

| Chapter |

01



# 대화형 인공지능 개요

---

# 인공지능



- 인류 10대 난제 / 인류의 생존을 위협하는 기술

- 핵융합발전
- 암정복
- 줄기세포치료
- 뇌의비밀
- 수소혁명
- 우주개발
- 유전체혁명
- **인공지능**
- 노화와죽음
- 대도시



<http://www.yonhapnews.co.kr/medialabs/v2/post/229>

<http://news.joins.com/Issue/11054>

# AI 기술의 양면성



# 인류의 비서? 경쟁자?

- 완전한 인공지능의 개발이 인류의 종말을 초래할 수 있다(스티븐 호킹)
- 인공지능은 인류에게 핵무기보다 더 큰 위협이 될 수 있어 인공지능의 개발은 악마를 소환하는 행위이다(일론 머스크)



	10%	50%	90%
TOP100	2024	2050	2070
Combined	2022	2040	2075

# 문제는 사람

- 마이크로소프트 인공지능 채팅봇(테이)
- 공개 후 트위터 팔로어 10만명 이상 참여
- 공개 16시간 만에 운영 중단



백인 우월주의자, 여성과 무슬림 혐오주의자 등 일부 이용자가 욕설과 인종·성차별 발언 등을 가르쳐 테이를 '삐뚤어진 AI 로봇'으로 만듦

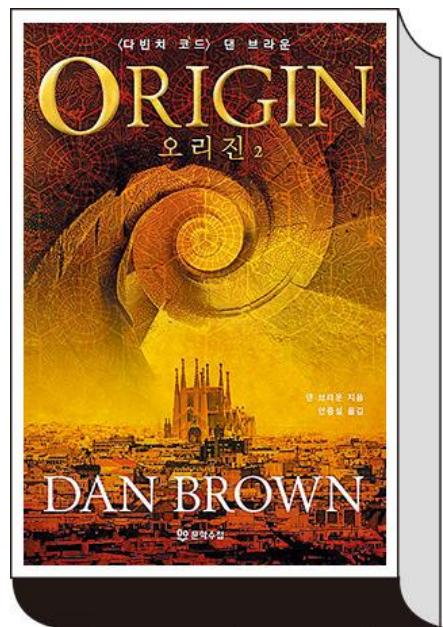
# 영화속의 대화형 인공지능



아이언맨

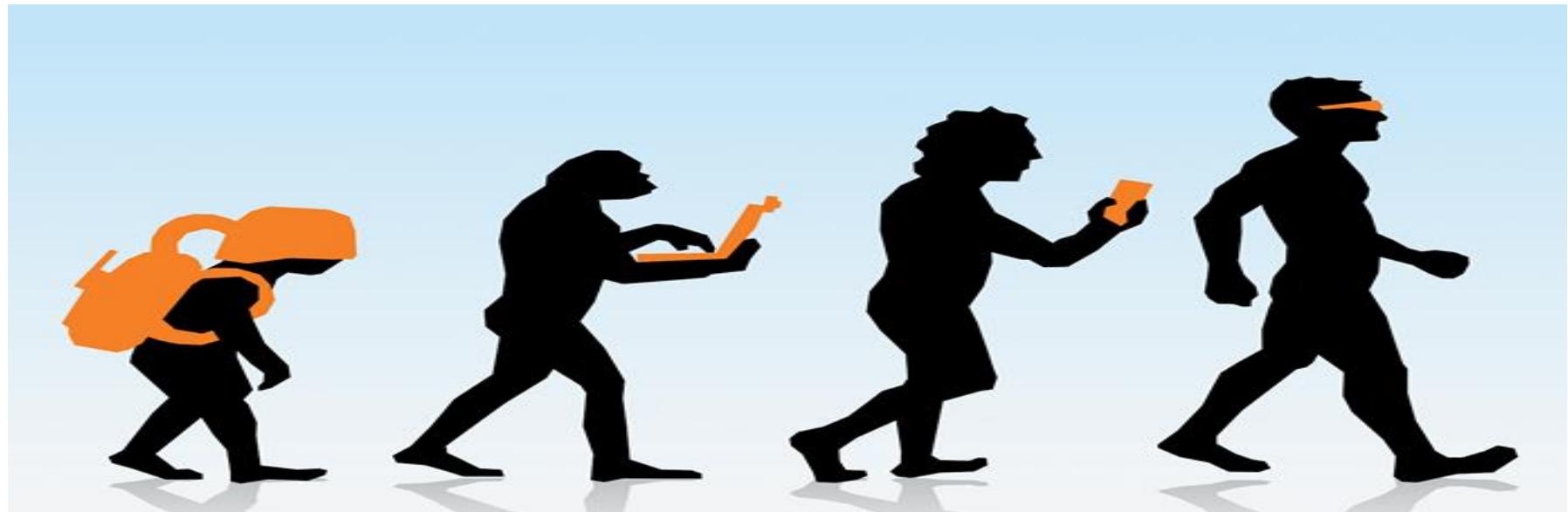


her



오리진

# 대화형 인공지능은 어디에 존재 하는가?



# 4차 산업 혁명의 핵심 인공지능

Big data / Industrial Internet / Sensor /  
Customization / AI / Self-driving cars / Smart  
City / Sharing economy / Data Science /  
Machine learning / Ubiquitous / Robotics /  
Open source / On demand / RFID /  
Wearable / Cloud computing / GPS / Cyber  
Physical System / Virtual Reality / Bio

“모든 것이 연결되고  
보다 지능적인 사회로 변화

*Velocity, Scope, System Impact*  
“Mastering the 4<sup>th</sup> Industrial Revolution”

