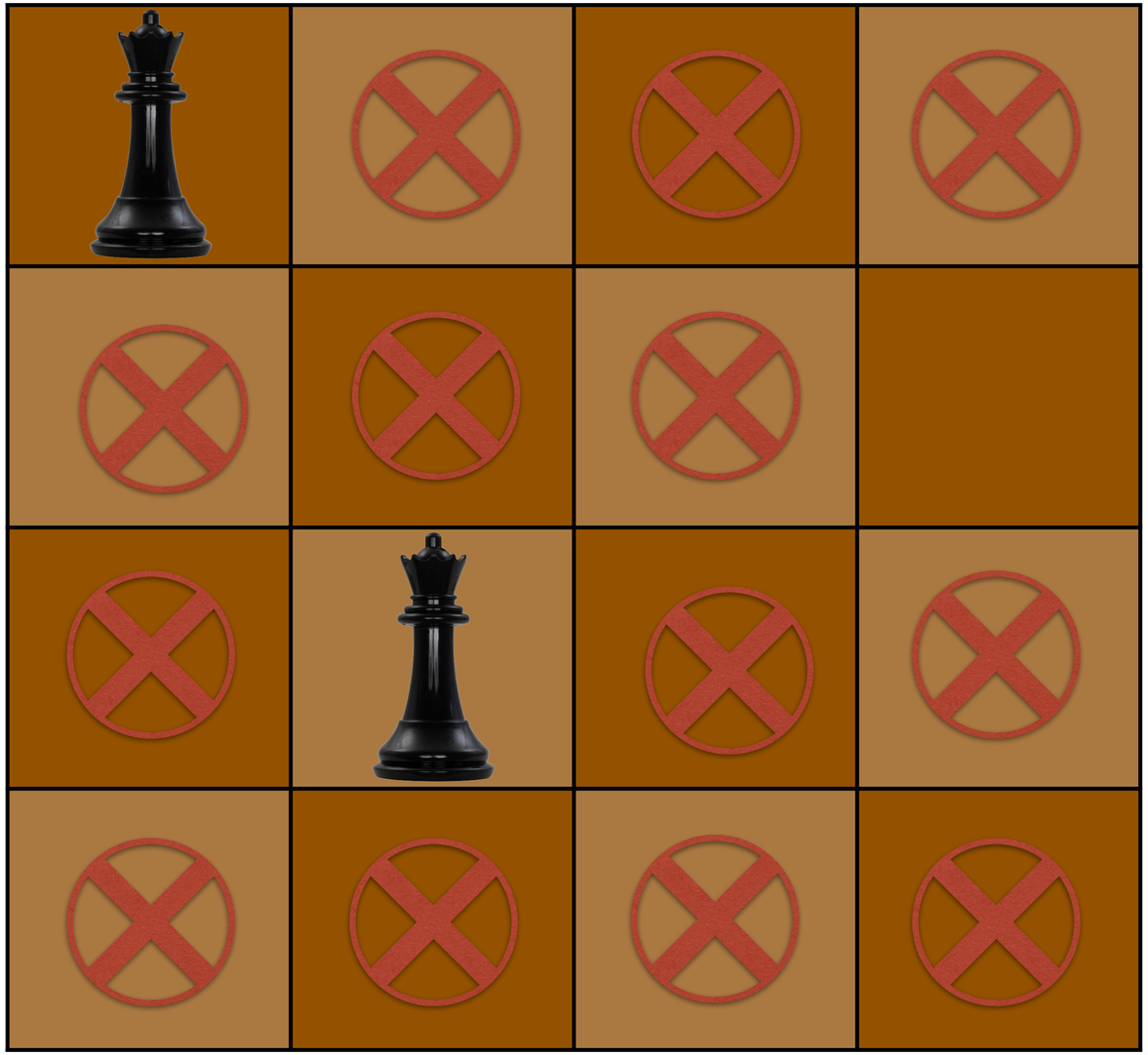




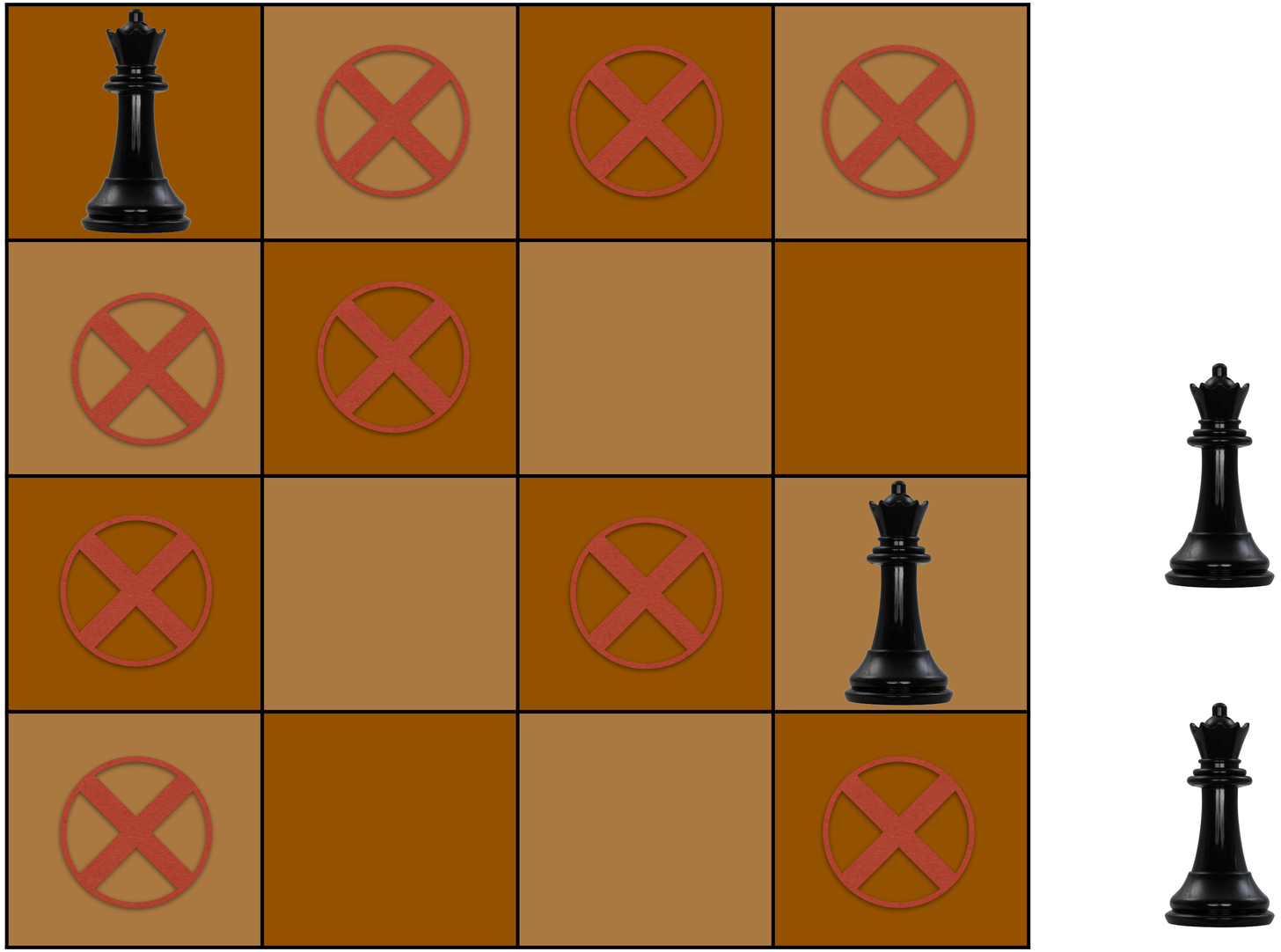
벌써 망삘

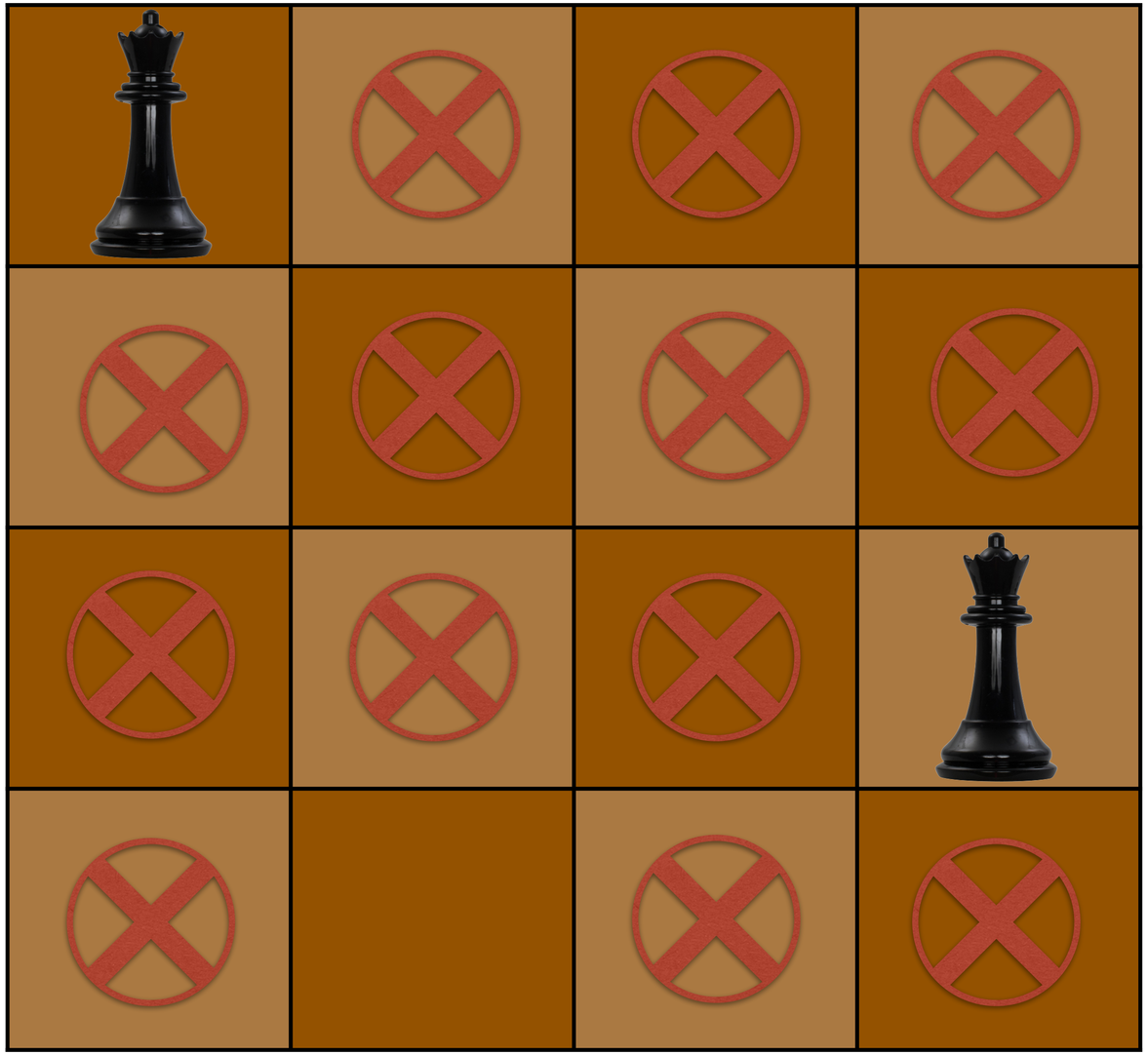
(다시 멀리) 돌아가 보자



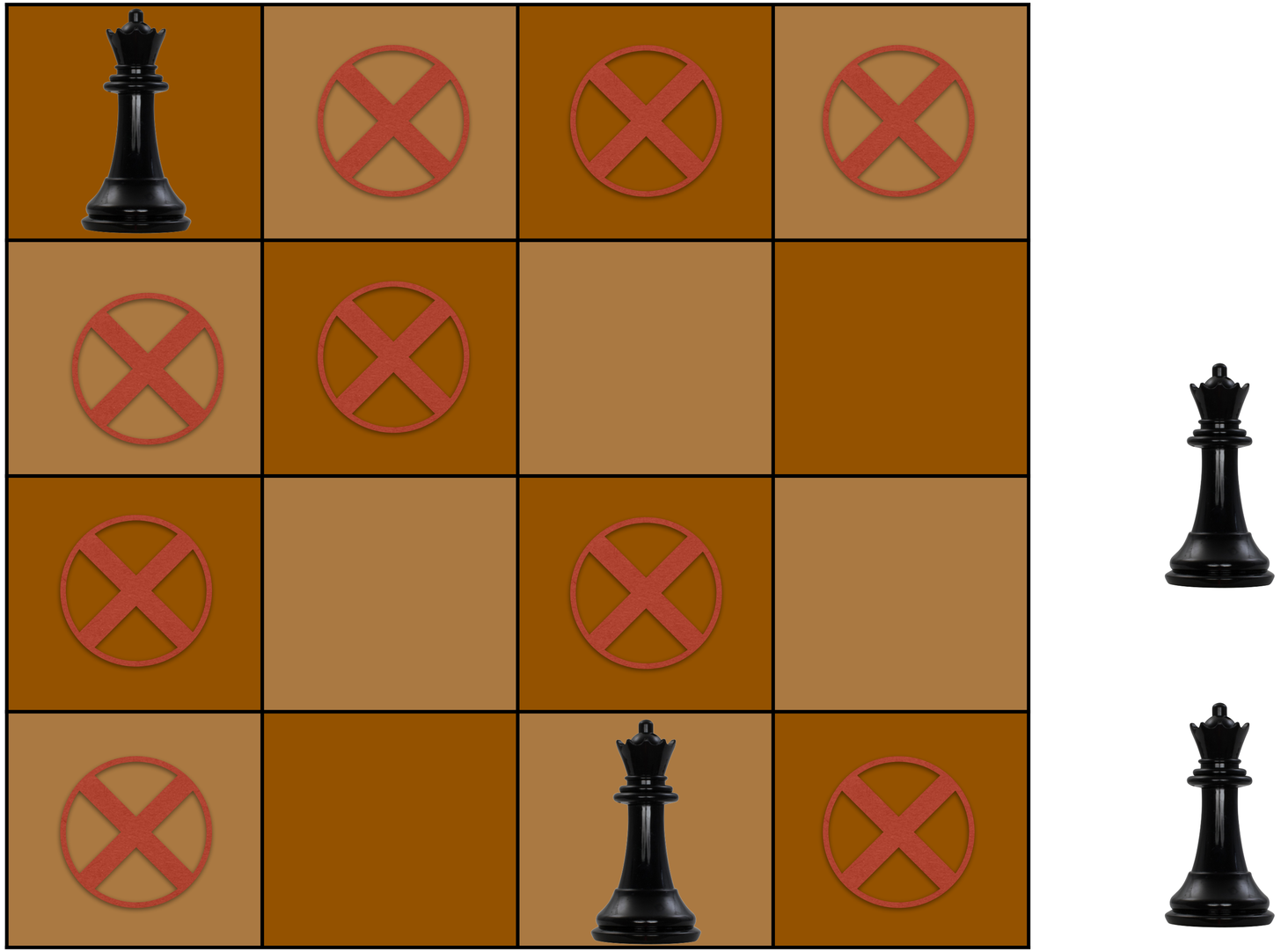


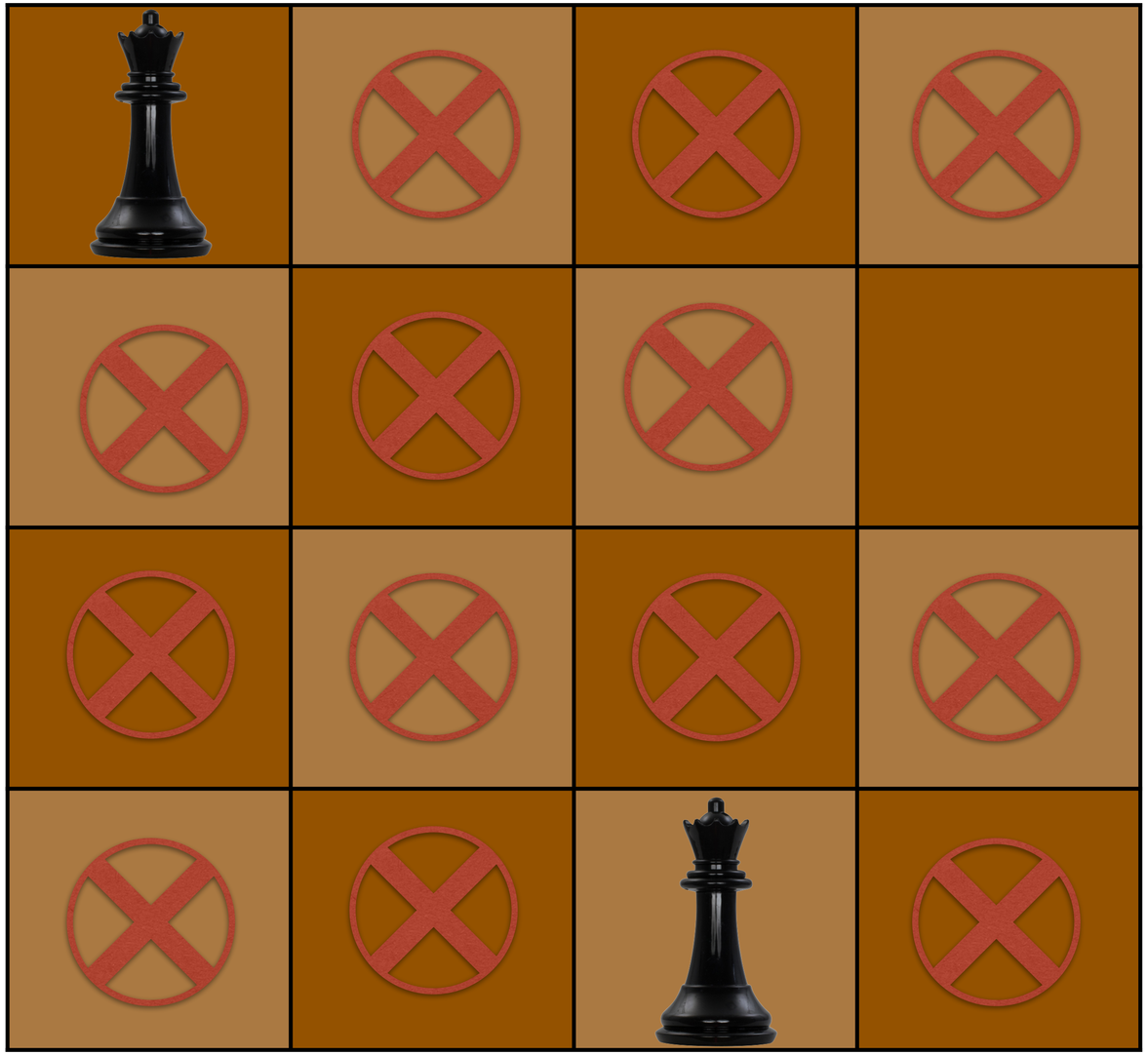
(다시 한 번) 돌아가 보자





이것도 아니고,



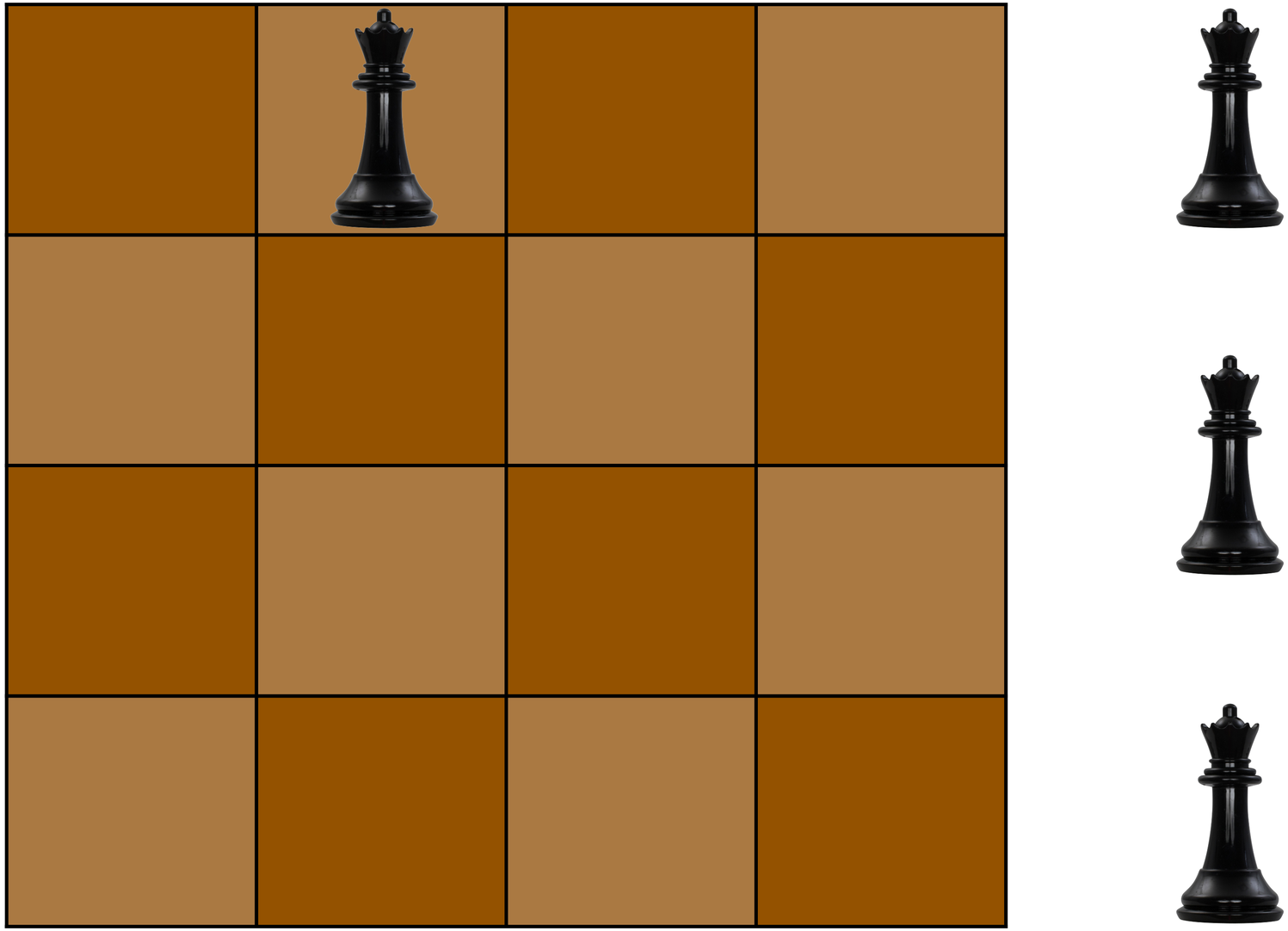


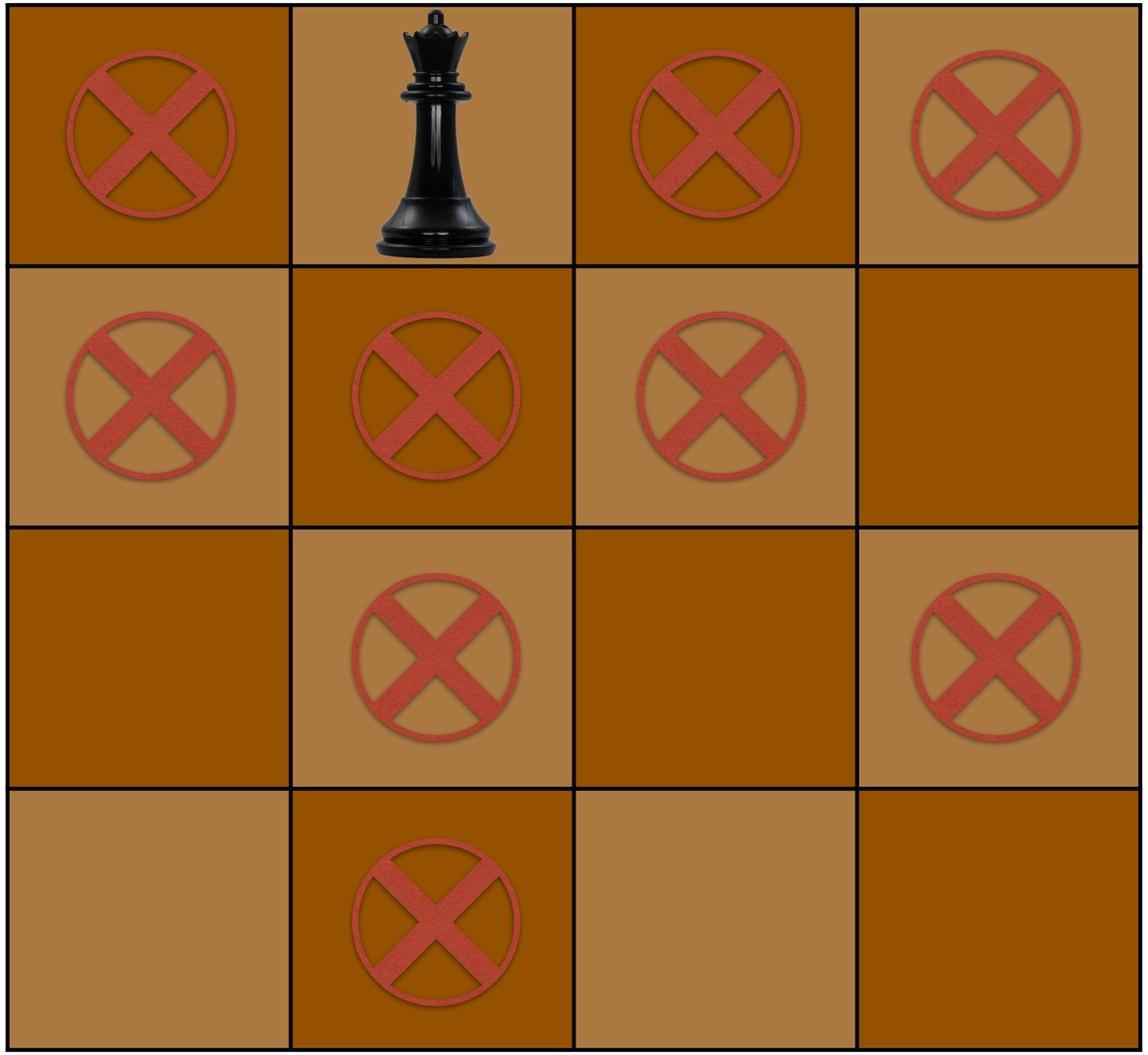
이것도 아닌데,

답은 없나?