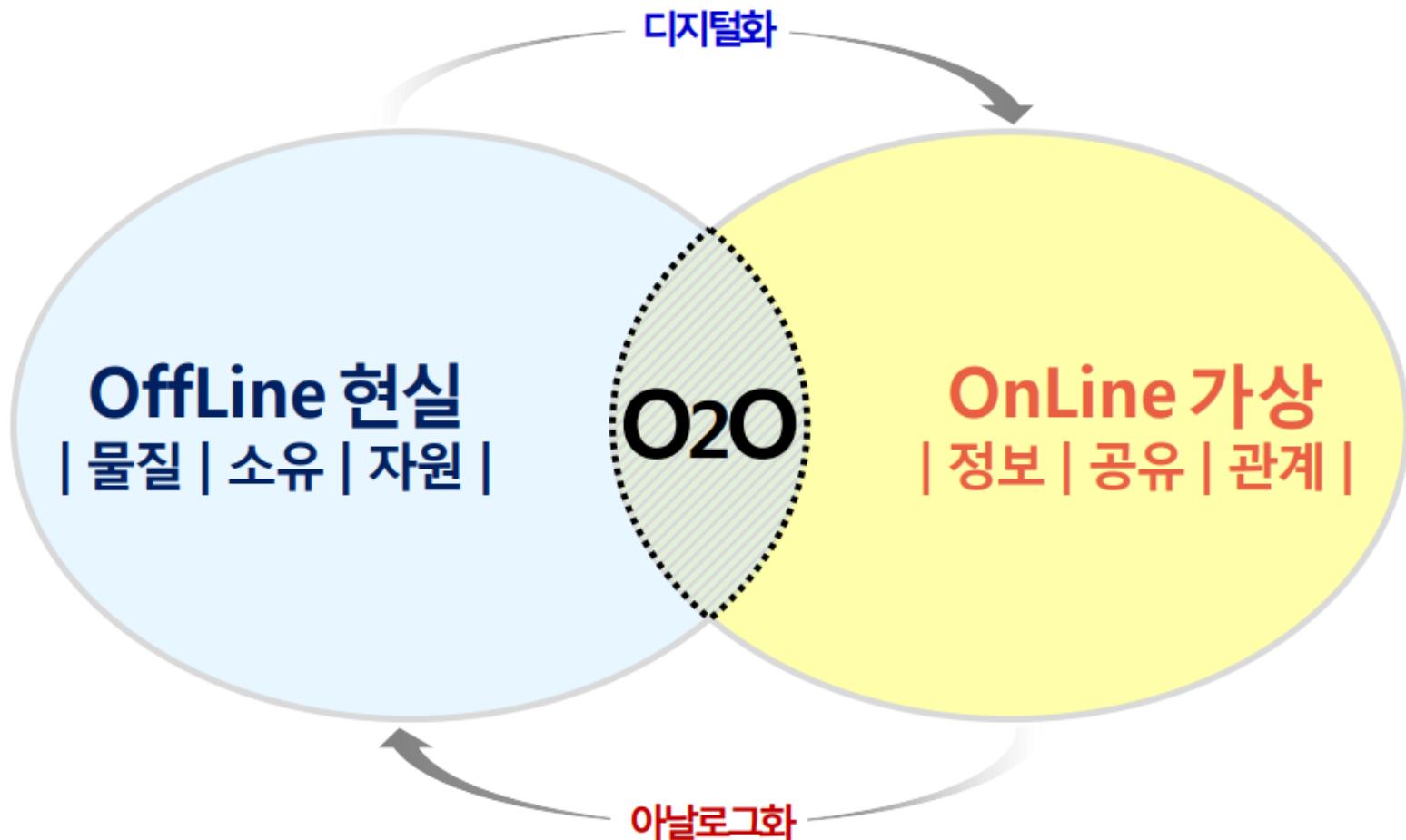


그러나, 아직  
4차산업혁명은  
미 정의





## 4차 혁명, 현실(1,2차)과 가상(3차)의 융합



# 아마존고(무인점포)



<https://www.youtube.com/watch?v=NrmMk1Myrxc&t=10s>  
<https://www.youtube.com/watch?v=7MIHgOf4P-s>

# 챗봇을 이용한 O2O

- 시간의 제약을 받지 않는 서비스제공을 통해 고객 만족, 서비스 질 향상, 비용 절감  
(쇼핑 사이트의 상품 검색/클릭 → 내일 12시까지 A 배달해줘)



AI 숙박 챗봇 '알프레도'



배달 민족 인공지능 챗봇 데이빗

[http://blog.daum.net/\\_blog/BlogTypeView.do?blogid=04N9i&articleno=9303422&categoryId=1&regdt=20170726113008](http://blog.daum.net/_blog/BlogTypeView.do?blogid=04N9i&articleno=9303422&categoryId=1&regdt=20170726113008)

# 인공지능 챗봇, 이런 것도?



*"Hi, I'm calling to book a  
women's haircut for a client."*

구글 듀플렉스

<https://www.youtube.com/watch?v=pci3IMxFk4M>

# 인공지능 독해력, 인간을 추월하다

- 스탠퍼드 대 독해력 시험
  - 500개 이상의 위키피디아 글을 토대로
  - 10만개 이상의 질문을 만들어 정확한 답을 제출
- 알리바바그룹 홀딩스가 개발한 인공지능(AI)이 인간보다 높은 성적
  - 알리바바 : 82.44    사람:82.304    MS:82.65
- 인간의 단어 및 문장 이해 방식을 흉내낸 컴퓨터의 자연어 처리 능력이 인간을 앞서기는 이번이 처음
- 객관적 질문에 기계가 아주 정확하게 대답할 수 있는 수준에 도달했음
- 고객 서비스, 박물관 안내, 환자 질의에 대한 온라인 응답 같은 광범위한 분야에 활용가능

<http://www.hani.co.kr/arti/science/technology/827949.html>

- 구글 딥마인드 문자-음성 변환 시스템 태코트론2
- 사람 수준으로 발성...문장부호·대문자 맞춰 문장 발음
- 평균 발성 점수(MOS) 비교
  - 태코트론2 : 4.53점으로( 2016년도 4.21)
  - 숙련된 사람의 녹음 목소리 : 4.58점
- 사람 목소리와의 간극이 사실상 없어짐

<https://google.github.io/tacotron/publications/tacotron2/index.html>

# 스타크래프트 대결

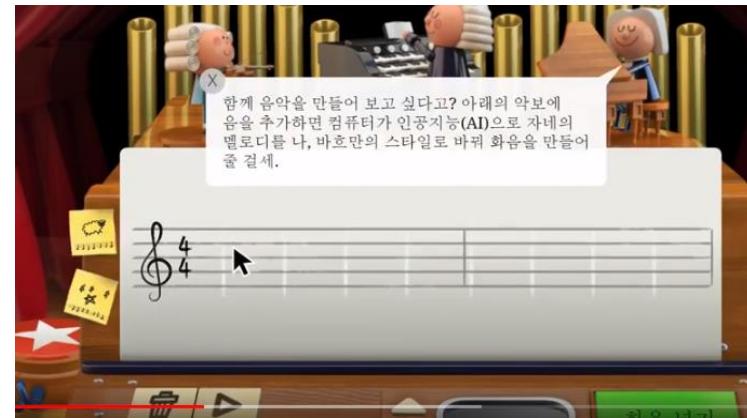
- 세종대(2017.10, SDS후원)
- 세종대 MJ봇, ZZK(호주, 세계1위), TSCMOO(노르웨이, 세계2위), Cherry Pi(Facebook)