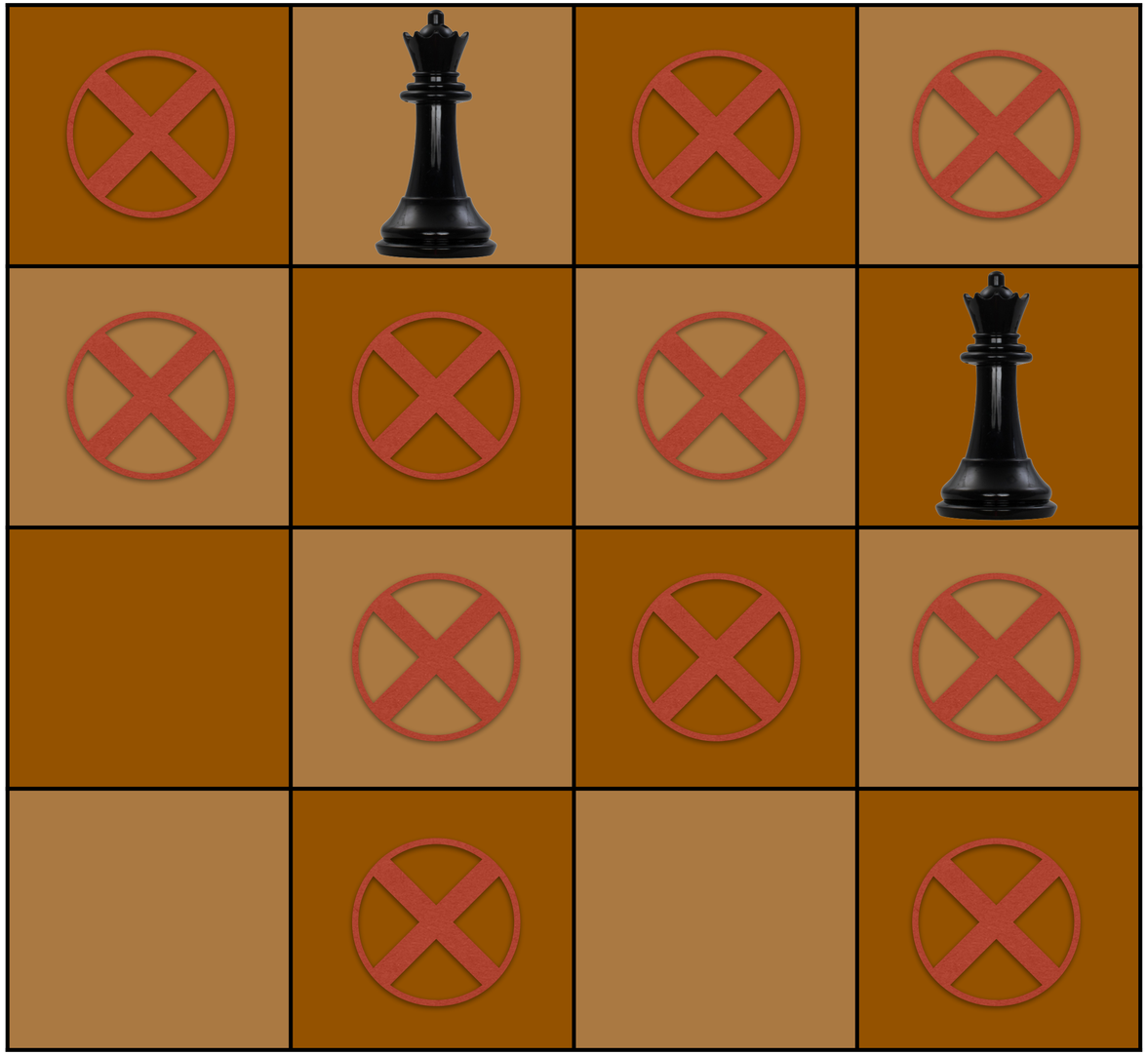
첫 수가 잘 못 되었나?

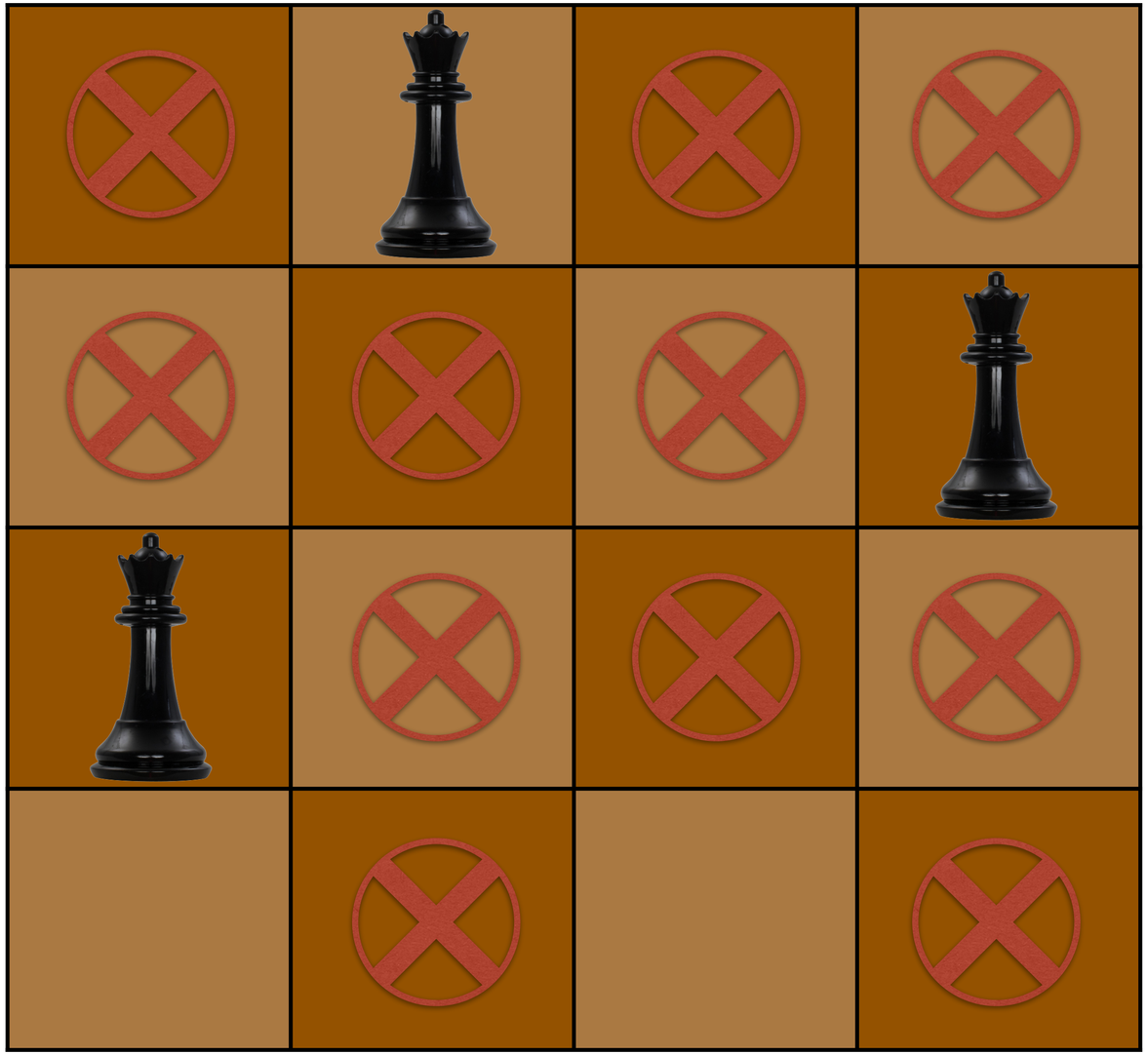
첫 수를 다시 두자!



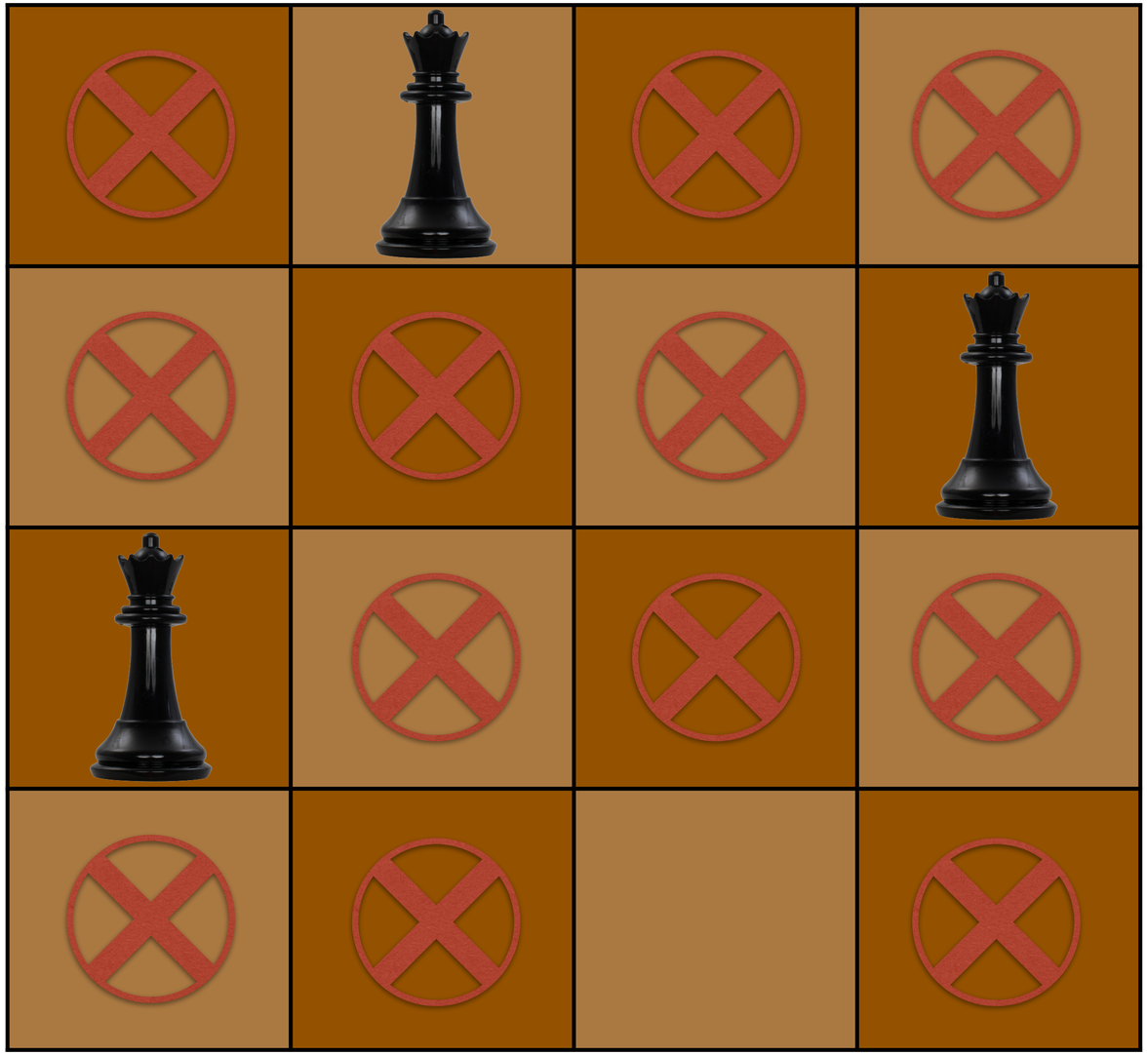


오 아까보다 훨씬 나은 듯

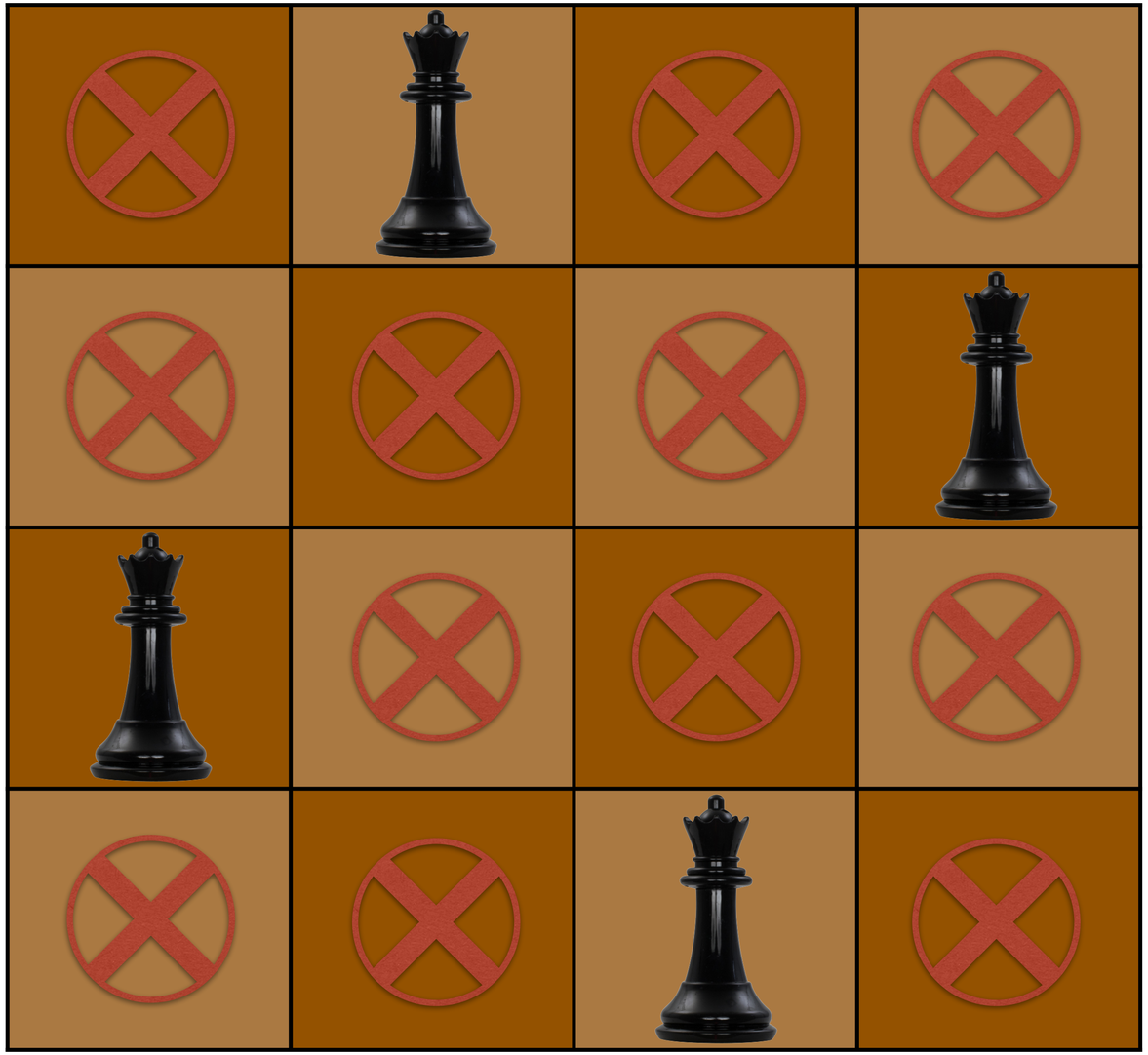




hphk.



Happy Hacking



성공

말로 설명하려면 어렵고,

이걸 공식화하려면
더 힘들다

1. 시도 되지 않은 수가 있는 한 계속해서,
2. 다음 (가능한) 수를 시도해라.
3. 만약에 퀸(queen)들이 서로 잡지 못하게 되면,
4. 답을 출력해라.

1. 왼쪽 가장 위의 칸부터 시작해라.
2. 만약 모든 퀸(queen)들이 서로 잡지 못하게 되면,
답을 출력해라.
3. a) 퀸들이 줄에 안전하게 놓일 수 있다면, 해당 칸을 선택하고, 여기에 여왕을 배치하면 해결책이되는지 반복적으로 확인해라.
b) 해당 칸에 퀸을 배치하면 해결책이 생겨 반환된다면,
답을 출력해라.
c) 퀸을 배치하는 것이 해결책으로 이어지지 않는다면, (a)로 이동하여 다른 칸을 시도해라.
3) 모든 행을 시도했지만 아무 것도 작동하지 않으면,
답이 없음을 출력해라.

1. 왼쪽 가장 위의 칸부터 시작해라.
2. 만약 모든 퀸(queen)들이 서로 잡지 못하게 되면,
답을 출력해라.
3. a) 퀸들이 줄에 안전하게 놓일 수 있다면, 해당 칸을 선택하고, 여기에 여왕을 배치하면 해결책이되는지 반복적으로 확인해라.
b) 해당 칸에 퀸을 배치하면 해결책이 생겨 반환된다면,
답을 출력해라.
c) 퀸을 배치하는 것이 해결책으로 이어지지 않는다면, (a)로 이동하여 다른 칸을 시도해라.
3) 모든 행을 시도했지만 아무 것도 작동하지 않으면,
답이 없음을 출력해라.

(다시) 돌아가 보자

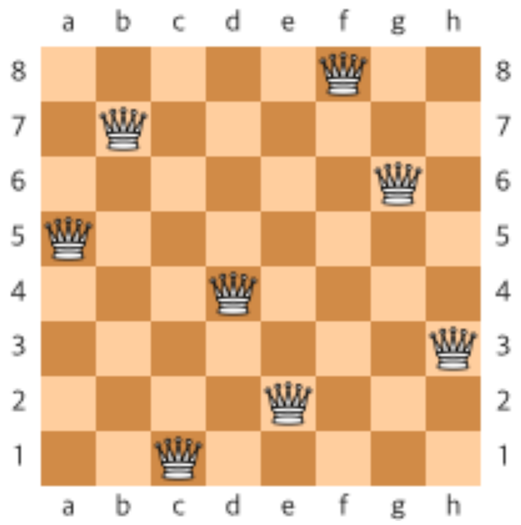
퇴각검색

Backtracking

```python
def solveNQUtil(board, col):
    if col >= N:
        return True
    for i in range(N):
        if isSafe(board, i, col):
            board[i][col] = 1
            if solveNQUtil(board, col+1) == True:
                return True
            board[i][col] = 0
    return False

def solveNQ():
    board = [ [0, 0, 0, 0],
              [0, 0, 0, 0],
              [0, 0, 0, 0],
              [0, 0, 0, 0]
             ]
    if solveNQUtil(board, 0) == False:
        print "정답이 존재하지 않습니다"
        return False
    printSolution(board)
    return True
```

N Queens Problem



n	Unique Solution	Total Solution
1	1	1
4	1	2
5	2	10
6	1	4
7	6	40
8	12	92
27	29,363,495,934,315,694	234,907,967,154,122,528

출처 : 위키피디아

알고리즘 :(

마지막으로,
알고리즘이 어려운
또 다른 이유

Logical Thinking

컴퓨터
== 논리기계

Logical?

Activity 4.

D

3

F

7

주장 : 만약 한쪽이 D이면 반대쪽은 3이다

주장이 사실임을 확인하기 위해 뒤집어야할 카드는?

D	3	F	7

주장 : 만약 한쪽이 D이면 반대쪽은 3이다

답은 D와 7

또 다른 예시 :)

16세

31세

콜라

맥주

주장 : 미성년자는 맥주를 마실 수 없음

답은 16세와 맥주

문제가 다른가?

아닌데...

왜 난이도가 다르게 느껴지지?

~~논리 구조 -> 문제 해결~~

경험(배경 지식) -> 문제 해결

야 16se! 너 뭐마시고 있어?(경험)

맥주 마시고 있는 너! 몇살이야?(경험)

Computational Thinking?

How to program

for problem solving

Session 1
SW 의 시대 속에서

Session 1.
SW 의 시대 속에서

#4차산업혁명 #소프트웨어 #오픈소스 #API

4차 산업혁명????

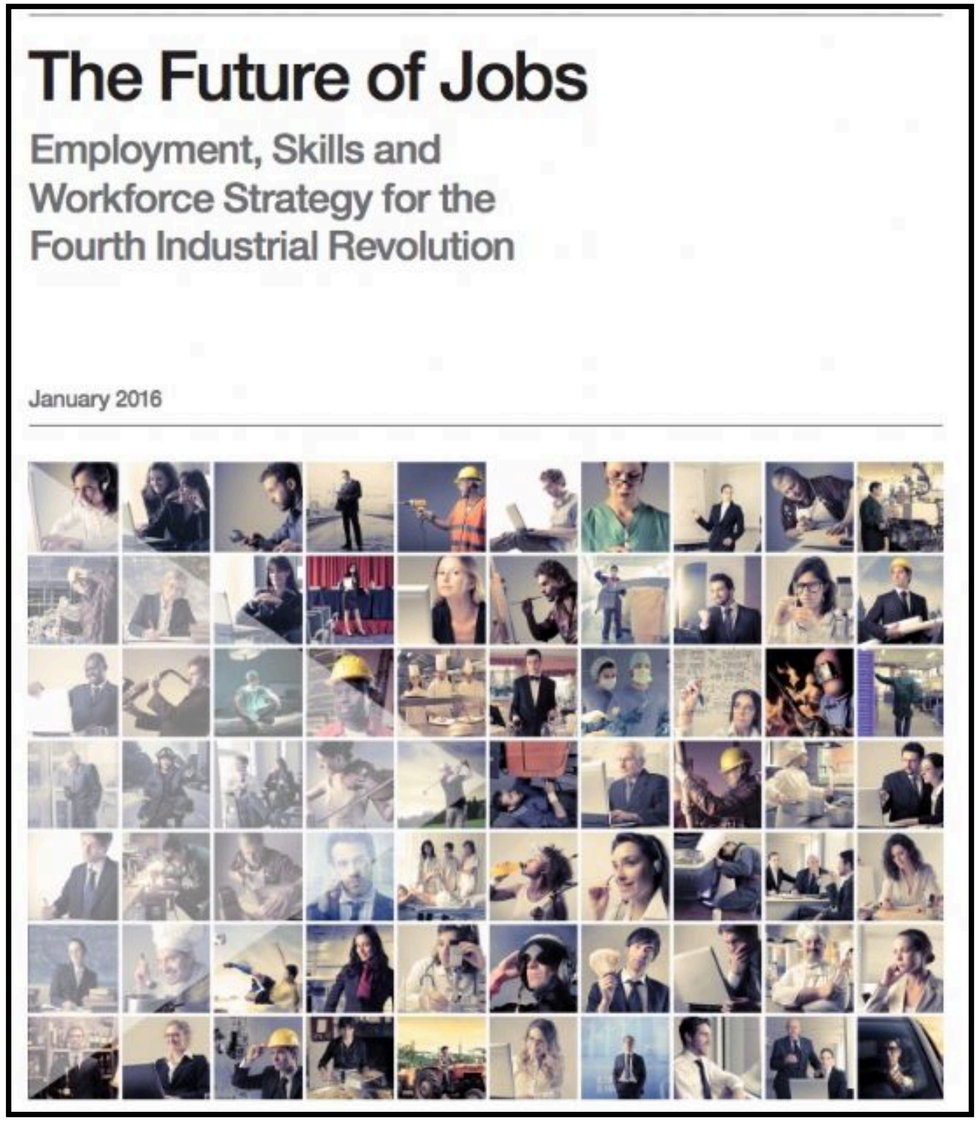
#4차산업 #혁명 #이게 #뭡니까 대체

제1차 산업혁명
18세기
증기기관 기반의
기계화 혁명
제2차 산업혁명
19~20세기 초
전기 에너지 기반의
대량생산 혁명
제3차 산업혁명
20세기 후반
컴퓨터와
인터넷 기반의
지식정보 혁명
제4차 산업혁명
(제2차 정보혁명)
21세기 초반~
[지능정보기술]
지능
AI SW
+
정보
빅데이터
IoT
클라우드

[미디어 세상]'4차 산업혁명' 구호는 버려야

이준웅 | 서울대 언론정보학과 교수

이제 제발 그만



The Future of Jobs

Employment, Skills and Workforce Strategy for the Fourth Industrial Revolution

January 2016

사회

일자리 위협하는 4차 산업혁명, 내 직업은 괜찮을까

조선닷컴 인포그래픽스팀

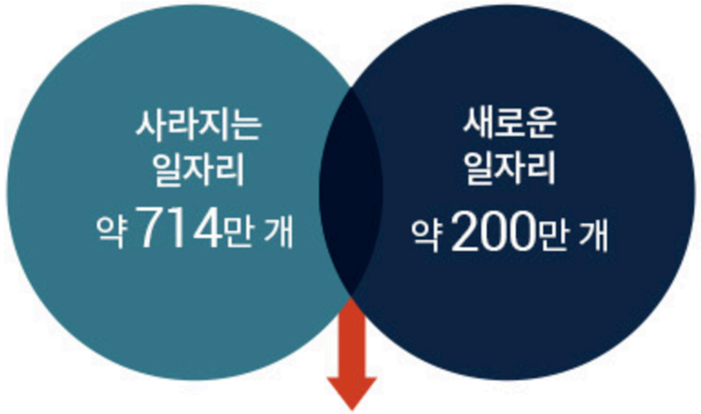
[단독] 국내 1800만명 일자리, 10년내 인공지능·로봇에 위협

4차산업혁명…일자리 500만개 사라진다

WEF 보고서 "사무-관리직 특히 큰 타격"

4차 산업혁명 시대의 일자리 변화

(자료:세계경제포럼 2015~2020년도 고용 전망)



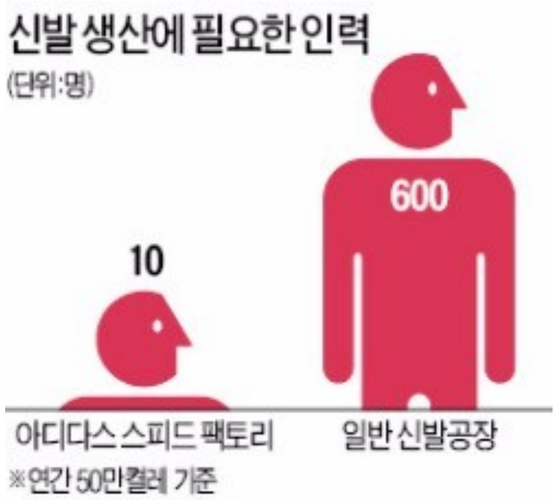
전체적으로 5년간 514만 여개의 일자리가 감소될 전망이며, **해마다 103만 개 일자리가 줄어드는 셈이다.**

【2015-2020년 직군별 고용 증감규모 추정(단위:천명)】

순고용 감소	사무·행정	제조·생산	건설·채굴	디자인·스포츠·미디어	법률	시설·정비
	-4,759	-1,609	-497	-151	-109	-40
순고용 증가	비즈니스·금융	경영	컴퓨터·수학	건축·엔지니어링	영업·관리직	교육·훈련
	492	416	405	339	303	66

adidas speed factory

23년 만에 독일로 돌아온 아디다스.
100% 로봇 자동화를 통해
10명의 인력으로 연간 50만 켤레 생산.



Amazon
물류 관리

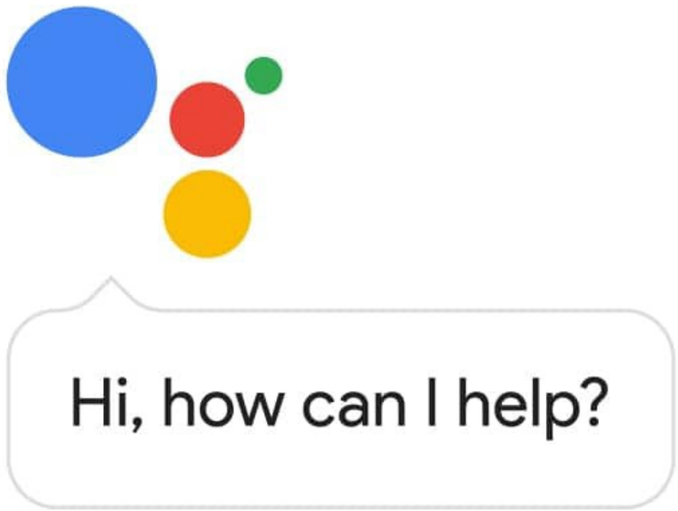
KIVA.

20만개 물류 처리.

25대의 KIVA.(150명 인력분)

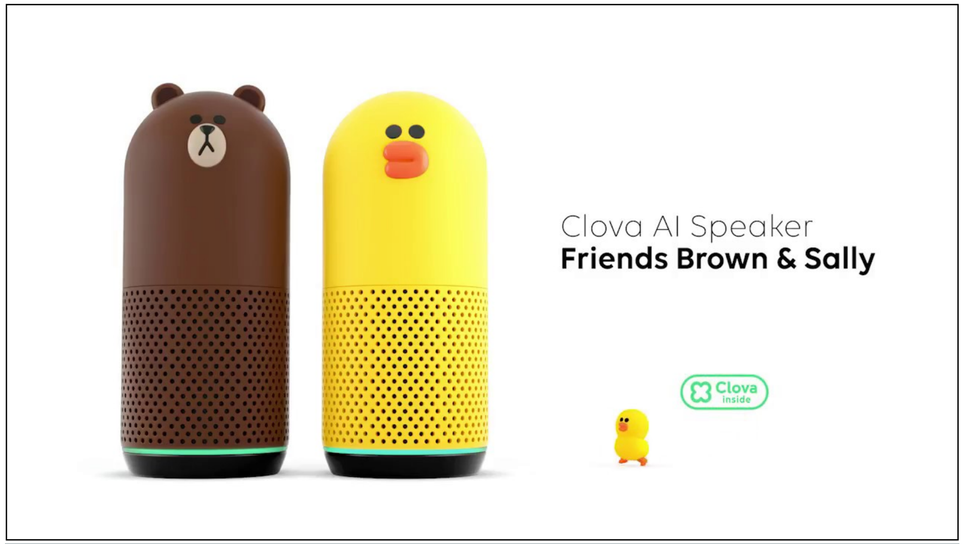
아마존 드론 배송

드론 물류 배송.
케임브리지 근처에서 드론 배송 허가.
2016년 12월부터 실제 배송에 활용되는 중.


Hi, how can I help?

Google Lens

급성장하는 AI 스피커 시장, 국내외 주요 제품은
아마존
제품명 • 에코
출시 시점 • 2014년 11월
특징 • 음성비서 알렉사 탑재
구글
• 구글홈
• 2016년 10월
• 음성비서 구글 어시스턴트 탑재
애플
• 홈팟
• 2017년 12월
• 시리 탑재, 스피커 기능 강조
마이크로소프트
• 인보크
• 올 하반기
• 음성비서 코타나 탑재
네이버
• 웨이브
• 올 하반기
• AI 플랫폼 클로바 탑재
SK텔레콤
• 누구
• 2016년 9월
• 음악 및 쇼핑 서비스 연계
자료:업계 종합


Clova AI Speaker
Friends Brown & Sally
Clova inside

kakaomini

	Echo Dot	Echo	Echo Plus	Echo Spot	Echo Show
Price	$49.99	From: $99.99	$149.99	$129.99	$229.99
Overview	Add Alexa to any room	Room filling sound with six fabrics or finishes	Includes a built-in smart home hub	Stylish and compact Echo with a screen	Optimized for visuals and room filling sound
Screen size				2.5" Screen	7" Screen
Speaker size	0.6" speaker	2.5" woofer and 0.6" tweeter	2.5" woofer and 0.8" tweeter	1.4" speaker	Dual 2.0" stereo speakers
Dual-speakers with room-filling sound, powered by Dolby		✓	✓		✓
Play video from Amazon Video and more				✓	✓
Built-in hub for simple setup of compatible connected smart home devices			✓		
Streaming Wi-fi music (including Amazon Music, Spotify, Pandora, and more)	✓	✓	✓	✓	✓
Line-out with 3.5 mm cable or Bluetooth	✓	✓	✓	✓	Bluetooth only
Free audio calls to US, Mexico, and Canada	✓	✓	✓	✓	✓
Control smart home devices	✓	✓	✓	✓	✓
Device size (actual size and weight may vary)	1.3" x 3.3" x 3.3", 5.7oz (32 mm x 84 mm x 84 mm, 163 grams)	5.9" x 3.5" x 3.5", 29.0 oz. (148 mm x 88 mm x 88 mm, 821 grams)	9.3" x 3.3" x 3.3", 33.6 oz. (235 mm x 84 mm x 84 mm, 954 grams)	4.1" x 3.8" x 3.6", 14.8 oz. (104 mm x 97 mm x 91 mm, 419 grams)	7.4" x 7.4" x 3.5", 41.0 oz. (187 mm x 187 mm x 90 mm, 1170 grams)

YOLO?!

TESLA

기술의 변화 속도가
과거에 비해 빨라졌다.

#어떻게 #살아남을 것인가.

4차 산업혁명 참여 기업별 주요 전략

기업명	전략	응용 분야
폭스콘	로봇 생산 체제 도입	스마트폰
닛산	무인자동차 개발	전지, 자율주행 차
아마존	신개념 쇼핑 체계(아마존고) 구축, 인공지능 비서(에코) 프로그램 개발	쇼핑, 스마트홈
GE	디지털 트윈 활용 '비포서비스' 구현	발전기, 엔진
스타벅스	O2O 기술(사이렌오더), 고객 맞춤형 서비스 구축	비콘(Beacon)[4], 커피
아디다스	로봇 생산 체제 도입, 공장 본국(독일) 소환	신발, 웨어러블 기기
IBM	인공지능 컴퓨터 시스템(왓슨·Watson) 개발	센서
우버	운송·물류 효율화, 스마트 시티 구축	자동차(자율주행)
츠타야	동네 친화적 놀이터, 마을 사랑방 구축	도서, 마일리지 체계
테슬라	무인 전기자동차 개발, 재생에너지 생산	전기차, 솔라시티

Forbes 2019 Technology Trend

1. 5G
2. Chatbots
3. Connected Clouds
4. Blockchain
5. Data Analytic(Machine Learning)
6. GDPR
7. AR (feat.VR)
8. Much more IoT

[출처] https://www.forbes.com/sites/jaysondemers/2017/12/30/7-technology-trends-that-will-dominate-2018/#35ad78de57d7

4차 산업혁명
== 소프트웨어의 시대

#4차 산업혁명 #그딴 거 없다 #열심히 코딩하면 된다

왜 프로그래밍을 배워야 하는가?

소프트웨어 개발자가 아닌 우리가 배워야 하는 이유

프로그래밍을 배워야 하는 이유

1. 일자리 변화

- 소프트웨어 중심의 사회 변화에 대한 이해
  - 4차산업혁명에 따른 정보기술융합

- 데이터 주도 의사결정
  - 프로그래밍 이해가 있는 인력 선호

프로그래밍을 배워야 하는 이유

2. 프로그래밍 == MS Office

- 업무를 위한 필수 도구
  - 소프트웨어 시대의 필수 도구

- 프로그래밍을 통한 사무자동화
  - 업무 생산성의 차이

일반

삼성전자, 대졸공채에 '파이썬' 도입... "오픈소스로 생태계 선점"

최종수정 2018.11.19 12:08 기사입력 2018.11.19 12:08

댓글 쓰기

이번 하반기 공채부터 적용...비 개발 직군에도 파이썬 가능여부 질문

프로그래밍을 배워야 하는 이유

3. 프로그래밍 == 요리

중식당에서 짜장면을 먹지만,
가끔은 짜파게티 요리사.

전문적인 개발자가 되지 않더라도,
가끔은 내가 필요한 것을 직접 만들 수 있다.

Session 2
HACK YOUR LIFE

Session 2.
HACK YOUR LIFE

#소프트웨어 #Life #Work #Programmable

프로그래밍의 혜택
: Life Hacking, Work Hacking

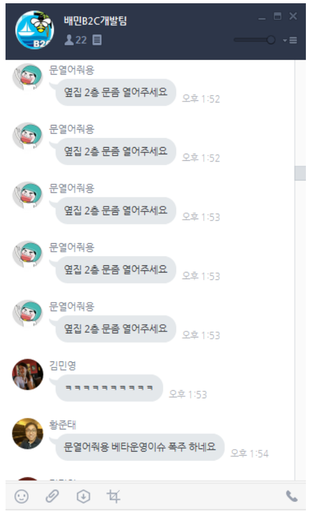
Life Hacking,

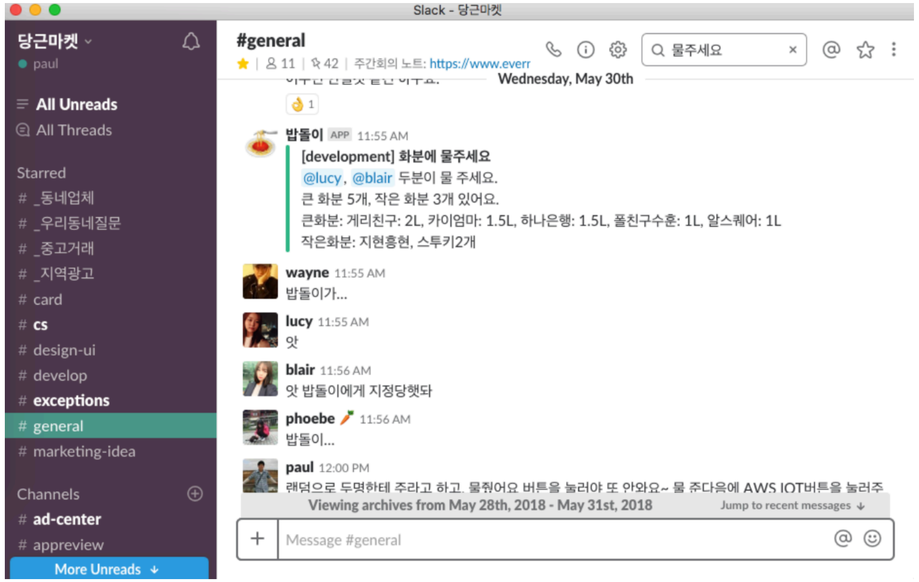
Programmable Life.

Delay start
Temperature
OK
Spin speed
Start
Stop
Tide
Cottons
Minimum iron
Woollens
Outerwea
Proofing
Expre
Further program

amazonbasics
Goldfish
STARBUCKS
DOUBLESHOT
NERF
BURT'S BEES
Doritos
Red Bull
ENERGY DRINK
smartwater
Tide

hphk.





Happy Hacking

https://www.youtube.com/1

https://www.youtube.com/2

기술 생산

기술 활용

기술생산 == 기술활용

Samsung
Microsoft
Oracle, SAP

기술 생산

기술 활용

기술 이해만 있다면 누구든 활용 가능

Google Tensorflow - Airbnb, Xiaomi
AWS - Pinterest, Netflix(All AWS), Samsung
IBM Watson - Legends, Minter Ellison

Open Source
: 모두에게 열려있는 (소스)코드

오픈 소스
Open Source

제작자의 권리를 지키면서,
누구나 코드 열람이 가능한,
오픈 소스 라이센스가 등장.

공동 참여 프로젝트들이 생겨나는 등의
생태계가 구성 됨.

실제 오픈소스 활용 예시 : 카카오톡

수레바퀴를
두 번 만들지 마라

모든 것을 바닥부터 만들지 않아도 된다.

거인의 어깨 위에서
프로그래밍 시작하기

이미 만들어진 것을 잘 활용하면 된다.

그리고,
인생을 즐기는 도구로
활용하자

요약

1. 소프트웨어 시대의 필수 도구
2. Life Hacking
3. 프로그래밍이 쉬워졌다.

Session 3
Python 기초

Session 3.
Python 기초

#LifeIsShort #YouNeedPython

컴퓨터
프로그래밍
언어

기술을 사용하기 위한, 결과물을 만들기 위한 기초 학습.

컴퓨터(computer)

컴퓨터(computer)
= 저장, 계산

프로그램

프로그램

컴퓨터 프로그램(영국 영어: computer programme, 미국 영어: computer program, 문화어: 콤퓨터프로그람)은 컴퓨터에 의해 실행되는 **지시사항**의 모음인 컴퓨터 소프트웨어의 한 예이다. 대부분의 프로그램은 실행 중(즉, 명령어를 '불러들일' 때)에 사용자의 입력에 반응하도록 구현된 **명령어**의 집합으로 구성되어 있다. 대부분의 프로그램들은 하드디스크 등의 매체에 바이너리 형식의 파일로 저장되어 있다가 사용자가 실행시키면 메모리로 적재되어 실행된다.

출처 위키피디아

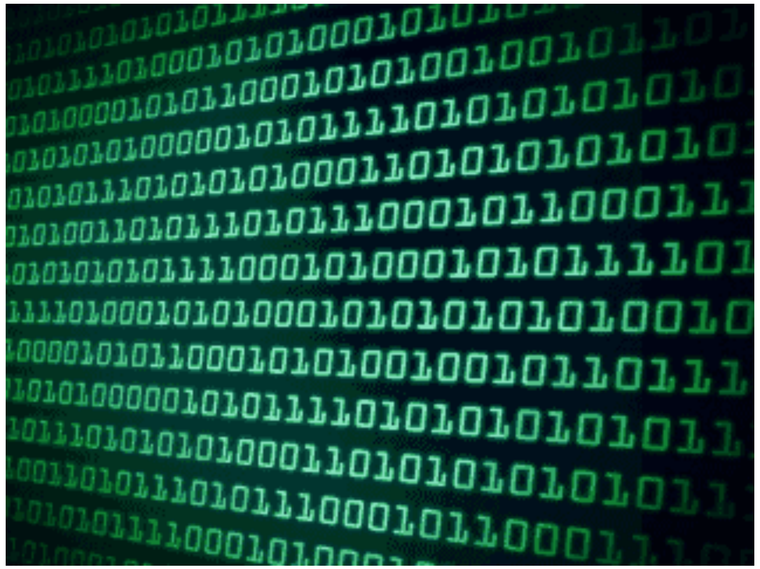
프로그래밍
= 명령하는 거, 지시하는 거
= 뭘 시키는 것

컴퓨터 프로그래밍 언어
= 컴퓨터에게 뭔가를 시킬 때 쓰는 말

그러면 뭘 시킬수 있는데요??

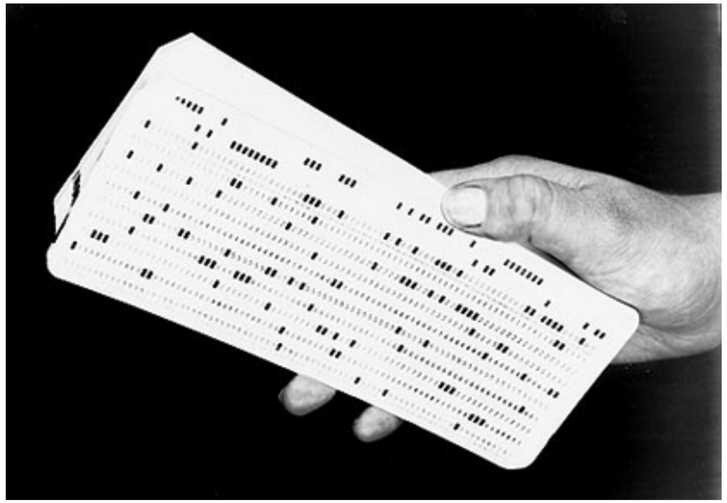
당신이 상상하는
(컴퓨터 한테 시킬 수 있는)
모든 일

컴퓨터 프로그래밍 언어
: 원래는 아주 어려웠음..(๑˃̵ᴗ˂̵)



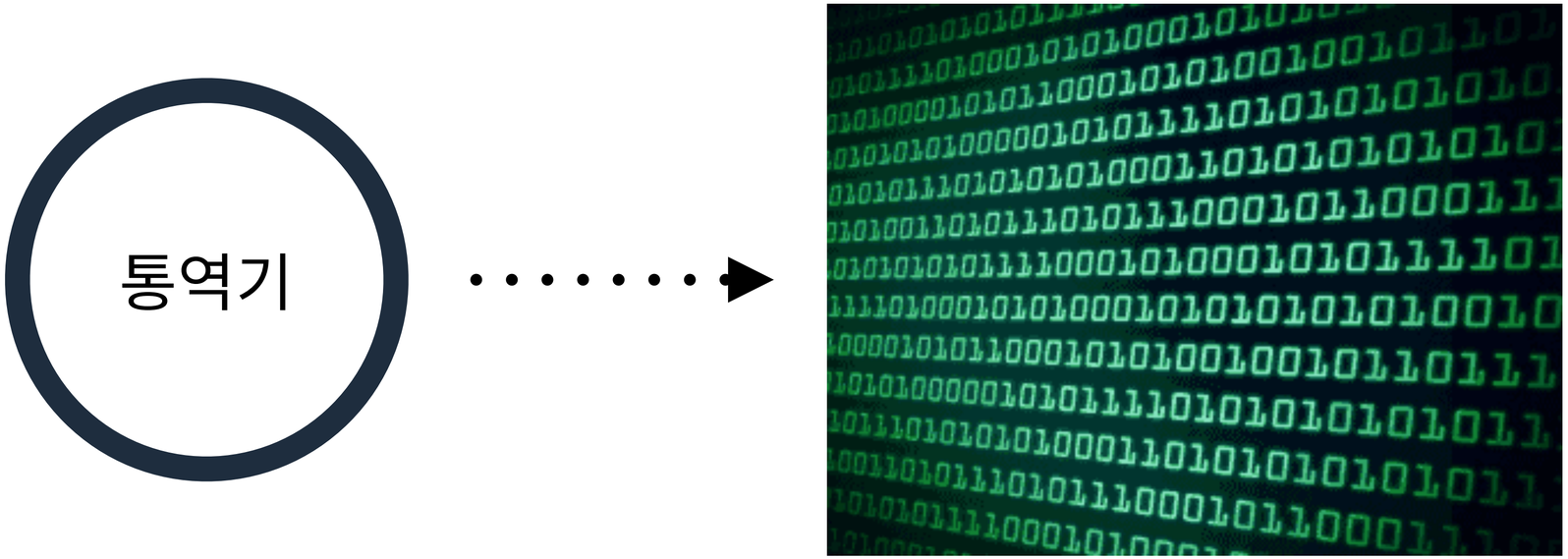
컴퓨터가 이해하는 언어

????