<https://www.youtube.com/watch?v=uordmU2zOjE>

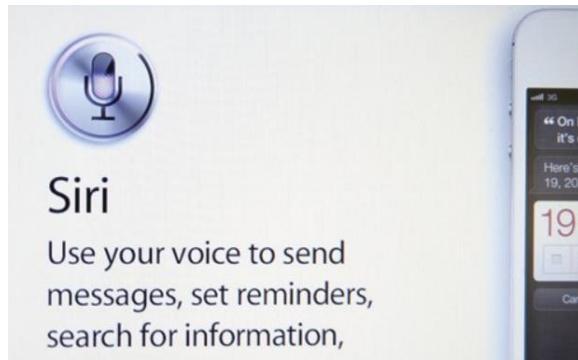
# 인간과 AI 대결 : 음악

- 키스 더 라디오
  - PD와 음악 선곡 대결
- AI 두들
  - 작곡가 바흐의 탄생일을 기념하여 구글 개발
  - 임의의 음을 입력하면 불완전한 부분을 채워 나감
  - 바흐의 음악들을 잘게 쪼개 파편화 후 무작위로 삭제/복원을 통해 바흐의 스타일에 맞는 음 추출



[https://www.youtube.com/watch?v=a75\\_7iDbjXA](https://www.youtube.com/watch?v=a75_7iDbjXA)