옛날 옛적..

컴퓨터한테 뭐라도 시키려면 이 고생을…



통역기를 사용하기 시작!

컴퓨터한테 얘기하기가 쉬워짐

```
.386
.model flat, c
.stack 4096
.data
    message db "Hello, World!", 0Dh, 0Ah, 00h

.code
    extern _heap_init: proc
    extern _mtinit: proc
    extern _ioinit: proc

    extern exit: proc
    extern printf: proc

    main proc
        ; _heap_init(1)
        push 1
        call _heap_init
        add esp, 4

        ; _mtinit(1)
        push 1
        call _mtinit
        add esp, 4
        ; _ioinit()
        call _ioinit

        ; main()

        push offset message
        call printf
        add esp, 4

        ; exit(0)
        push 0
        call exit
        add esp, 4

        ; return 0
        xor eax, eax
        ret
    main endp

    end main
```

어셈블리어

외계어

Link: http://codepad.org/QOmLUb6y [raw code | output | fork]

C, pasted 1 second ago:

```c
#include <stdio.h>

int main()
{
printf("Hello World");
return 0;
}
```

Output:

```
Hello World
```

C언어

이해 될 듯, 말 듯

print(“Hello World”)

Python

거의 영어

"Life is short,
You need Python."

-- Bruce Eckel

파이썬(Python)

1. 쉽다.
2. 많은 사람들이 사용한다.
3. 많은 것을 할 수 있다.

컴퓨터 프로그래밍 언어

언어?

=

언어?
= 약속

문법(文法)

문장을 [ ] 법(약속)

문법(文法)

문장을 [만드는] 법(약속)

영어 : 5형식

S + V

S + V + C

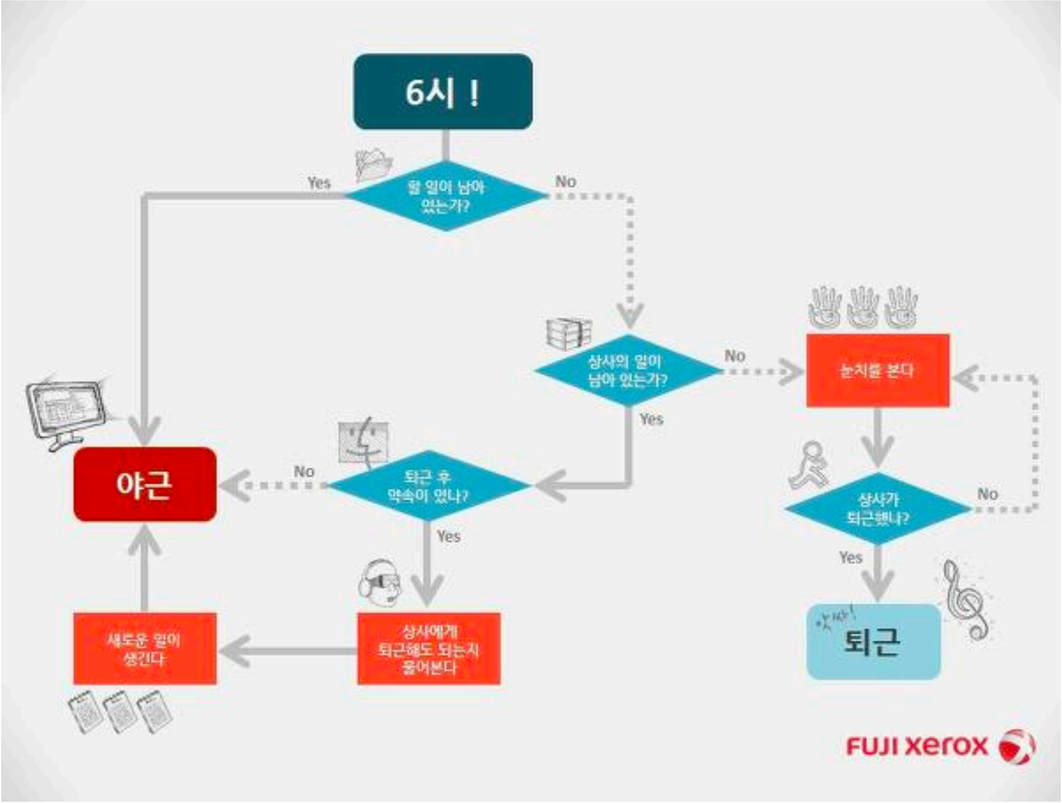
S + V + O

S + V + O + O

S + V + O + C

프로그래밍 언어 : 3형식

1. 저장
2. 조건(if)
3. 반복(while)



프로그래밍 언어의 모든 것

알고리즘 : 이미 다 알고 있음

파이썬을 왜 쓴다?

파이썬을 왜 쓴다?

1. 다른 언어보다 쉽다.
2. 다양한 부분에서 활용한다.
3. 그래서 많은 사람들이 쓴다.

본격적으로 (파이썬 코드를 활용해)
컴식이에게 일을 시켜보자

gm.py.hphk.io

시작하기 전에 명심할 것!

1. 대/소문자
2. 띄어쓰기 및 들여쓰기(indent)
3. 스펠링

Python의 문법

1. 저장
2. 조건
3. 반복

1. 저장

- 저장의 개념은? Save?
- 무엇을, 어떻게 저장할 수 있는가?

박스를 떠올리세요

dust = 60

= 60

dust 는 60이다 (X)

dust 에 60을 저장한다 (O)

== 60

dust == 60

dust 는 60과 같다.

1. 저장

- 저장의 개념은? Save?
- 무엇을, 어떻게 저장할 수 있는가?

무엇을 저장하는가

컴퓨터는 무엇을 인식할 수 있나?
박스에 담을 수 있는 것들

무엇을 저장하는가

1) 숫자
2) 글자
3) 참/거짓

1. 현실세계에 존재하는 모든 숫자.
(글자가 들어가면 안됨.)
2. 기본적인 연산이 가능. (계산기)

O : 12 -365 3.14159265358
X : 23ab ba123

무엇을 저장하는가

1) 숫자
2) 글자
3) 참/거짓

1. 현실세계에 존재하는 모든 글자.

2. 따옴표로 둘러싼 글자 or 숫자.

“미세먼지”
“58도 글자입니다.”
“글자는 반드시 따옴표를 붙여야합니다!”

58
숫자

VS

“58”
글자

무엇을 저장하는가

1) 숫자
2) 글자
3) 참/거짓

True, False

단 두 가지!

조건 / 반복에 사용됨.

300 > 200 => True

150 == 161 => False

실습 1

1) 변수 활용하기
- (공손하게) 여러번(5번) 인사하는 챗봇

어떻게 저장하는가.

다양한 종류의 박스

어떻게 저장하는가

1) 변수 (variable)

1. 저장된 값을 변경 할 수 있는(variable) 박스.
2. 숫자, 글자, 참거짓을 담을 수 있다.

```
dust = 58
dust = 60
print(dust)
```

어떻게 저장하는가

1) 변수 (variable)

1. 저장된 값을 변경 할 수 있는(variable) 박스.
2. 숫자, 글자, 참거짓을 담을 수 있다.

```
dust = 58
dust = 60
print(dust)
```

```
60
```

dust = 60 → 숫자

input = “미세먼지” → 글자

today = True → 참/거짓

print(hello)

hello 변수에 담긴 내용을 출력해주세요.

VS

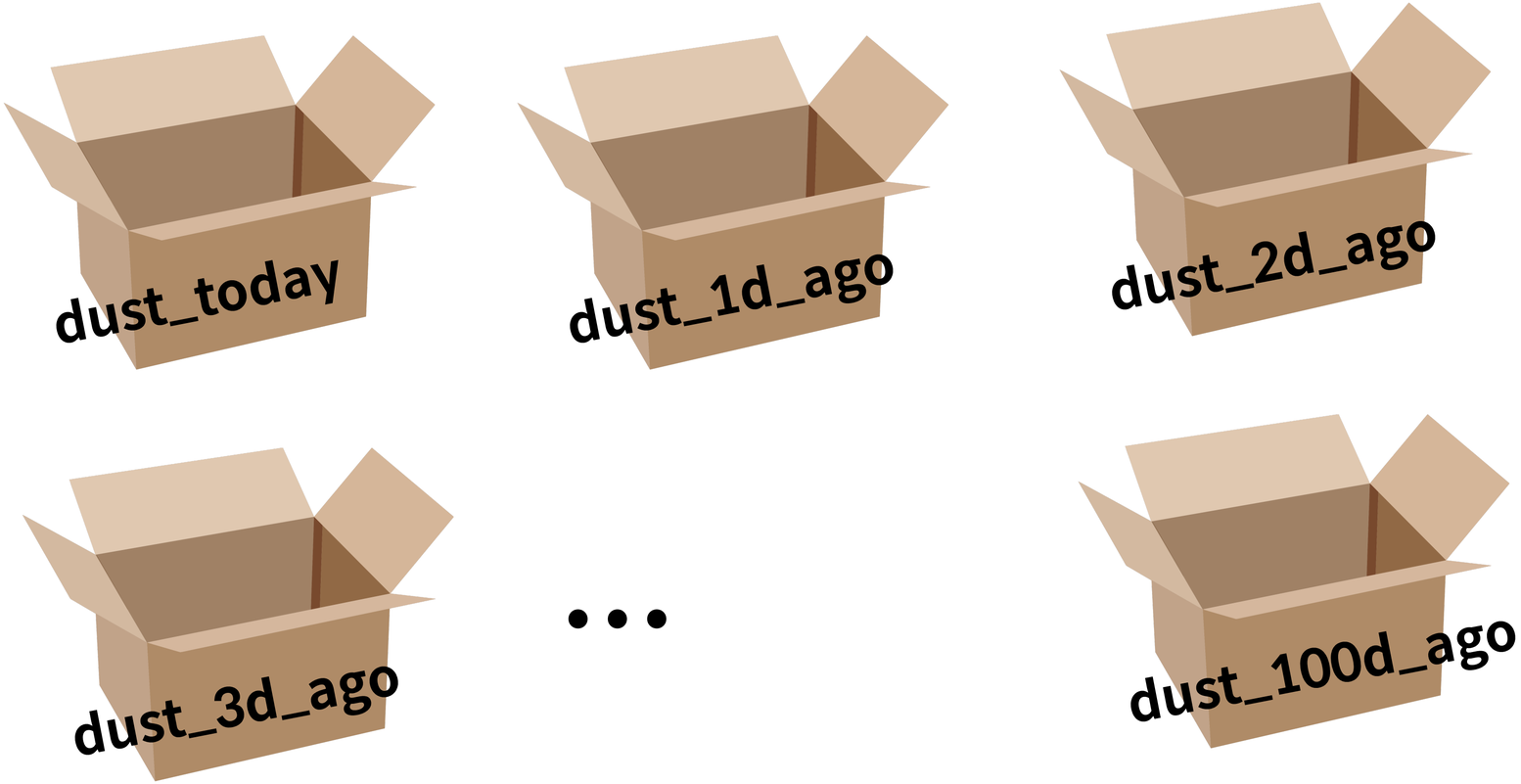
print(“hello”)

“hello”라는 글자를 출력해주세요.

3일치의 정보를 저장한다면?

dust_today
dust_1d_ago
dust_2d_ago

100일치의 정보를 저장한다면?


dust_today
dust_1d_ago
dust_2d_ago
dust_3d_ago
...
dust_100d_ago

코드로 작성해보면,

dust_today = 58

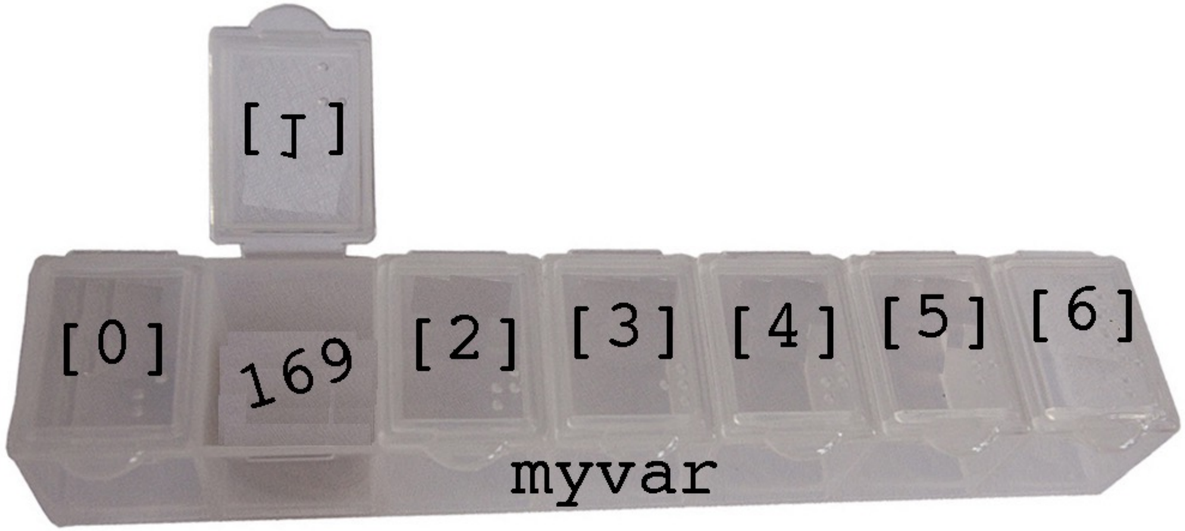
dust_1d_ago = 40

dust_2d_ago = 70

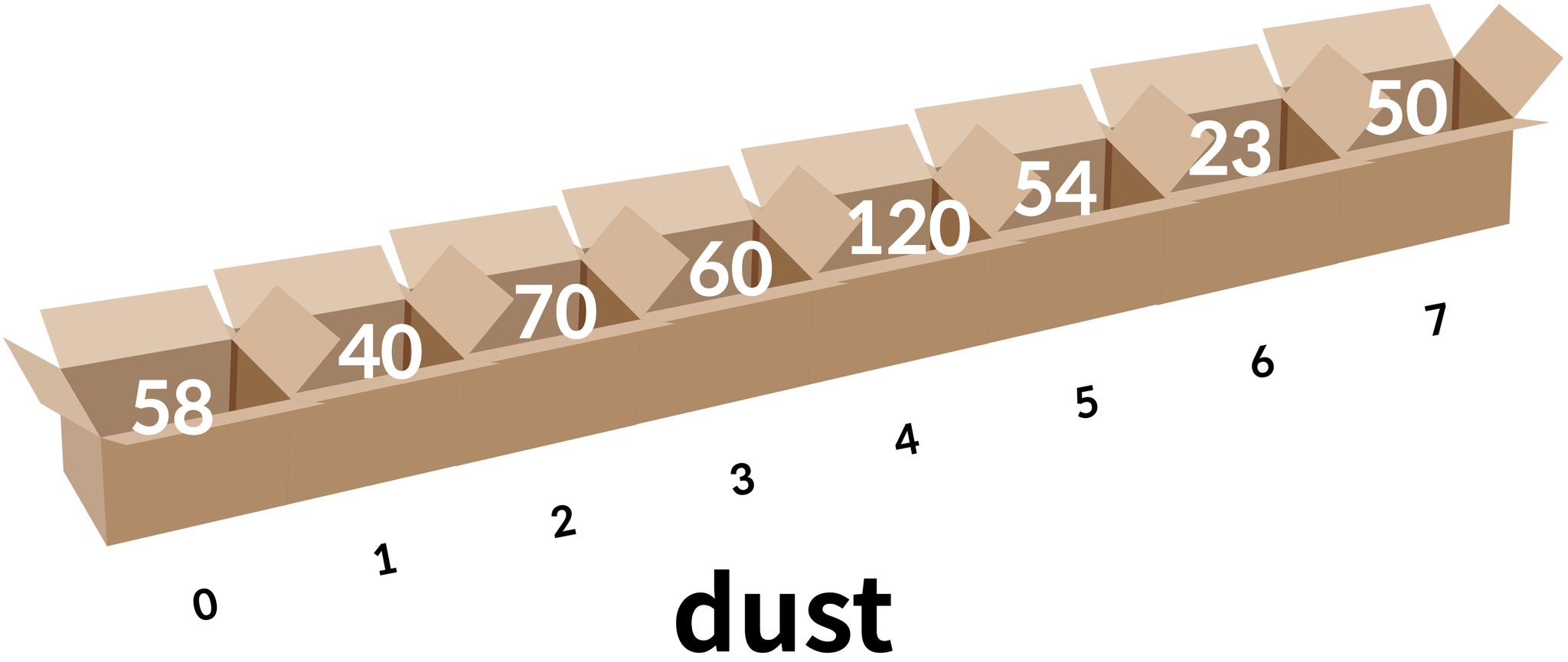
…

dust_100d_ago = 66

더 편한 방법은 없을까?



박스들을 붙이자.



dust

어떻게 저장하는가

2) 리스트 (List)

박스의 리스트(리스트)

박스가 순서대로 여러 개가 붙어있다.

```
dust = [58, 40, 70]
print(dust[1])
```

어떻게 저장하는가

2) 리스트 (List)

박스의 리스트(리스트)

박스가 순서대로 여러 개가 붙어있다.

```
dust = [58, 40, 70]
print(dust[1])
```

```
40
```

location = [“영등포구”, “강남구”, …]

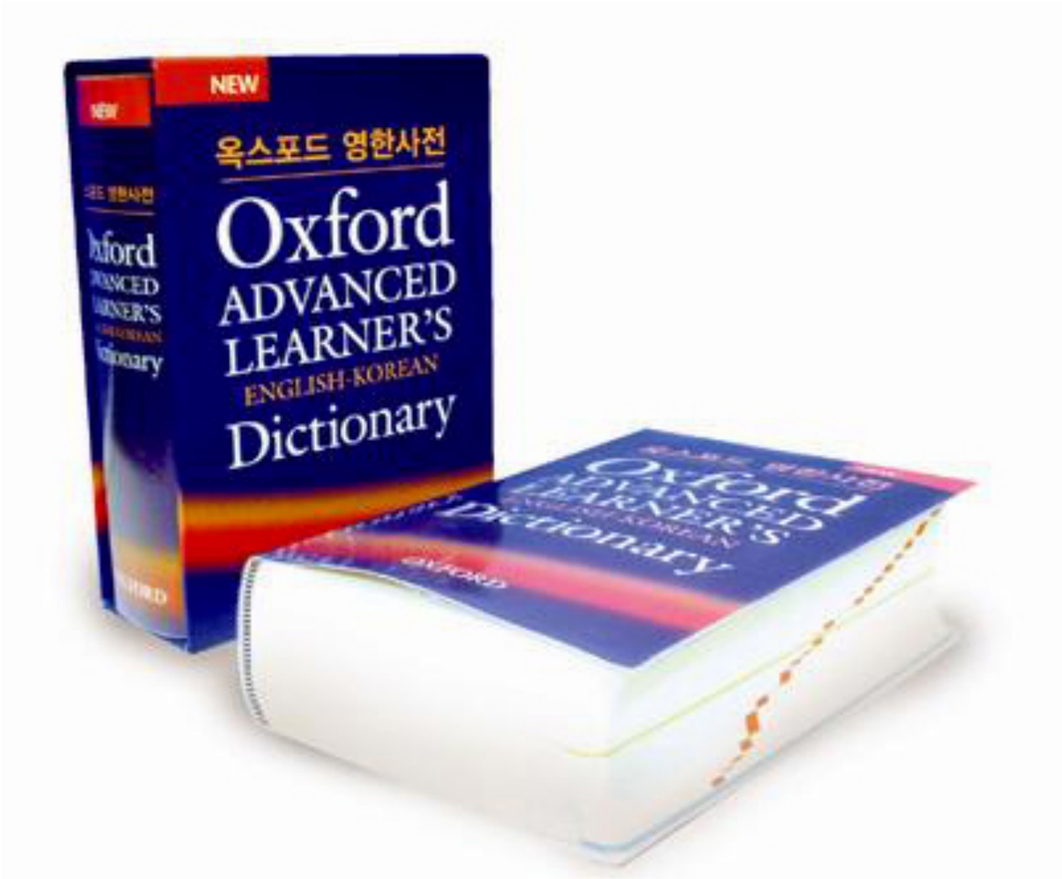
value = [58, 40, …]

더 편한 방법은 없을까?

Master
Bedroom
Kids'
Room
Dishes
Attic
Kitchen
Bathroom
HALL CLOSET
Living-
room
Toys

견출지를 붙여보자

dust


NEW
옥스포드 영한사전
Oxford
ADVANCED
LEARNER'S
ENGLISH-KOREAN
Dictionary

어떻게 저장하는가 3) 딕셔너리 (dictionary)

궁극의 박스 dictionary

“견출지 붙인 박스들의 묶음”

```
dust = {“영등포구” : 58, “강남구”: 40}
print(dust[“영등포구”])
```

어떻게 저장하는가 3) 딕셔너리 (dictionary)

궁극의 박스 dictionary

“견출지 붙인 박스들의 묶음”

```
dust = {“영등포구” : 58, “강남구”: 40}
print(dust[“영등포구”])
```

```
58
```

	저장하는 법	불러오는 법	출력값
변수 (박스 1개)	dust = 40	dust	40
리스트 (박스 여러개)	dust = [40, 50, 80]	dust[0]	40
딕셔너리 (견출지 붙인 박스)	dust = {“영등포구”: 40, “강남구”: 50}	dust[“영등포구”]	40

	저장하는 법	불러오는 법	출력값
변수 (박스 1개)	dust = 40	dust	40
리스트 (박스 여러개)	dust = [40, 50, 80]	dust[0]	40
딕셔너리 (견출지 붙인 박스)	dust = {“영등포구”: 40, “강남구”: 50}	dust[“영등포구”]	40

실습 2

1) 리스트 활용
   - 점심 메뉴 담기

menu
리스트

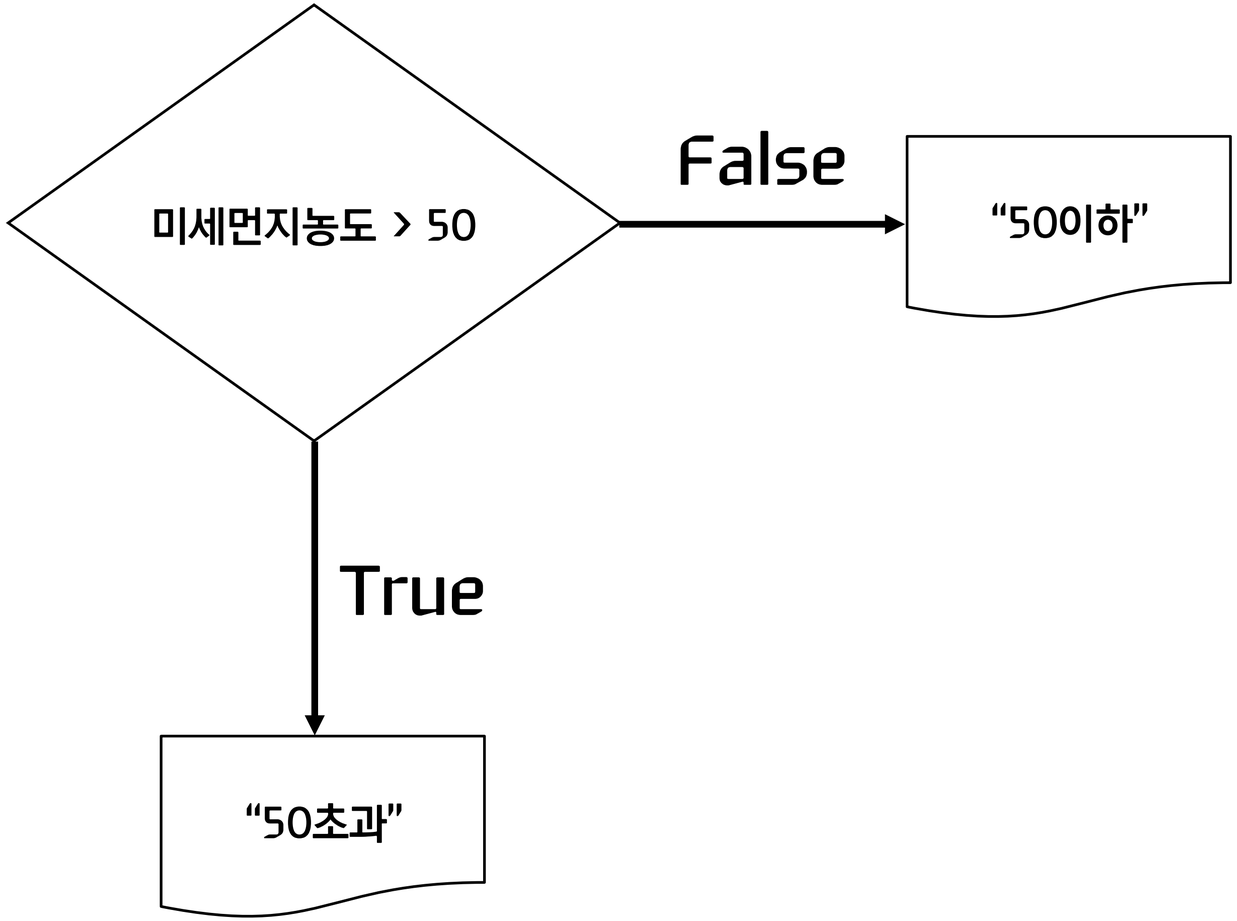
menu = [ ]

원하는 메뉴를
[ ] 안에 넣어주세요

2. 조건

```
if True:
    print("조건문입니다.")
```

조건
if/else



조건
if/else

```
if dust > 50:
  print("50초과")
else:
  print("50이하")
```

조건
if/else

```
if dust > 50:
  print("50초과")
else:
  print("50이하")
```

조건
if/else

```
if dust > 50:
    print("50초과")
else:
    print("50이하")
```

조건

if/else

dust = 60

```
if  60  > 50: True
    print("50초과")
else:
    print("50이하")
```

50초과

조건

if/else

dust = 40

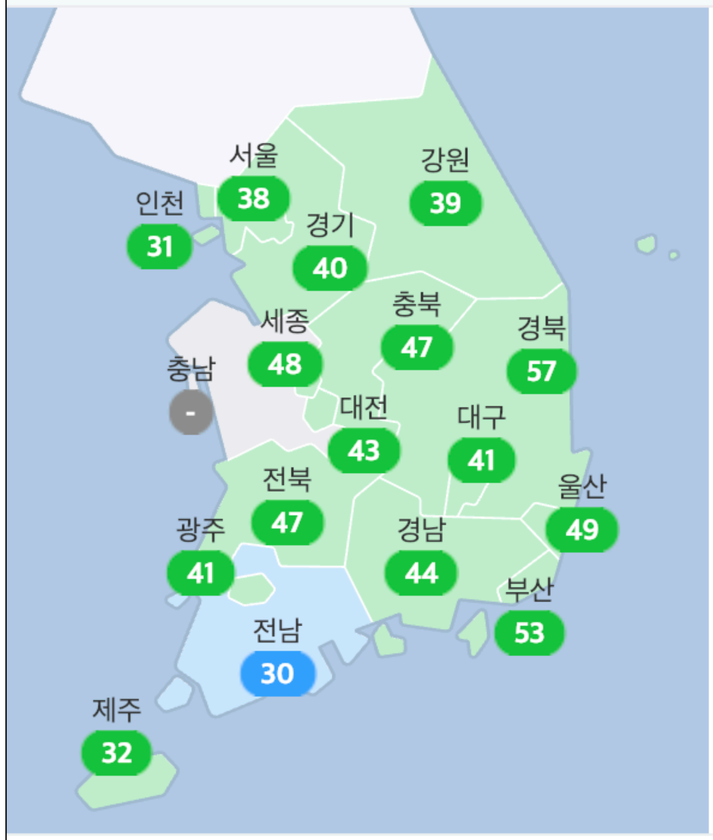
```
if  40  > 50:    False
  print("50초과")
else:
  print("50이하")
```

50이하

hphk.

미세먼지 | 초미세먼지 | 오존 | 자외선 | 황사

현재 | 내일 | 모레

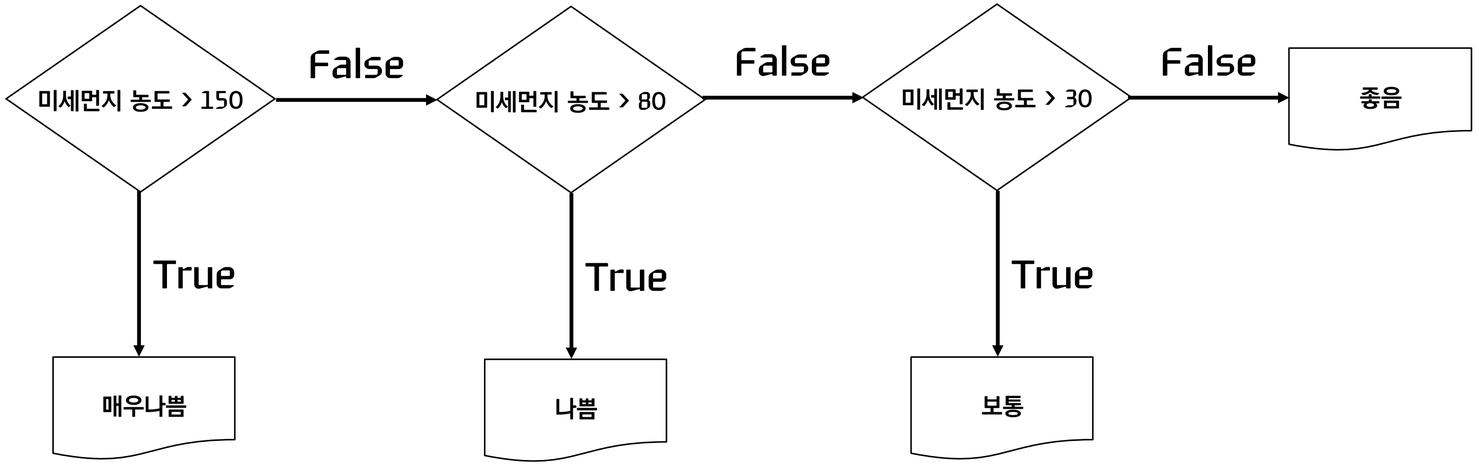


관측지점	현재	오전예보	오후예보
서울	38	보통	보통
경기	40	보통	보통
인천	31	보통	보통
강원	39	보통	보통
세종	48	보통	보통
충북	47	보통	보통
충남	-	보통	보통
대전	43	보통	보통

좋음 0 | ~30 보통 | ~80 나쁨 | ~150 매우나쁨 | 151~

Happy Hacking

조건
if/elif/else



조건
if/elif/else

```
if dust > 150:
    print("매우나쁨")
elif 80 < dust <= 150:
    print("나쁨")
elif 30 < dust <= 80:
    print("보통")
else:
    print("좋음")
```

미세먼지농도	등급
151~	매우나쁨
81 ~ 150	나쁨
31 ~ 80	보통
0 ~ 30	좋음

```
if dust > 150:
    print("매우나쁨")
```

1. dust라는 변수에 농도가 API로부터 받은 정보가 저장되어있다.
2. 기상청의 미세먼지 농도 등급은 아래와 같다.

Q. 조건문과 출력하는 법을 통하여 코드를 작성하라.

Ex) 55, 보통

미세먼지농도	등급
151 ~	매우나쁨
81 ~ 150	나쁨
31 ~ 80	보통
0 ~ 30	좋음

<Hint>

만약 (미세먼지 농도가 150보다 크면):

    출력하자(“매우나쁨”)

아닌데 만약에 (미세먼지 농도가 150이하 80초과면):

    출력하자(“나쁨”)

아닌데 만약에 (미세먼지 농도가 80이하 30초과면):

    출력하자(“보통”)

아니면:

    출력하자(“좋음”)

실습 3

1) 조건문 활용하여 미세먼지 코드 완성하기

```python
if(150 < dust):
  print("매우나쁨")
elif(80 < dust <= 150):
  print("나쁨")
elif(30 < dust and dust <= 80):
  print("보통")
else:
  print("좋음")
```

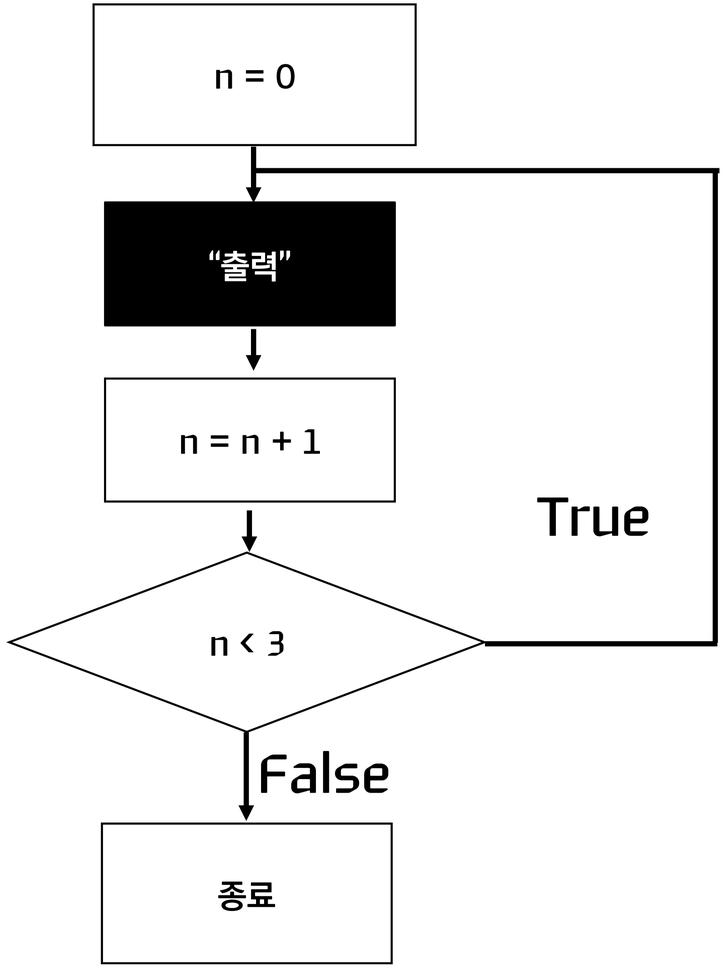
3. 반복

```
while True:
    print("계속해주세요.")
```

3. 반복

```
while True:
    print("계속해주세요.")
```

반복
while



반복
while

while 에 해당하는 조건일 동안 계속 반복.

```
n = 0
while n < 3:
    print("출력")
    n = n+1
```

?

반복

while에 해당하는 조건일 동안 계속 반복.

while

```
n = 0
while n < 3:
    print("출력")
    n = n + 1
```

3 < 3 False

```
출력
출력
출력
```

반복 while

while에 해당하는 조건일 동안 계속 반복.

```
dust = [59, 24, 102, 45, 64]
```

```
n = 0
while n < 3:
    print(dust[n])
    n = n + 1
```

3 < 3 False

```
59
24
102
```

반복 for

```
for i in List:
    print("정해진 범위 안에서 계속")
```

반복 for (리스트)

정해진 박스 내에서의 반복 시 사용

'가지고 있는 모든 것을 꺼낸다'

```
dust = [59, 24, 102]
for i in dust:
    print(i)
```

i = 102

print(102)

```
59  24  102
```

while True:

조건이 True일 때까지
반복적으로 실행 되기에
종료조건이 반드시 필요

VS

for i in dust:

정해진 범위를 반복하기에
종료조건이 필요 없음.

실습 4

- 반복문 활용하기(for)
  - 여러번 인사하기

우리 프로그램은 아주 잘 동작한다.
그런데.. 아까부터 살짝 맘에 걸리는게 있는데..?

이해하지 못했던 바로 그 코드
도대체 dust 는 어디서 왔지?

미세먼지 데이터는 어디서 왔을까

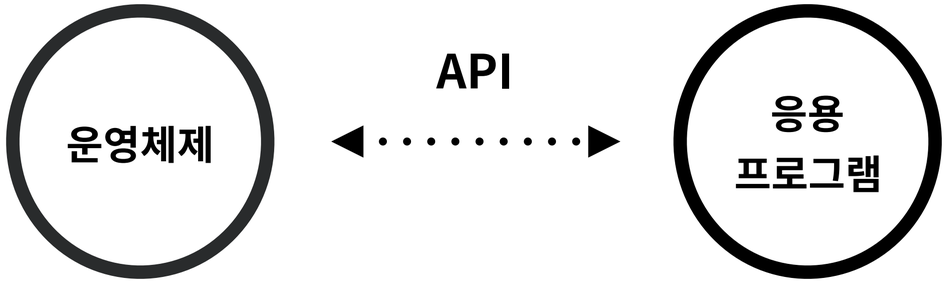
웹(WEB) API를 통해

API

: 다른 시스템 간의

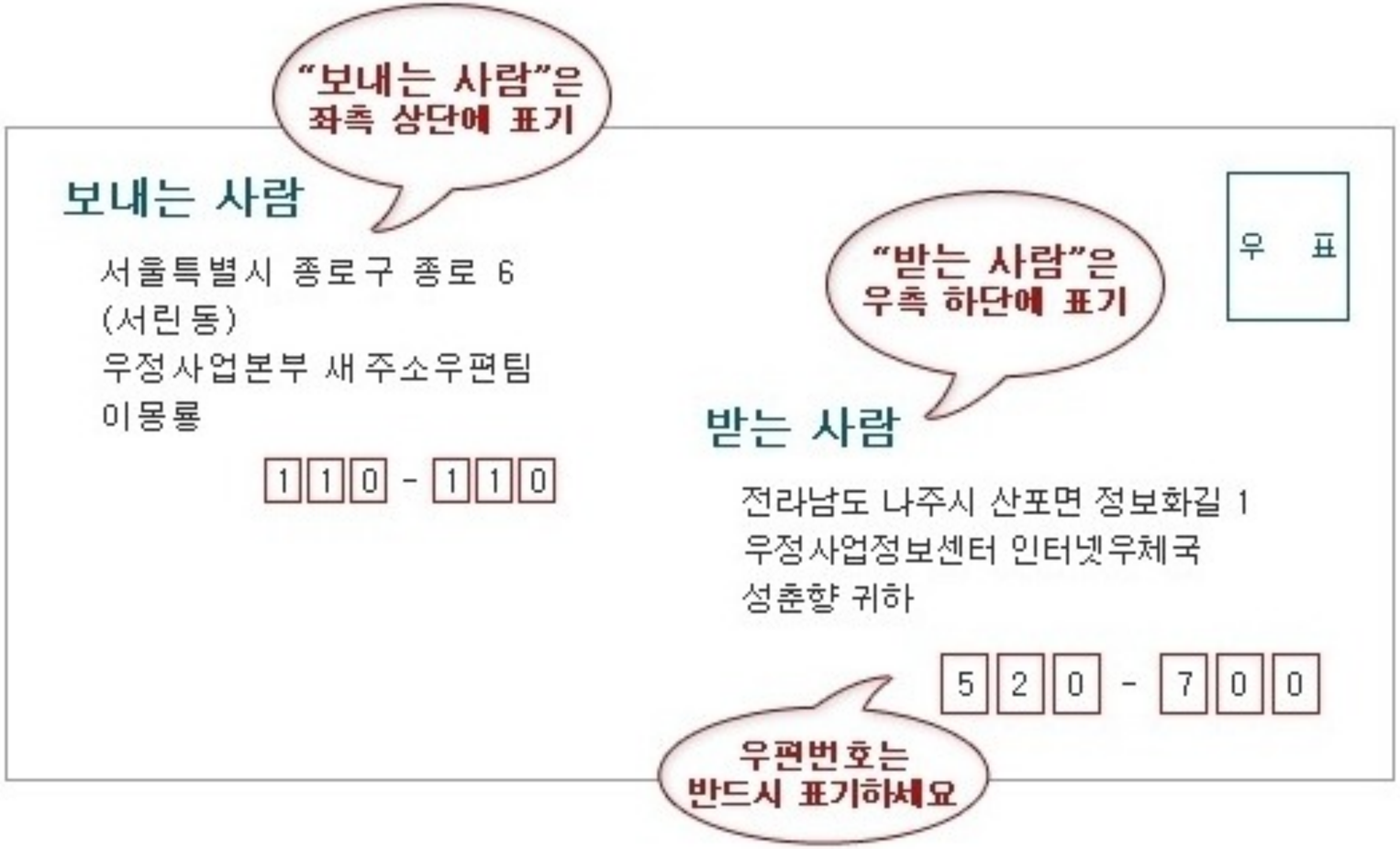
커뮤니케이션 방식

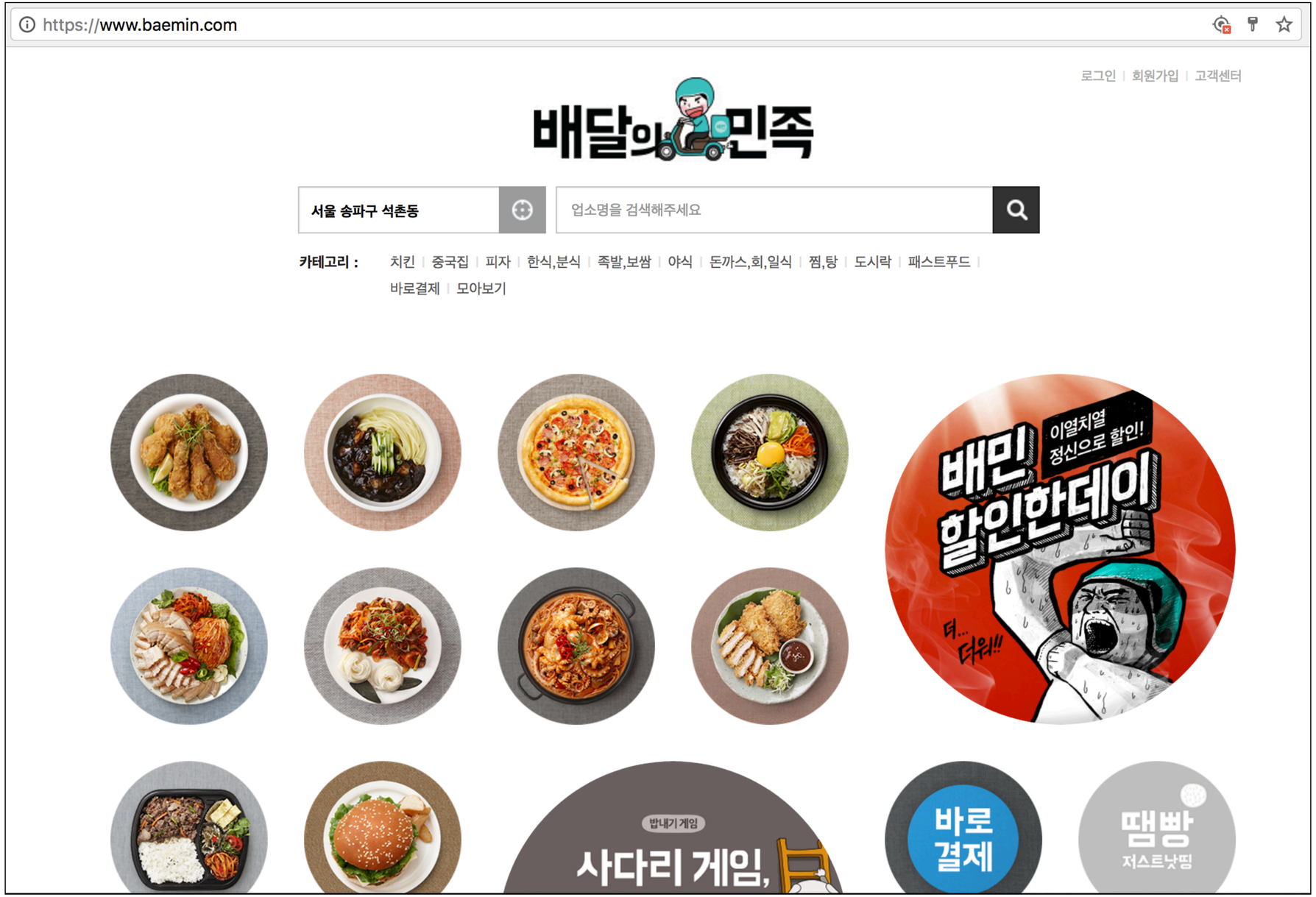
API란?



API(Application Programming Interface, 응용 프로그램 프로그래밍 인터페이스)

응용 프로그램에서 사용할 수 있도록,
운영 체제나 프로그래밍 언어가 제공하는 기능을
제어할 수 있게 만든 인터페이스를 뜻한다.
출처 - 위키백과


"보내는 사람"은
좌측 상단에 표기
보내는 사람
서울특별시 종로구 종로 6
(서린동)
우정사업본부 새주소우편팀
이몽룡
1 1 0 - 1 1 0
"받는 사람"은
우측 하단에 표기
우 표
받는 사람
전라남도 나주시 산포면 정보화길 1
우정사업정보센터 인터넷우체국
성춘향 귀하
5 2 0 - 7 0 0
우편번호는
반드시 표기하세요


https://www.baemin.com
로그인 | 회원가입 | 고객센터
배달의 민족
서울 송파구 석촌동
업소명을 검색해주세요
카테고리 : 치킨 | 중국집 | 피자 | 한식,분식 | 족발,보쌈 | 야식 | 돈까스,회,일식 | 찜,탕 | 도시락 | 패스트푸드 |
바로결제 | 모아보기
이열치열
정신으로 할인!
배민
할인한데이
더... 더워!!
밥내기 게임
사다리 게임,
바로
결제
땜빵
저스트낫띵

https://www.baemin.com/oauthmember/signup_step1

STEP 1 STEP 2

- 배달의민족은 배민ID로 로그인하실 수 있습니다.
- 배민ID는 이메일가입 또는 페이스북 가입이 가능합니다.
- 기존 배달의민족 앱을 이용하신 회원은 [이미 배민ID가 있으세요?] 버튼을 눌러주세요.
- 사용하고 있는 이메일을 입력해야 아이디/비밀번호 분실 시 찾을 수 있습니다.

배민ID 가입하기 이미 배민 ID가 있으세요?

배민ID(이메일)

비밀번호 (8~20자)

비밀번호 재입력 (8~20자)

다음

간편하게 가입하기

페이스북으로 시작하기

홍길동이 페이스북에 가입된 사람인지
확인 부탁드리며, 정보를 부탁드립니다.

보내는 사람
배달의민족

받는 사람
Facebook

Firstname : Gildong
Lastname : Hong
Email : honggildong@gmail.com

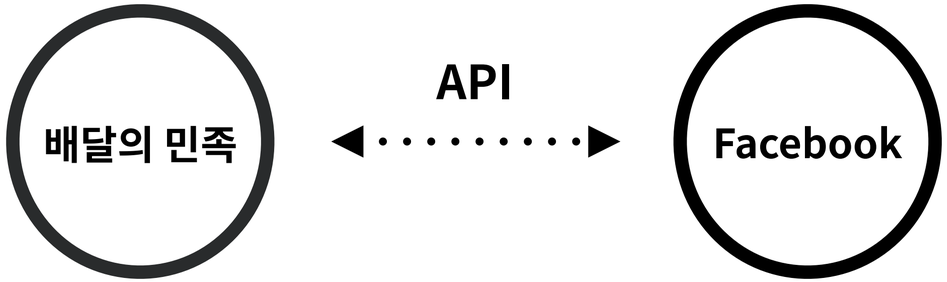
Firstname : Gildong
Lastname : Hong
Email : honggildong@gmail.com

보내는 사람
Facebook

받는 사람
배달의민족

로그인!

웹에서 API



- 서비스와 서비스 간의 대화 방식
- 요청을 받는 측에서 일정한 방식을 명세
  => 해당 방식대로 요청을 하면 정보 제공

API 예시

1. 페이스북 로그인
2. 카카오톡 API
3. 네이버 지도
4. Riot (League of Legends) API
5. 공공데이터 API

API 예시

1. 페이스북 로그인
2. 카카오톡 API
3. 네이버 지도
4. Riot (League of Legends) API
5. 공공데이터 API

웹(WEB)에서의 커뮤니케이션 방식

요청(request)와 응답(response)

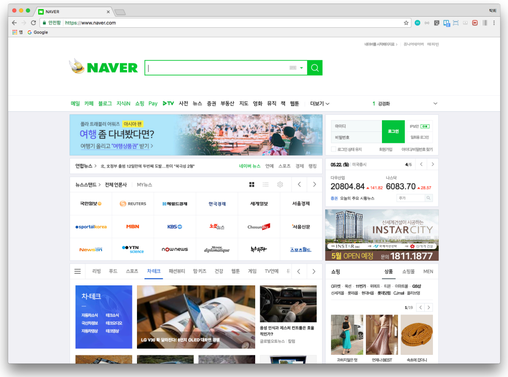
‘https://www.naver.com'

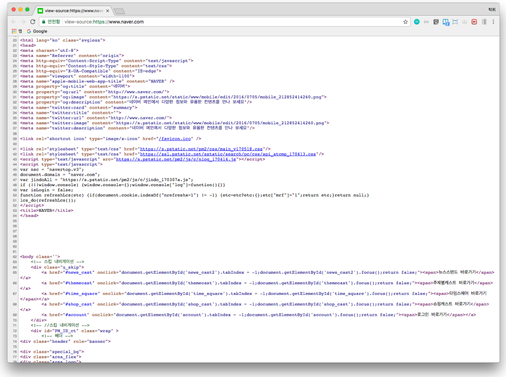
요청
(정보를 원하는 사람)

주소(Url)

문서
(HTML, XML, Json 등)

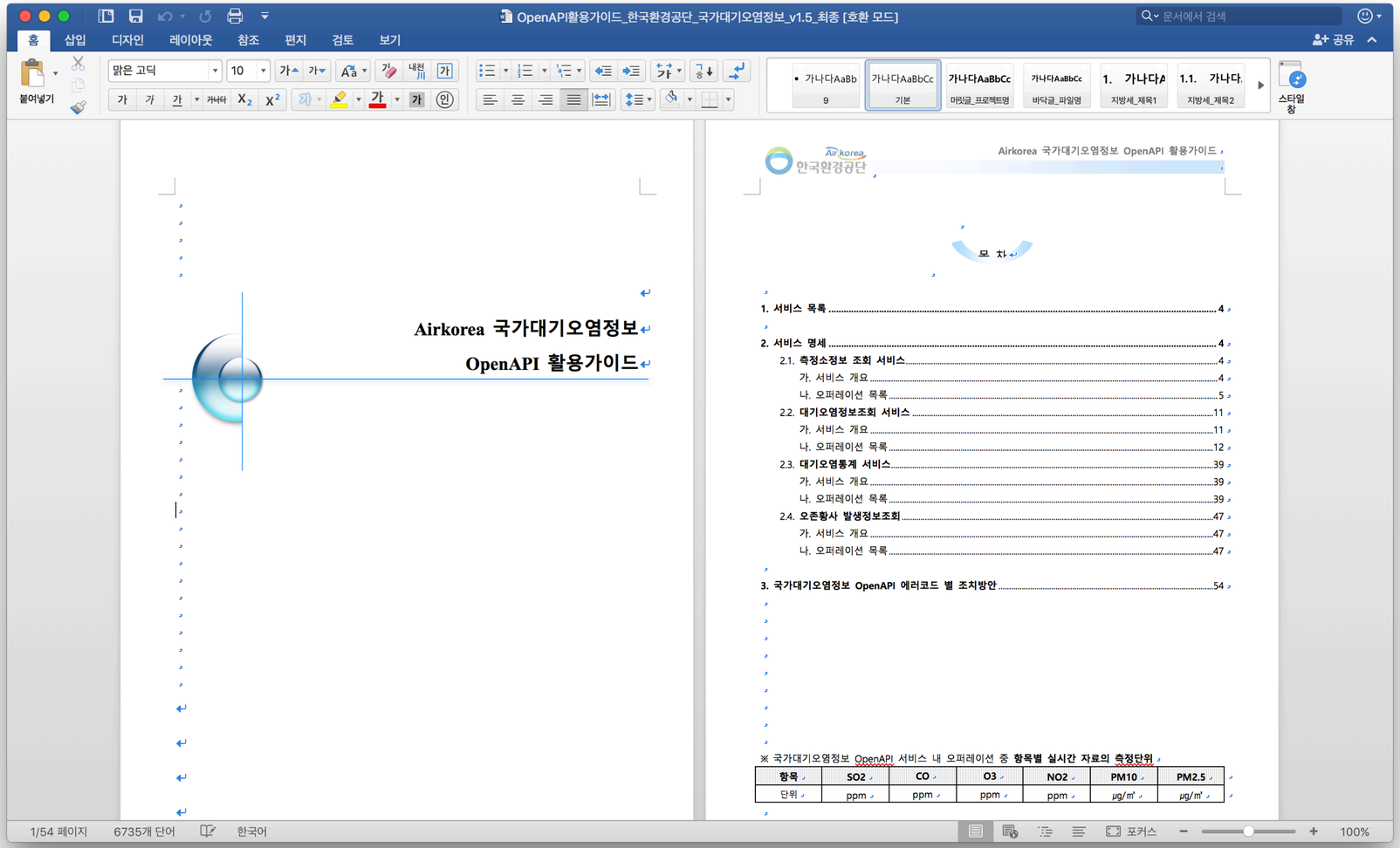
응답
(정보를 주는 사람)





요청
(사용자)
주소(URL)
문서(XML)
응답
(미세먼지 공공데이터)

hphk.



“무려 54page의 API 활용 가이드”

Happy Hacking

기본정보

	만드는 법
서비스명	대기오염정보 조회 서비스
문서	XML, JSON
메세지 교환 유형	요청(request) - 응답(response)
데이터 갱신주기	실시간 정보 : 10분, 대기질 예보 정보 : 매 시간 22분/57분
오퍼레이션	측정소별 실시간 측정정보 조회 통합대기환경지수 나쁨 이상 측정소 목록조회 시도별 실시간 측정정보 조회 미세먼지/오존 예보통보 조회 시도별 실시간 평균정보 조회

요청하는 법

변수	정보
numOfRows	한 페이지 결과 수
pageNo	페이지 번호
stationName	측정소 이름
dataTerm	요청 데이터 기간
ver	버전

http://openapi.airkorea.or.kr/openapi/services/rest/ArpltnInforInqireSvc/getMsrstnAcctoRltmMesureDnsty

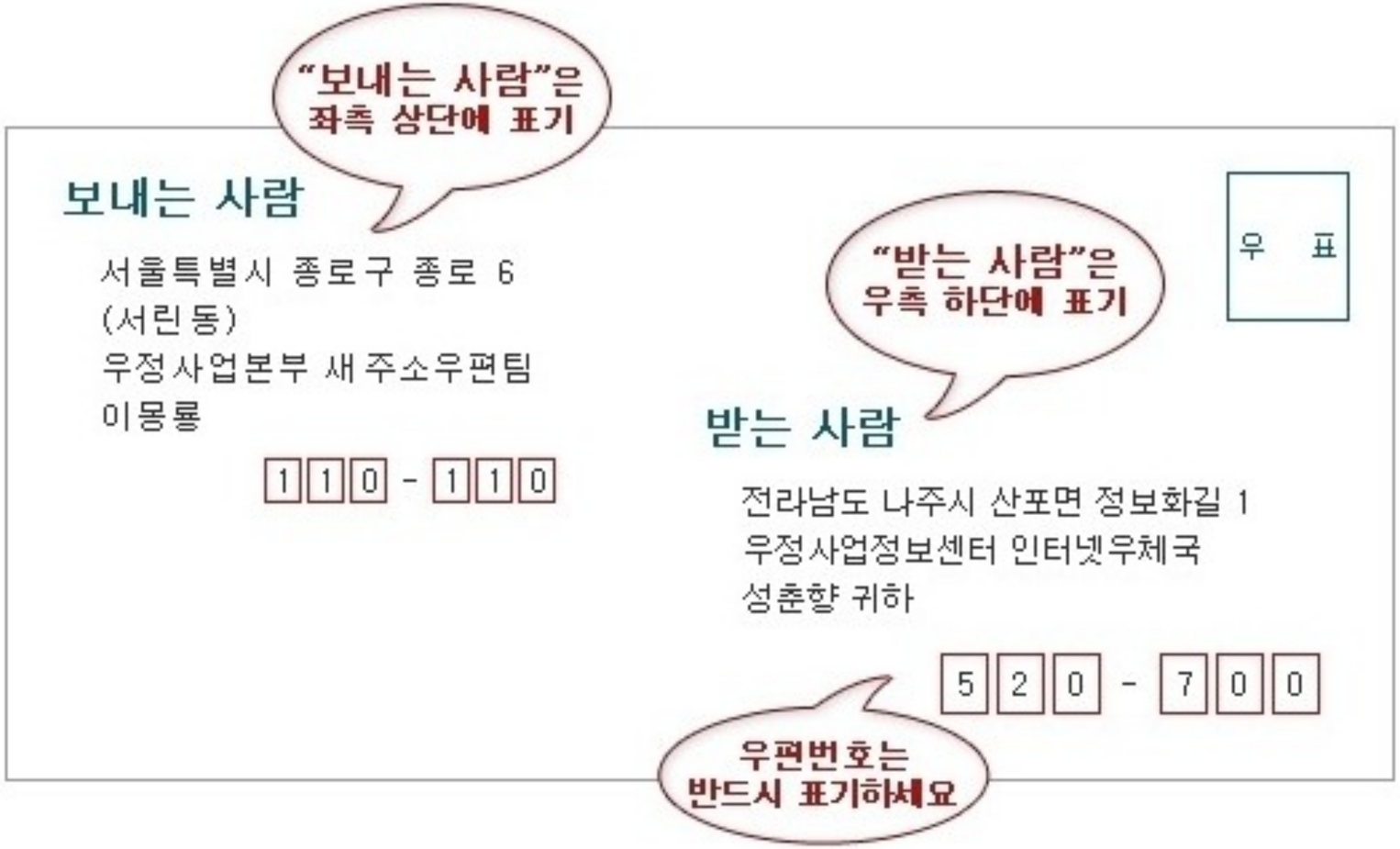
?stationName=3공단

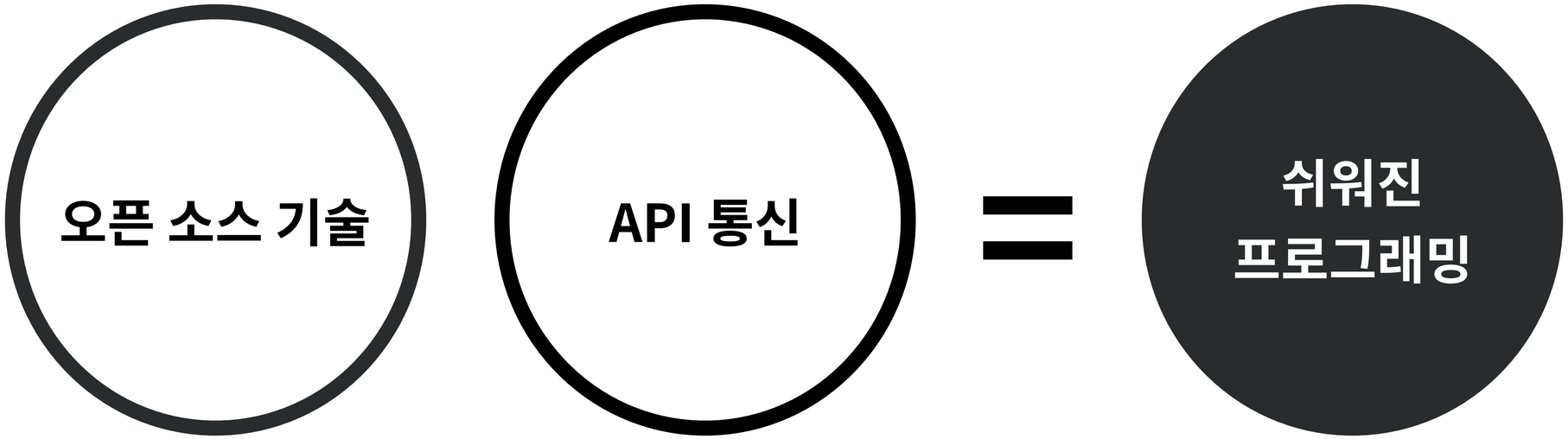
&dataTerm=MONTH

&numOfRows=100

&ServiceKey=sfE57qPQ5x1eAXU41ftcyXYJZK5eKEONTGzz%2FdCe069VtqGX4UwcpLZbJhXSayA%2FtauuMhXPXttKeoxuafYWPQ%3D%3D

&ver=1.6

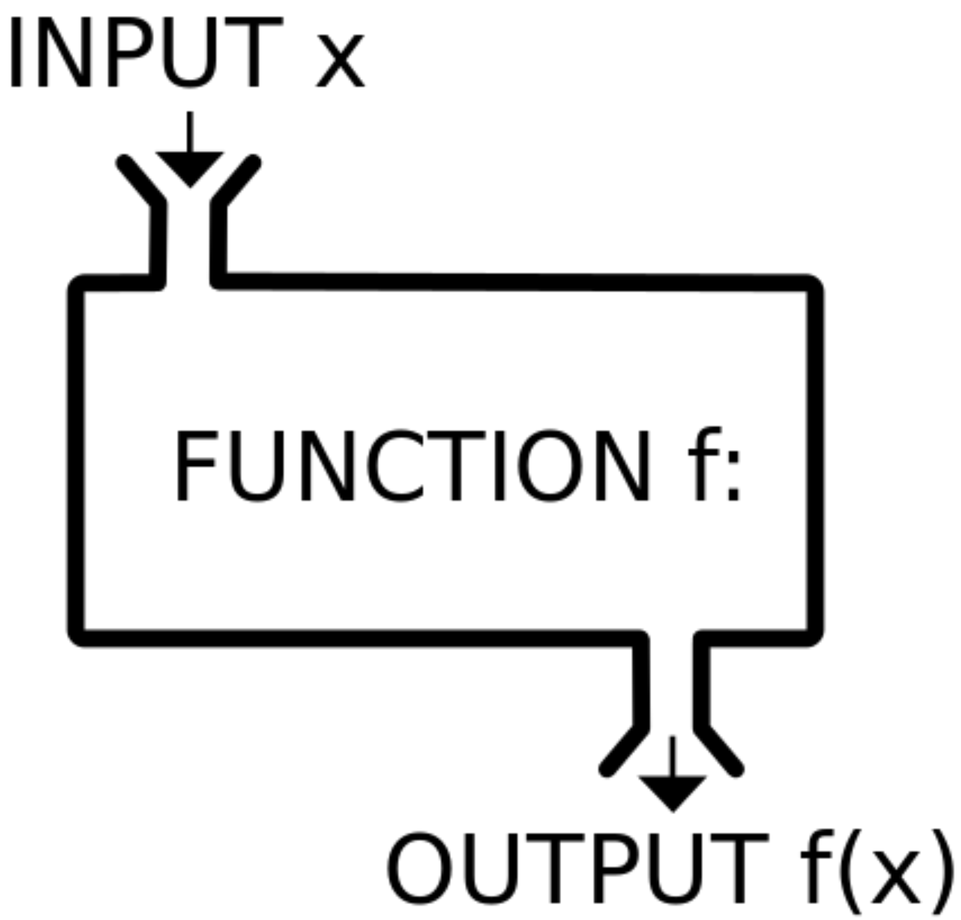

"보내는 사람"은
좌측 상단에 표기
보내는 사람
서울특별시 종로구 종로 6
(서린동)
우정사업본부 새주소우편팀
이몽룡
1 1 0 - 1 1 0
"받는 사람"은
우측 하단에 표기
우 표
받는 사람
전라남도 나주시 산포면 정보화길 1
우정사업정보센터 인터넷우체국
성춘향 귀하
5 2 0 - 7 0 0
우편번호는
반드시 표기하세요


오픈 소스 기술
API 통신
=
쉬워진
프로그래밍

+ α :
만들어진 코드를 활용하기
== 오픈소스 활용하기

함수(function)를 써보자

함수
Function



Excel 함수(function)

1) sum()
2) average()
3) count()

Python 함수

1) 내장함수
2) 외장함수

Python 내장함수

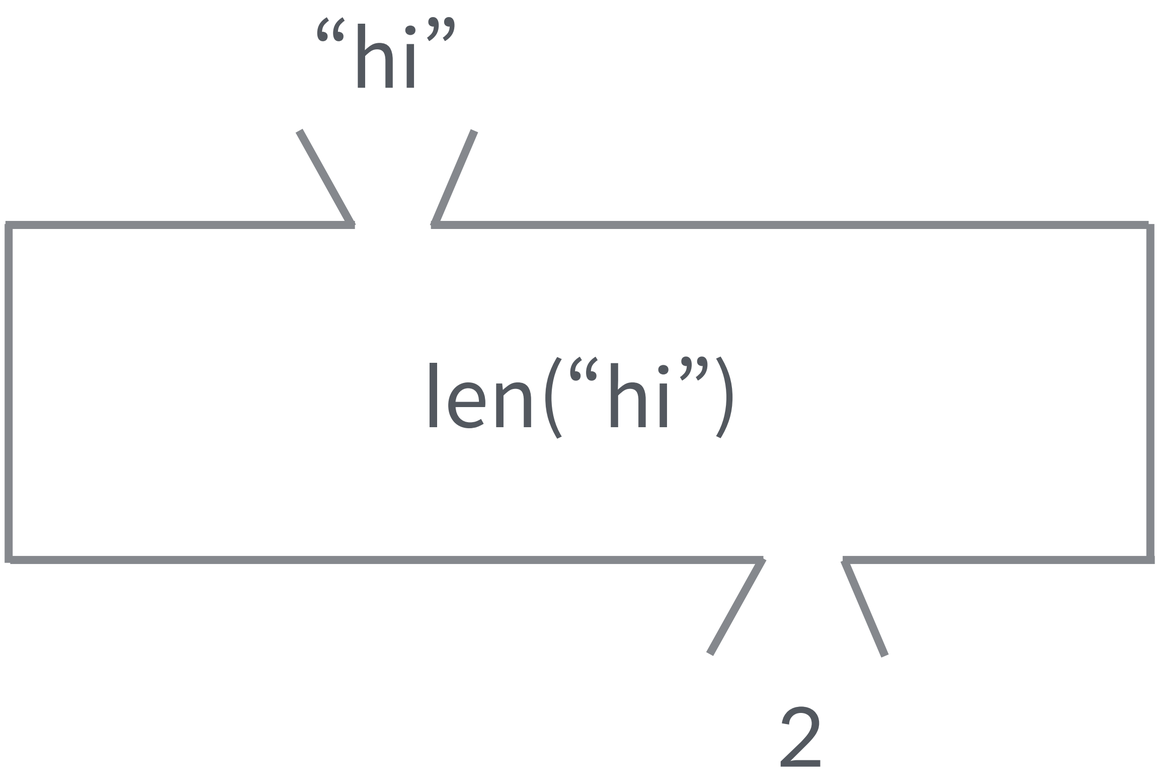
1) print('hi')
2) abs(-3) => 3
3) len('hi') => 2

2. 내장 함수

파이썬 인터프리터에는 항상 사용할 수 있는 많은 함수와 형이 내장되어 있습니다. 여기에서 알파벳 순으로 나열합니다.

		내장 함수		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

내장함수 예시
len()



Python 외장함수
- random

1) 오늘 점심 추천
2) 로또 번호 추첨

외장함수 활용 2 step

1. 함수가 포함된 코드 불러온다.(import)
2. 함수를 사용한다.

import random

1) random.choice
2) random.sample

random.choice(리스트)

리스트에서 임의적으로(randomly) 하나의 요소를 선택(choice)

ex) random.choice(menu)

외장함수 예시
random.choice()

menu = [“라면”, “김밥”, “떡볶이”, “순대”]

random.choice(menu)

김밥

외장함수 예시
random.choice()

1. 함수가 포함된 코드 불러온다.
2. 점심메뉴 리스트를 만든다.
3. 함수를 통해 결과 값을 저장한다.
4. 출력해본다.

외장함수 예시
random.choice()

1. 함수가 포함된 코드 불러온다.

   import random

2. 점심메뉴 리스트를 만든다.

   menu = [“라면”, “김밥”, “떡볶이”]

3. 함수를 통해 결과 값을 저장한다.

   choice = random.choice(menu)

4. 출력해본다.

   print(choice)

실습 5

- 외장함수(random) 활용하기
  - 점심메뉴 코드 업그레이드

```python
import random

# 메뉴들이 저장된 리스트를 만드세요.
menu = ["라면", "김밥", "떡볶이", "순대"]
choice = random.choice(menu)
print(choice)
```

+ α :
배달음식책자

메뉴 추천만 하지 말고, 시켜먹게 전화번호도 줘

+ α :
배달음식책자

- 외장함수(random) + 딕셔너리
  - 식당의 전화번호까지 출력

random.sample(리스트, 개수)

리스트에서 특정 수의 요소를
임의적으로(randomly) 비복원추출(sample)

ex) random.sample(numbers, 6)

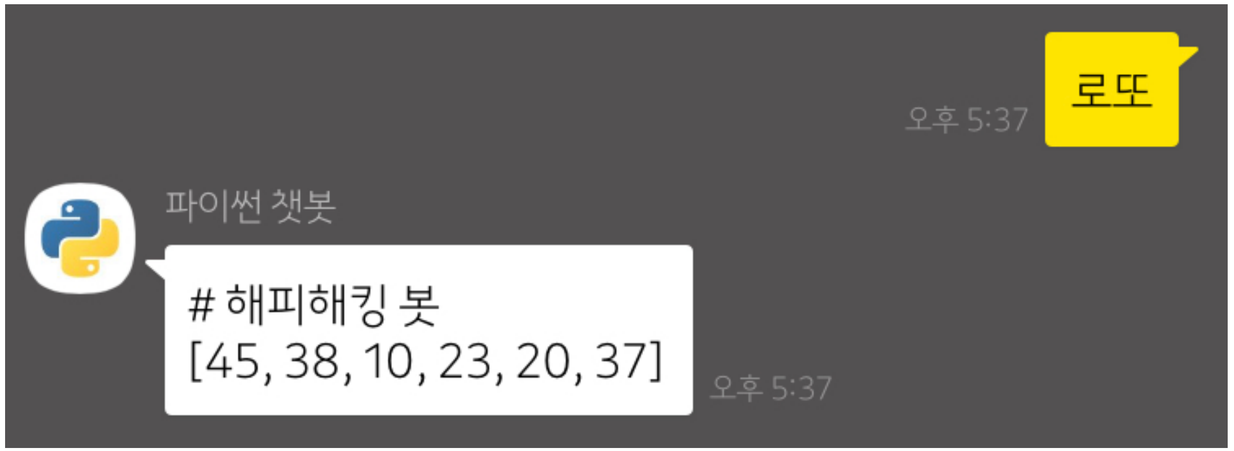
외장함수 예시
random.sample()

numbers = [1, 2, 3, 4, ⋯ , 44, 45]

random.sample(numbers, 6)

[13, 22, 6, 37, 1, 45]

실습 6



1) 외장함수 활용 - 로또 번호 추천

- 1~45의 수에서 뽑기!

<작성방법>

(1) random 친구를 불러 온다.

(2) 1부터 45 까지의 숫자를 저장한다.

(2) 외장함수 random.sample을 통해 6개의 숫자를 무작위로 뽑아 새로운 박스에 저장한다.

(3) 새로운 박스를 출력한다.

Summary

python 문법

1. 저장
2. 조건
3. 반복

python 문법 : 저장

1. 저장
2. 조건
3. 반복

저장	=
박스 내용	“글자”, 숫자, True/False
박스 종류	변수, [리스트], {딕셔너리}

python 문법 : 조건

1. 저장
2. 조건
3. 반복

```
if 비교문:
  실행문
elif 비교문:
  실행문
else:
```

python 문법 : 반복

1. 저장
2. 조건
3. 반복

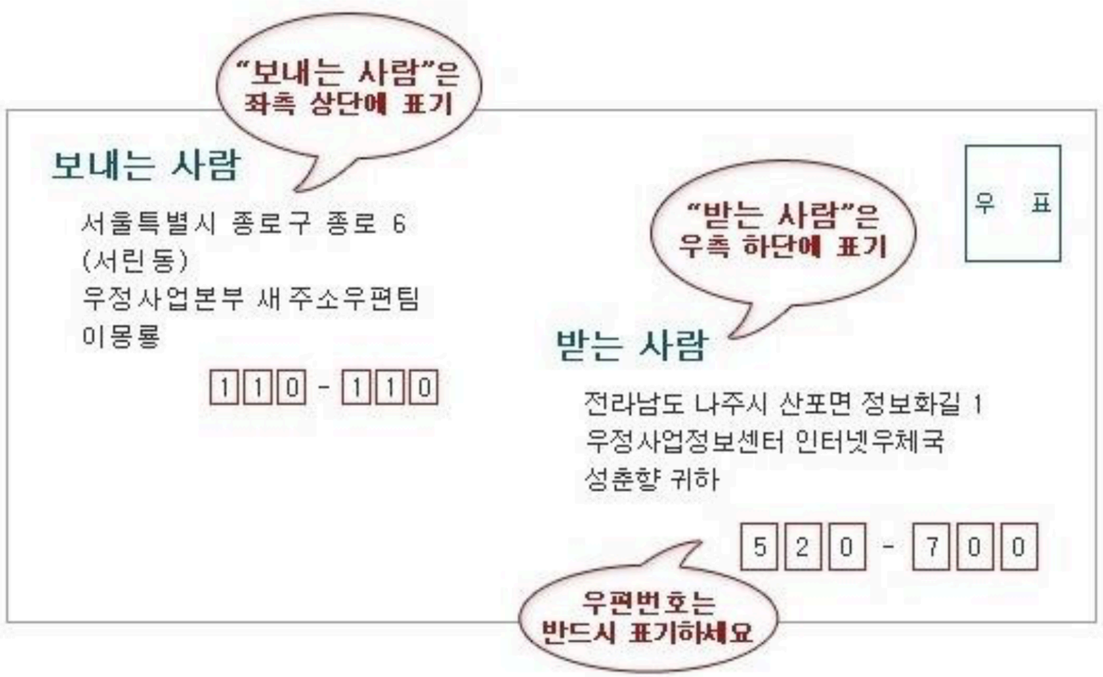
while True:	for i in dust:
조건이 False일 때까지 반복 종료조건이 필요함.	박스에 담긴 만큼 반복 종료조건 필요없음.

오픈소스

제작자의 권리를 지키면서 누구나 열람가능한
공개된 (소스)코드

API

서비스들간의 대화 방식



Session 4
Python 심화

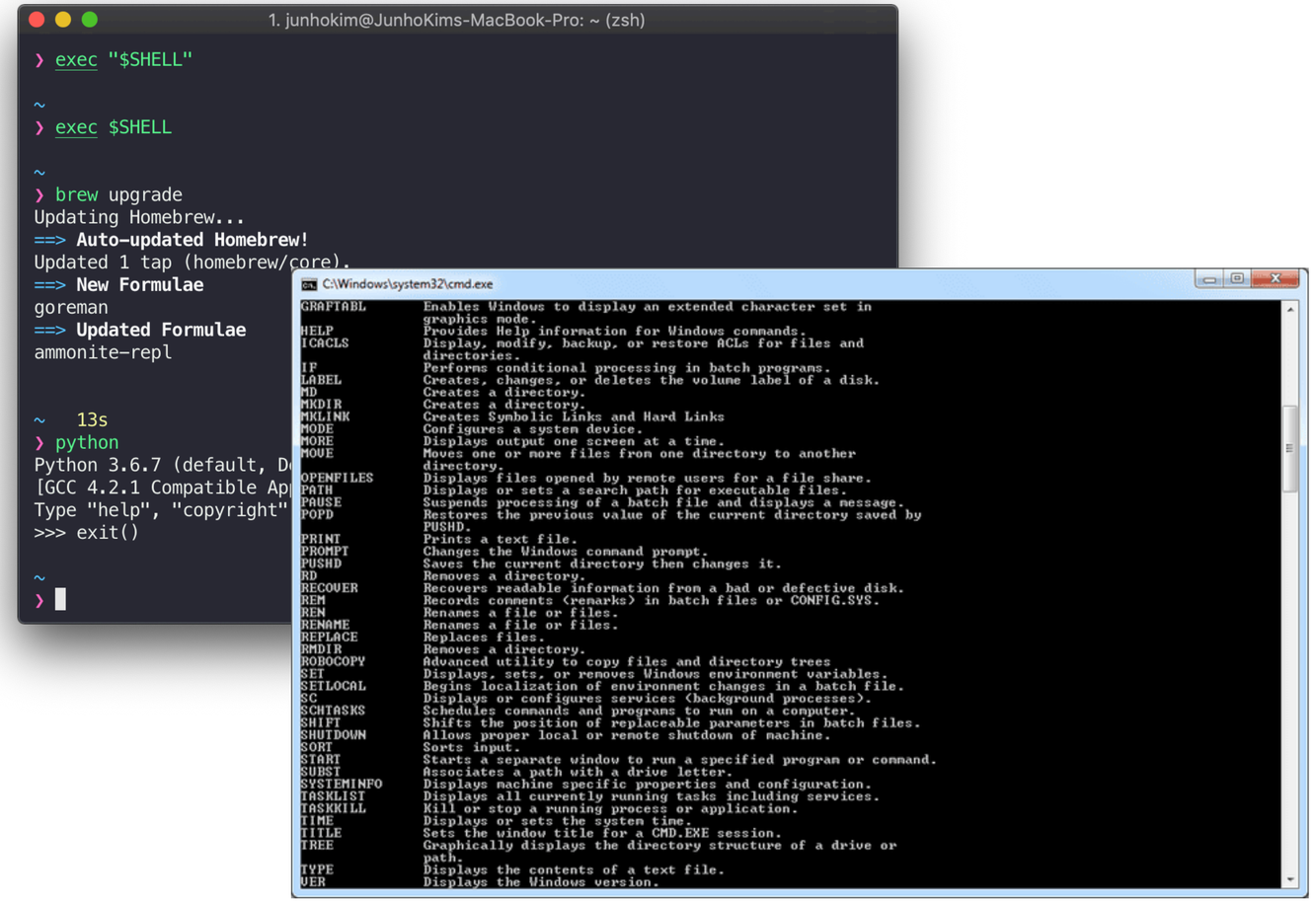
들어가기 전에:

CLI

이제는 개발자스럽게

까만화면
너무
멋있어..

아…
너무
멋지다.


1. junhokim@JunhoKims-MacBook-Pro: ~ (zsh)
❯ exec "$SHELL"
~
❯ exec $SHELL
~
❯ brew upgrade
Updating Homebrew...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
goreman
==> Updated Formulae
ammonite-repl
~ 13s
❯ python
Python 3.6.7 (default, D
[GCC 4.2.1 Compatible Ap
Type "help", "copyright"
>>> exit()
~
❯
C:\Windows\system32\cmd.exe
GRAFTABL Enables Windows to display an extended character set in graphics mode.
HELP Provides Help information for Windows commands.
ICACLS Display, modify, backup, or restore ACLs for files and directories.
IF Performs conditional processing in batch programs.
LABEL Creates, changes, or deletes the volume label of a disk.
MD Creates a directory.
MKDIR Creates a directory.
MKLINK Creates Symbolic Links and Hard Links
MODE Configures a system device.
MORE Displays output one screen at a time.
MOVE Moves one or more files from one directory to another directory.
OPENFILES Displays files opened by remote users for a file share.
PATH Displays or sets a search path for executable files.
PAUSE Suspends processing of a batch file and displays a message.
POPD Restores the previous value of the current directory saved by PUSHD.
PRINT Prints a text file.
PROMPT Changes the Windows command prompt.
PUSHD Saves the current directory then changes it.
RD Removes a directory.
RECOVER Recovers readable information from a bad or defective disk.
REM Records comments (remarks) in batch files or CONFIG.SYS.
REN Renames a file or files.
RENAME Renames a file or files.
REPLACE Replaces files.
RMDIR Removes a directory.
ROBOCOPY Advanced utility to copy files and directory trees
SET Displays, sets, or removes Windows environment variables.
SETLOCAL Begins localization of environment changes in a batch file.
SC Displays or configures services (background processes).
SCHTASKS Schedules commands and programs to run on a computer.
SHIFT Shifts the position of replaceable parameters in batch files.
SHUTDOWN Allows proper local or remote shutdown of machine.
SORT Sorts input.
START Starts a separate window to run a specified program or command.
SUBST Associates a path with a drive letter.
SYSTEMINFO Displays machine specific properties and configuration.
TASKLIST Displays all currently running tasks including services.
TASKKILL Kill or stop a running process or application.
TIME Displays or sets the system time.
TITLE Sets the window title for a CMD.EXE session.
TREE Graphically displays the directory structure of a drive or path.
TYPE Displays the contents of a text file.
VER Displays the Windows version.

GUI

Graphic User Interface

그래픽 인터페이스

CLI

Command line Interface

명령 줄 인터페이스

CLI 종류

1. 유닉스 shell(sh, zsh, bash 등)
2. CP/M
3. DOS 의 command.com
4. etc...

CLI 종류

1. **유닉스 shell(sh, zsh, bash 등)**
2. CP/M
3. DOS 의 command.com
4. etc...

bash 명령어 기초

ls	현재 디렉토리의 내용들을 나열
cd	현재 작업하는 디렉토리를 변경
mkdir	새로운 디렉토리 생성
echo	문자열 출력
rm	파일 지우기
exit	터미널 종료

[참고]유용한 bash 명령어

CLI 에서는 항상
자신이 어디에 있는지 주의하자
(pwd)

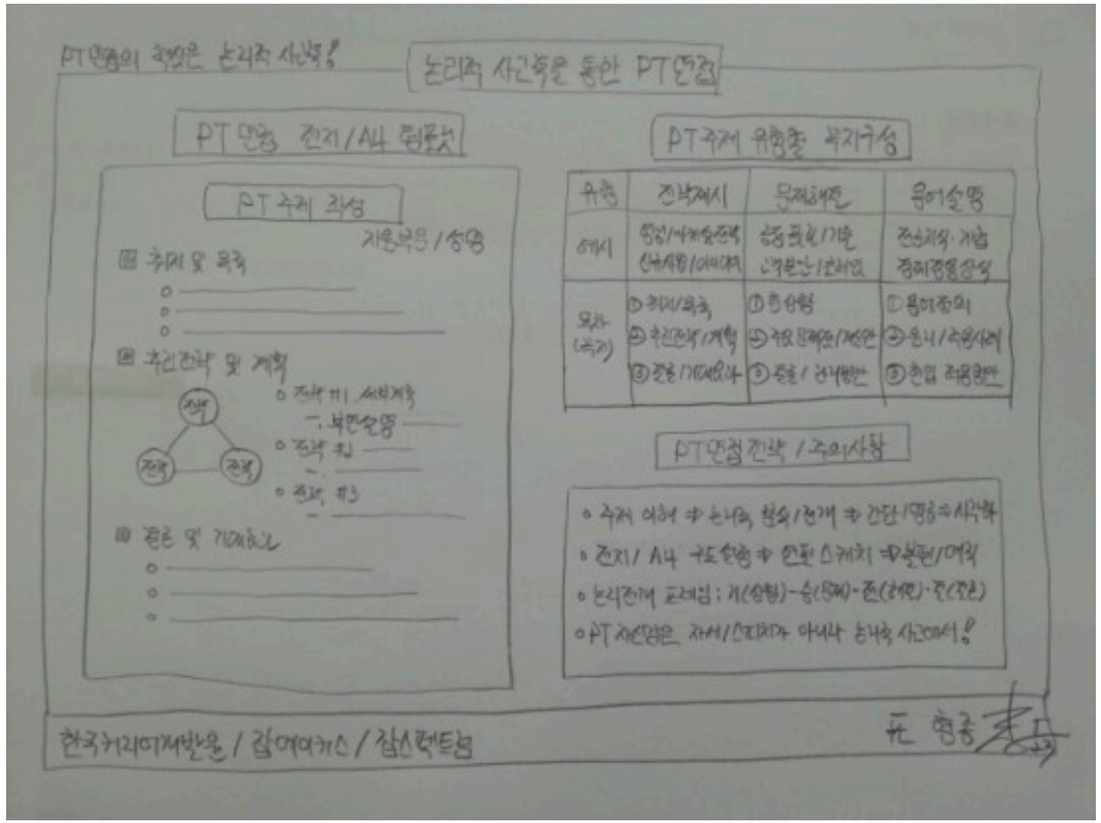
Session 4.
Python 심화

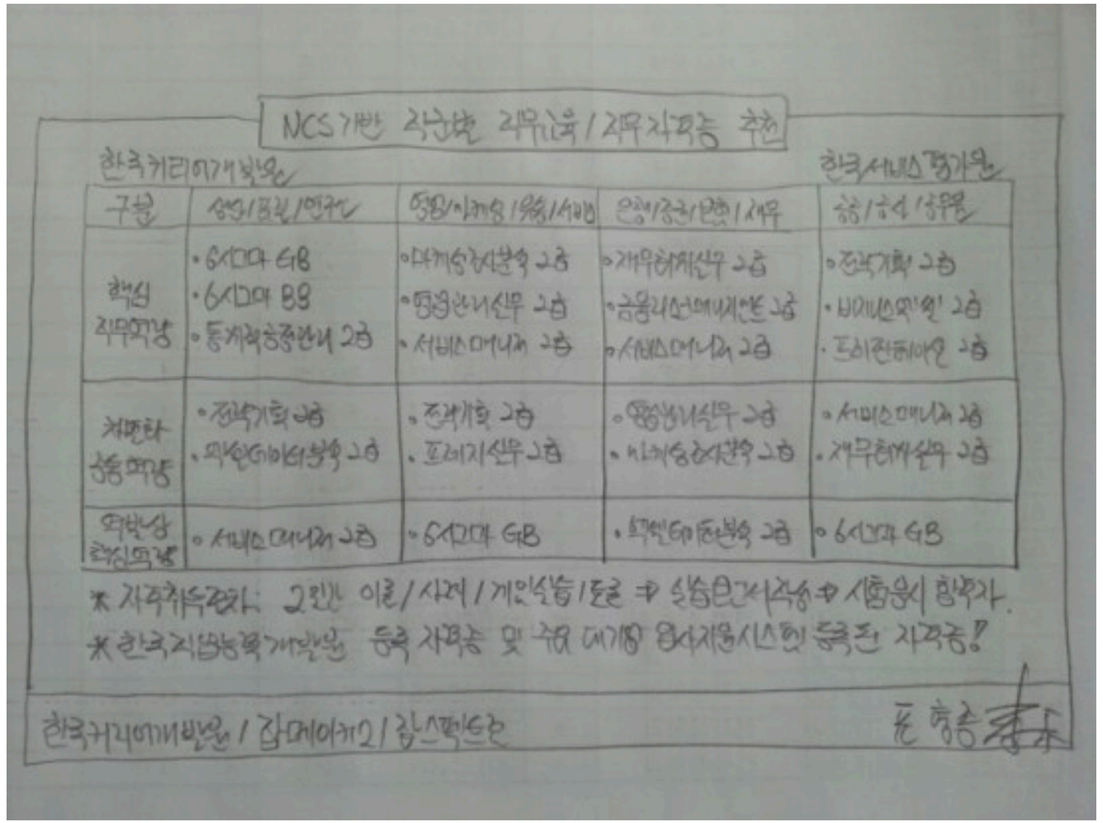
#컴퓨터조작 #웹스크래핑 #패키지 #라이브러리

Life Hacking

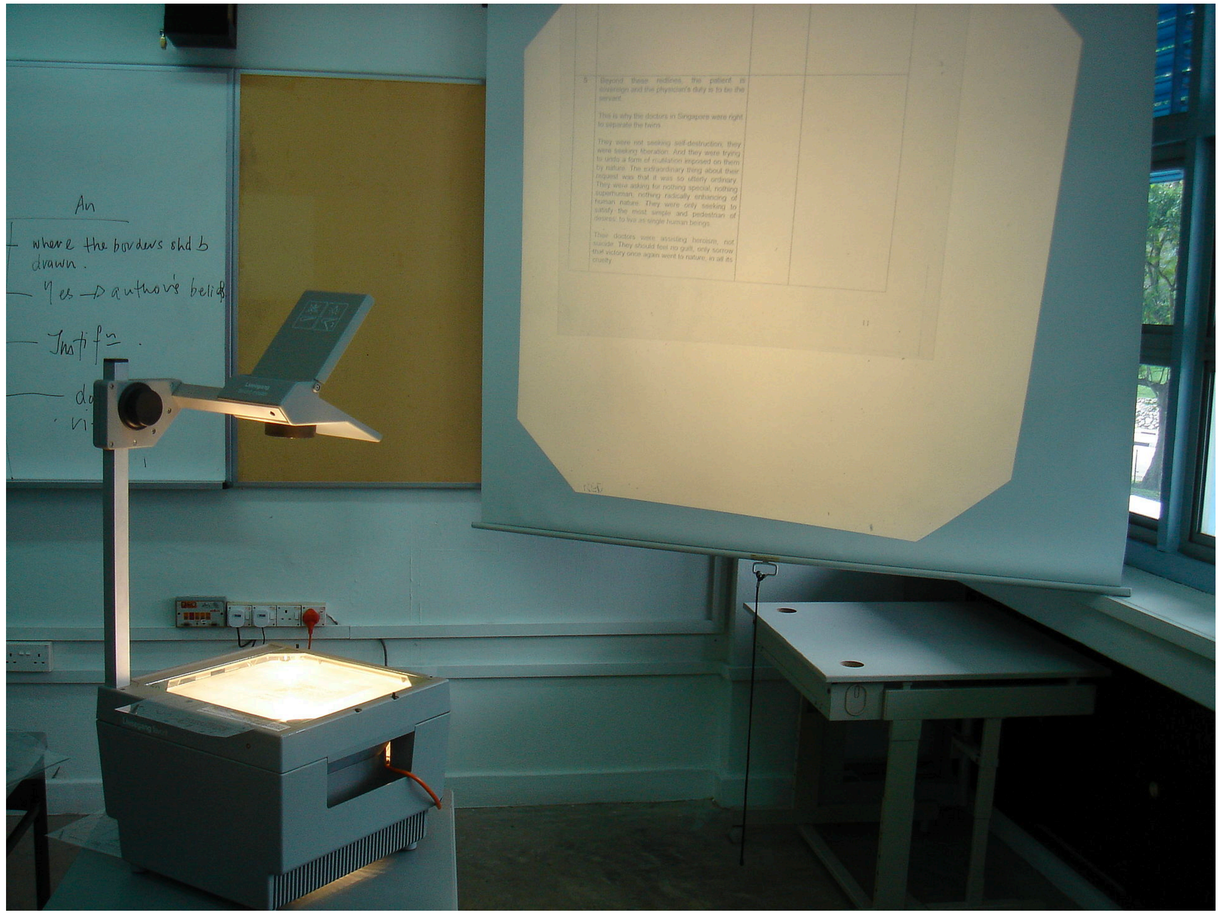
Work Hacking

Work Hacking



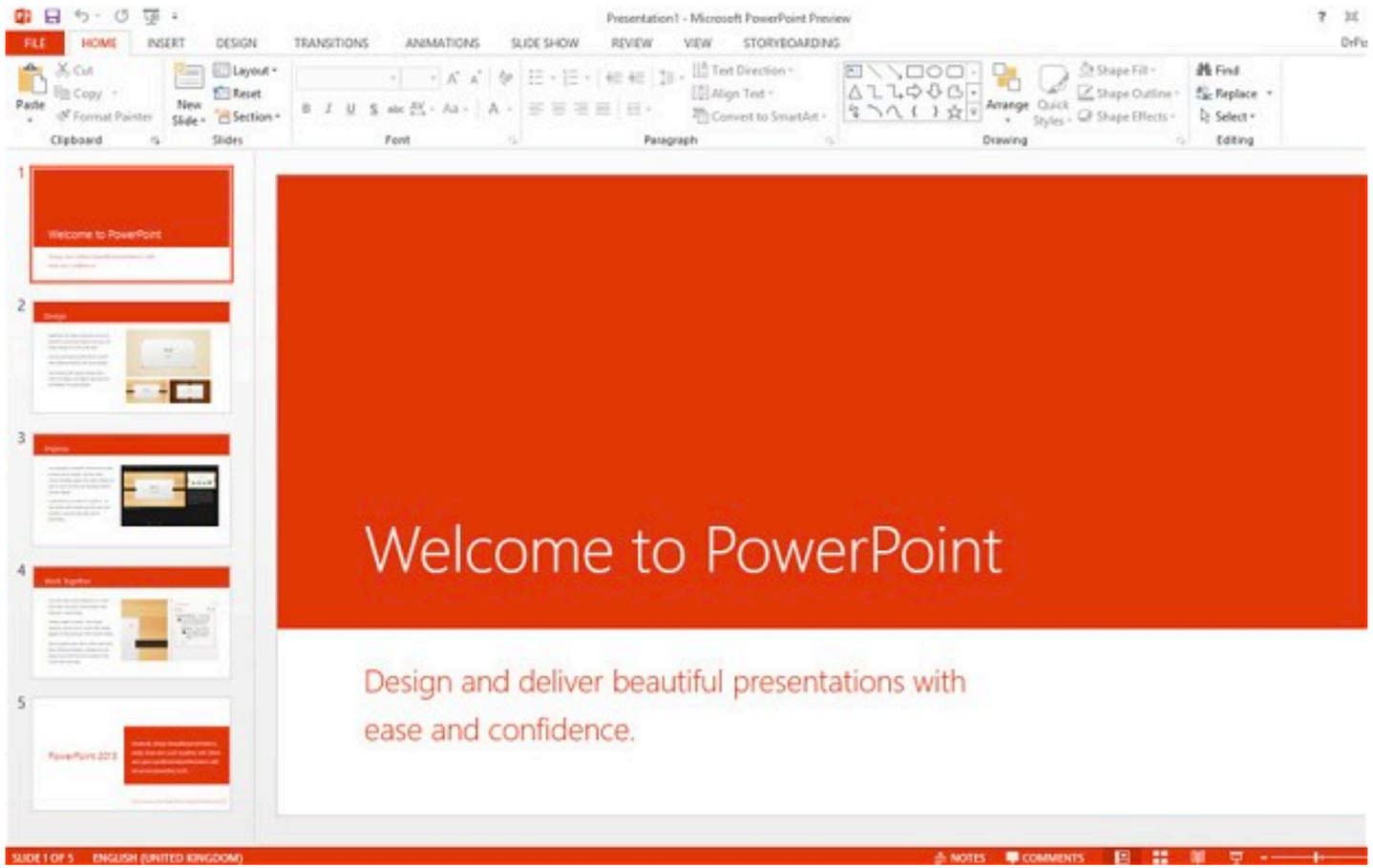


Work Hacking



hphk.

Work Hacking



Work Hacking

Work Hacking

파워포인트로 톱니바퀴 도형 그리기, PPT로 연출하는 스팀펑크

- 파워포인트로 톱니바퀴를 그려보자
  - 완성본 프리뷰
  - 톱니바퀴 그리기
  - 응용하기

Work Hacking

개발자가 되자?

피피티는 하자!

들어가기 전에..

#Git bash #Python #VScode

설치하기

1. git bash
2. python 3.7.3
3. vscode
4. typora

Python IDE

Integrated Development Environment
통합개발환경

Python IDLE

Python으로
내 컴퓨터를 조작하는 툴

첫번째 조작,
컴퓨터를 조작하기

웹 브라우저를 조작하기

import webbrowser

1) webbrowser.open(주소)
2) webbrowser.open_new(주소)
3) webbrowser.open_new_tab(주소)

webbrowser.open()

web browser를 열어줘

ex) webbrowser.open(‘http://google.com')

webbrowser.open_new()

web browser를 새 windows에서 열어줘

ex) webbrowser.open_new(‘http://google.com')

webbrowser.open_new_tab()

web browser를 새 tab 에서 열어줘

ex) webbrowser.open_new_tab(‘http://google.com')

좀 더 복잡한 Task 를 맡겨보자

list 를 사용해서 아이돌 검색 결과 여러 tab으로 띄우기

두번째 조작,
정보 스크랩하기

자주 확인하는 정보를 (자동)스크랩하기
ex) 코스피지수, 부동산가격 등

1. Naver 에 들어간다.
2. 정보를 검색한다.
3. 원하는 정보만 복사한다.
4. 내 컴퓨터에 저장한다.

import requests

1) requests.get(주소)
2) requests.get(주소).text
3) requests.get(주소).status_code

import requests

requests.get(‘주소’)

‘주소’에 요청(request) 보내서, 정보를 받아줘(get)

ex) requests.get('http://naver.com')

requests.get(‘주소’).text

‘주소’에 요청(request) 보내서, 정보 받아서(get),
글(text)만 뽑아줘

ex) requests.get('http://naver.com').text

requests.get(‘주소’).status_code

‘주소’에 요청(request) 보내서, 정보 받아서(get),
상태(status_code)만 뽑아줘

ex) requests.get('http://naver.com').status_code

정보 스크랩 1단계

1. 원하는 정보가 있는 주소로 요청을 보내, 응답을 저장한다.

2. 정보를 출력하여 확인한다.

정보 스크랩 1단계

1. 원하는 정보가 있는 주소로 요청을 보내, 응답을 저장한다.

```
import requests
response = requests.get('주소').text
```

2. 정보를 출력하여 확인한다.

```
print(response)
```

KOSPI 지수만 뽑을 수는 없을까?

import bs4

import bs4

1) bs4.BeautifulSoup(문서)
2) bs4.BeautifulSoup.select(경로)
3) bs4.BeautifulSoup.select_one(경로)

bs4.BeatifulSoup(문서)

받은 문서를 예쁘게(보기 좋게, 검색하기 좋게) 만들어줘

ex) text = bs4.BeautifulSoup(response)

.select(selector)

문서 안에 있는 특정 내용을 뽑아줘(select)

ex) text.select(response)

.select_one(selector)

문서 안에 있는 특정 내용을 하나만 뽑아줘(select_one)

ex) text.select_one(‘정보경로’)

정보 스크랩 1단계

1. 원하는 정보가 있는 주소로 요청을 보내, 응답을 저장한다.

2. 정보를 출력하여 확인한다.

정보 스크랩 1단계

1. 원하는 정보가 있는 주소로 요청을 보내, 응답을 저장한다.

```
import requests
response = requests.get('주소').text
```

2. 정보를 출력하여 확인한다.

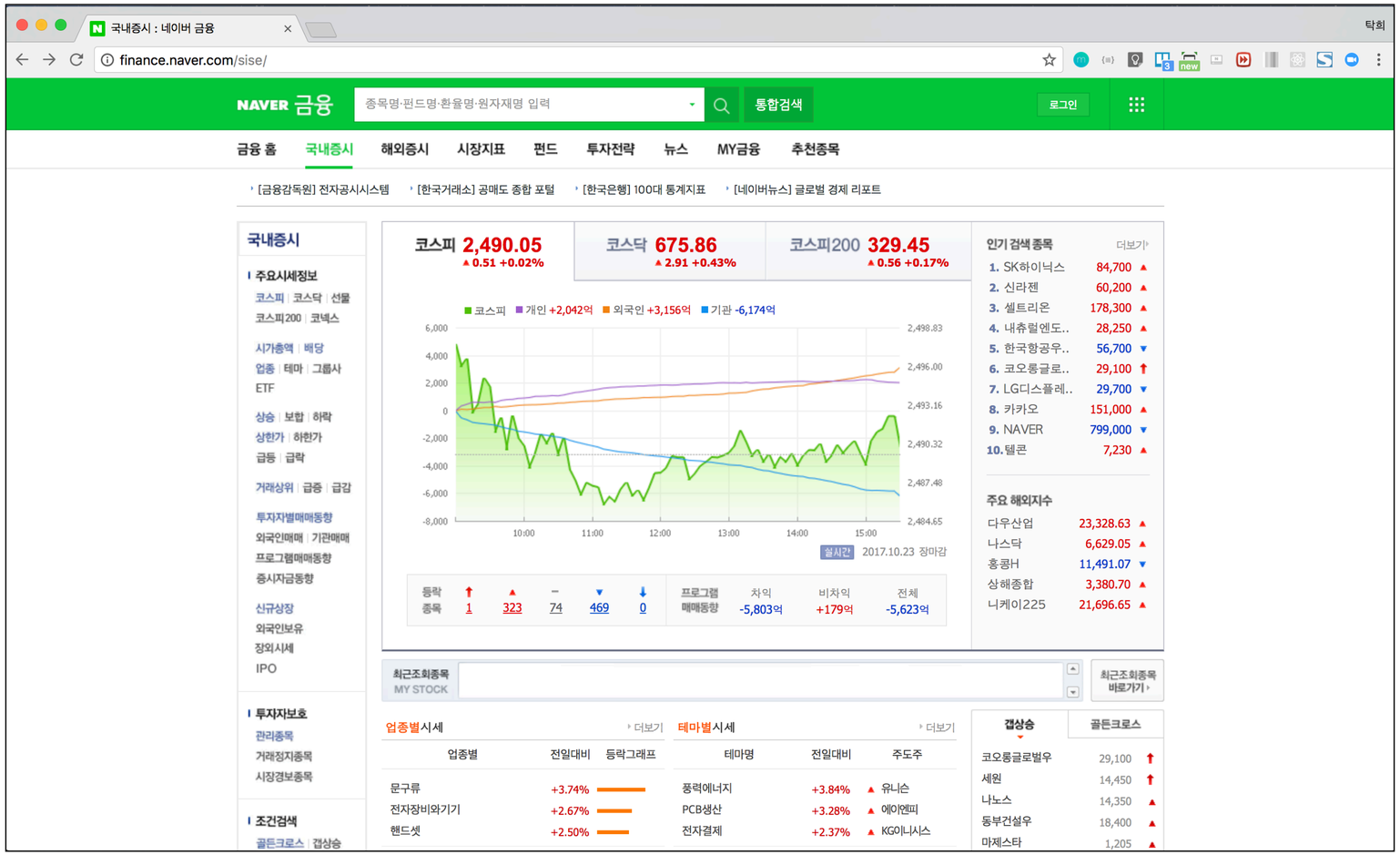
```
print(response)
```

정보 스크랩 2단계

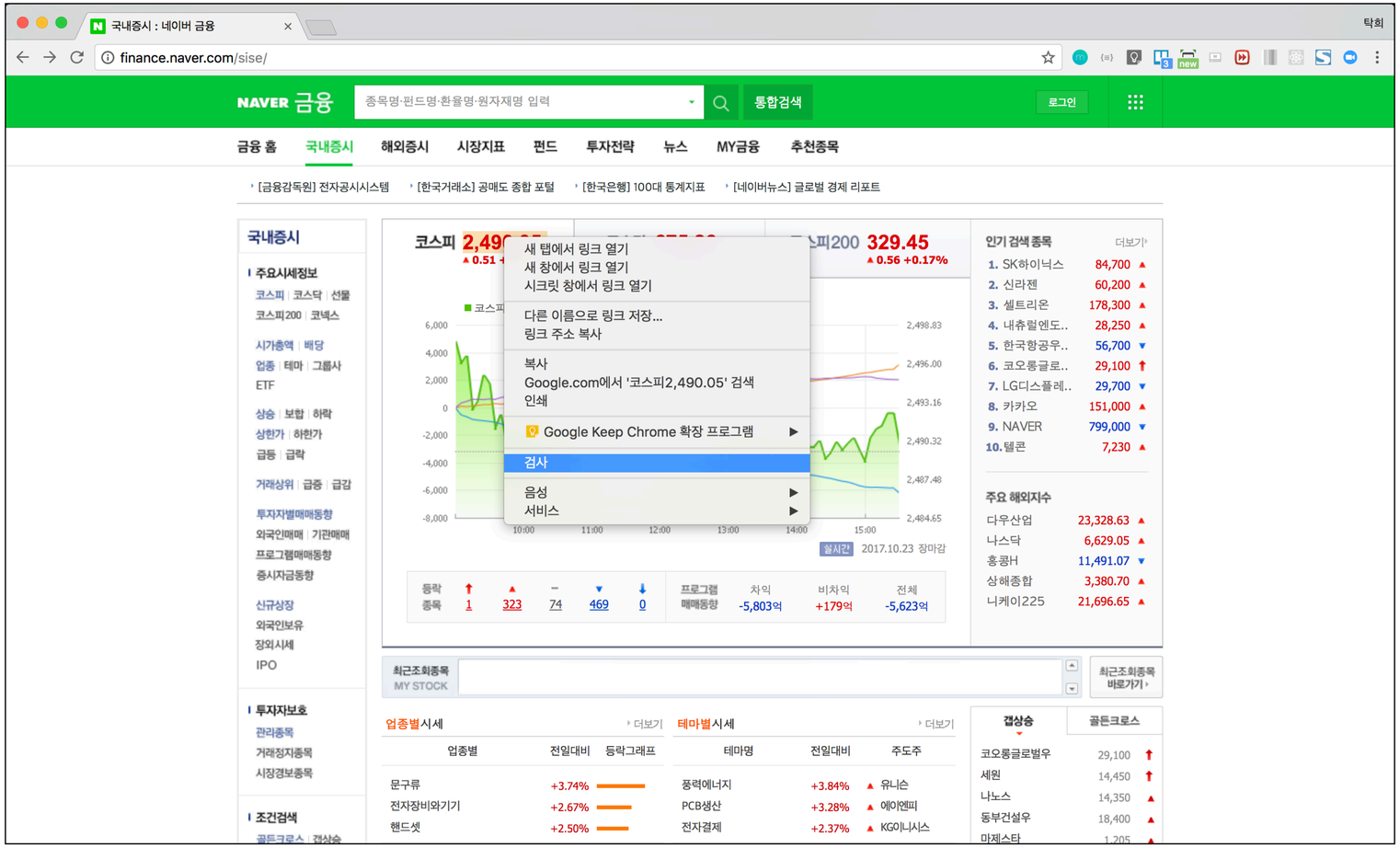
1. 정보를 조작하기 편하게 바꾸고,
2. 바꾼 정보 중 원하는 것만 뽑아서,
3. 출력한다.

정보 스크랩 2단계

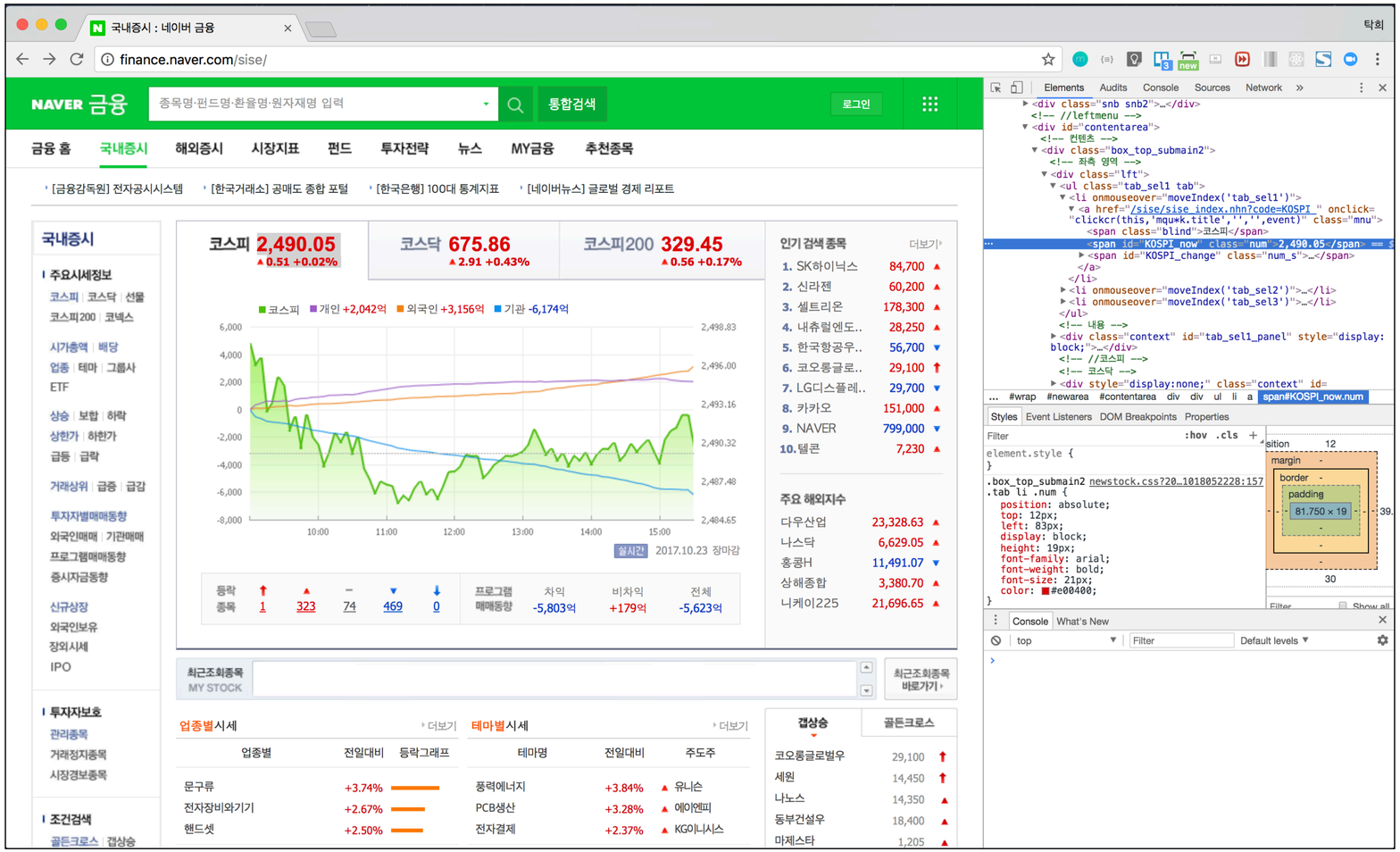
1. 정보를 조작하기 편하게 바꾸고,
   import bs4
   text = bs4.BeautifulSoup(response)
2. 바꾼 정보 중 원하는 것만 뽑아서,
   kospi = text.select_one('selector 경로')
3. 출력한다.
   print(kospi.text)



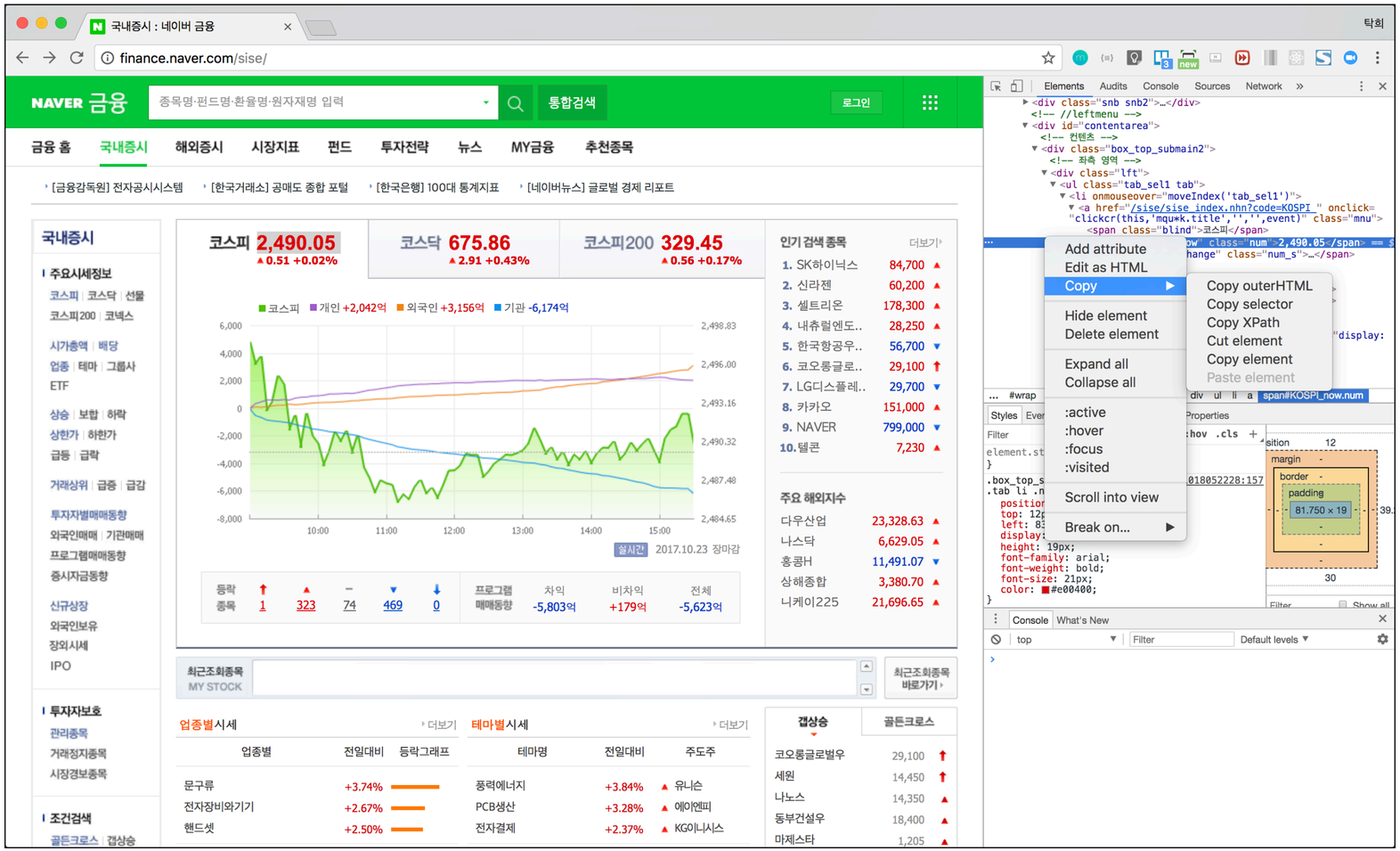
원하는 정보를 선택 후 우클릭, ‘검사’ 선택



개발자 도구(우측)에서 해당 정보 선택


국내증시 : 네이버 금융
finance.naver.com/sise/
NAVER 금융
종목명·펀드명·환율명·원자재명 입력
통합검색
로그인
금융 홈
국내증시
해외증시
시장지표
펀드
투자전략
뉴스
MY금융
추천종목
[금융감독원] 전자공시시스템
[한국거래소] 공매도 종합 포털
[한국은행] 100대 통계지표
[네이버뉴스] 글로벌 경제 리포트
국내증시
주요시세정보
코스피 2,490.05
0.51 +0.02%
코스닥 675.86
2.91 +0.43%
코스피200 329.45
0.56 +0.17%
인기 검색 종목
1. SK하이닉스 84,700
2. 신라젠 60,200
3. 셀트리온 178,300
4. 내츄럴엔도.. 28,250
5. 한국항공우.. 56,700
6. 코오롱글로.. 29,100
7. LG디스플레.. 29,700
8. 카카오 151,000
9. NAVER 799,000
10. 텔콘 7,230
주요 해외지수
다우산업 23,328.63
나스닥 6,629.05
홍콩H 11,491.07
상해종합 3,380.70
니케이225 21,696.65
업종별시세
테마별시세
캡상승
골든크로스
Elements
Audits
Console
Sources
Network
Styles
Event Listeners
DOM Breakpoints
Properties
Console
What's New

우클릭 후, 복사(copy)



CSS selector 복사(copy selector)

그럼 같이
한번
해보자!

+ α1 :
네이버 금융 KOSPI 지수 가져오기

+ α2 :
네이버 금융 원/달러 환율 가져오기

+ α3 :
네이버 실시간 검색어 스크래핑

API vs Crawling

프로그래밍 하라고 준 데이터 쓰기

vs

사람 보라고 준 데이터 긁기

API 는 충분히 친절하다

우리가 hacking 할 수 있다면 얼마든지
가져다 쓸 수 있다.(무료라고는 안했다)

그런데..

더 쉬운 방법은 었는걸까..?

까놓고 말해서 더 코드가 짧았으면 좋겠다.

Python Package

iexfinance 활용

주식정보 가져오기

https://iextrading.com/developer/

챗봇에 '주가' 검색

```
from iexfinance.stocks import Stock
tsla = Stock('TSLA')
tsla.get_price()
```

API vs Package

프로그래밍 하라고 준 데이터 쓰기

vs

제—발 쓰라고 주는 코드(함수) 덩어리

Summary

이 세상에는 수많은 데이터들 있고,
우리는 그 데이터를 가져다 Hacking 할 수 있다 ٩(ˆ◡ˆ)و

Session 5
MarkDown

hphk.

Markdown 맛보기

Markdown Guide

Happy Hacking

Typora 설치

1. 기본 문법

Headings

<h1> : #

<h2> : ##

…

<h6> : ######

#는 h1 태그

##는 h2 태그

######는 h6 태그

2. 기본 문법

Emphasis

<b> : ** ** / __ __
<i> : * * / _ _

*를 1개 쓰면 이탤릭체

**를 2개 쓰면 보드체

_도 가능합니다.

_도 가능합니다.

3. 기본 문법

Blockquotes

<pre> : >

> 는 pre 태그에여!

4. 기본 문법

Code block

1. ```작성 코드 종류
2. `짧은 코드`

```python
이라고 입력하면

```

이 만들어 지고 아래와 같이 코드를 작성할 수 있습니다.

```
print('안녕하세요')
```

짧은 코드는 `print('안녕하세요')`로도 표현 가능해요!

5. 기본 문법

img / url

1. 이미지
2. url

[이렇게 표현하면 됩니다](url 넣으시구요)

오늘 학습한 내용을 정리 해봅시다.