# 주변의 챗봇



시리



빅스비

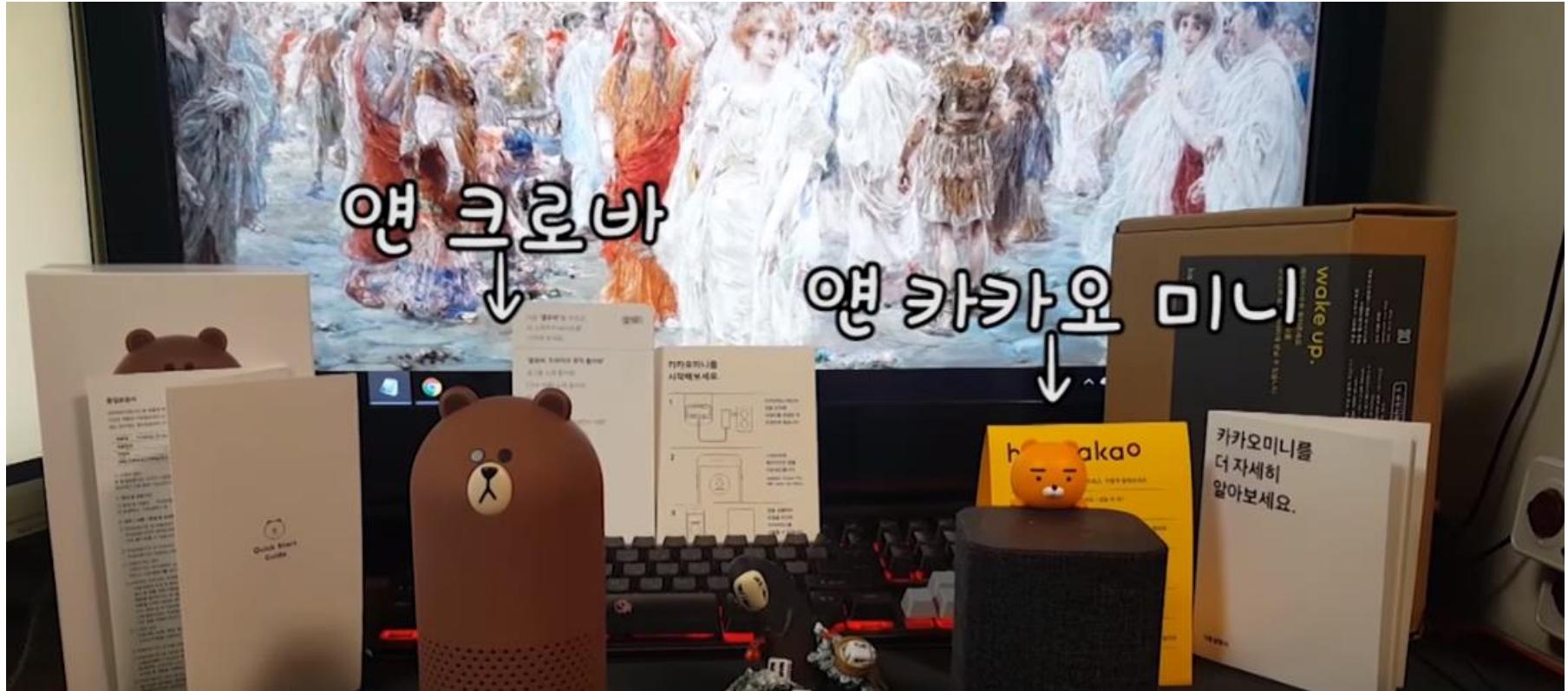


네이버



카카오

# 인공지능 스피커 대결



<https://www.youtube.com/watch?v=Y7IL2ZwkCJg>

<https://www.youtube.com/watch?v=p1qsKL4NBE4&t=1042s>

# 급성장하는 인공지능 스피커

<b>아마존</b>		<b>구글</b>	
제품명	• 에코	구글홈	
출시 시점	• 2014년 11월	• 2016년 10월	
특징	• 음성비서 알렉사 탑재	• 음성비서 구글 어시스턴트 탑재	
<b>애플</b>		<b>마이크로소프트</b>	
• 홈팟		• 인보크	
• 2017년 12월		• 올 하반기	
• 시리 탑재, 스피커 기능 강조		• 음성비서 코타나 탑재	
<b>네이버</b>		<b>SK텔레콤</b>	
• 웨이브		• 누구	
• 올 하반기		• 2016년 9월	
• AI 플랫폼 클로바 탑재		• 음악 및 쇼핑 서비스 연계	

자료:업계 종합

- 음성만으로는 정보 전달 한계 발생
- 시각적인 UI와 터치 인터페이스



아마존 Echo Show



SKT, 누구 네모

# 저커버그, AI 비서 '자비스' 영상 공개

- 2016년 개인 프로젝트 중 하나인 AI 음성비서의 첫 번째 버전
- AI 비서로 집안 가전제품들을 제어
- 저커버그는 “자비스에게 유머를 가르쳐 이젠 내 어린 딸 맥스도 자비스와 소통할 수 있다”
  - 자비스에게 “누구를 간지럽힐까?”라고 질문하면 자비스가 저커버그의 가족 구성원 중 아무나 지목하는 게임도 가능



▲ 저커버그(왼쪽)가 영화 아이언 맨(오른쪽)의 자비스에서 영감을 받아 직접 AI 음성비서를 만들었다. / 조선 DB

[https://www.youtube.com/watch?time\\_continue=83&v=ZGLPxEv\\_EWo](https://www.youtube.com/watch?time_continue=83&v=ZGLPxEv_EWo)

# 인공지능 스피커, 이런 것도?



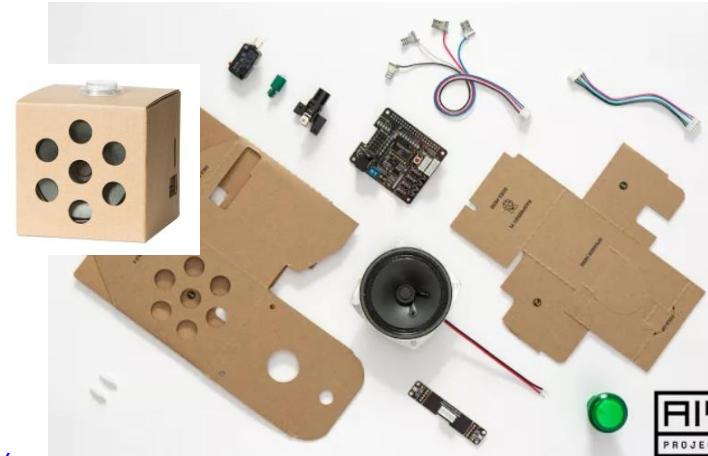
클로바 프렌즈 미니 도라에몽  
[https://www.youtube.com/watch?v=J\\_KTAwzYA2A](https://www.youtube.com/watch?v=J_KTAwzYA2A)



클로바 미니언즈  
<https://www.youtube.com/watch?v=YMI-CGOSHdY>



U+AI\_어벤져스' 만나보세요 .

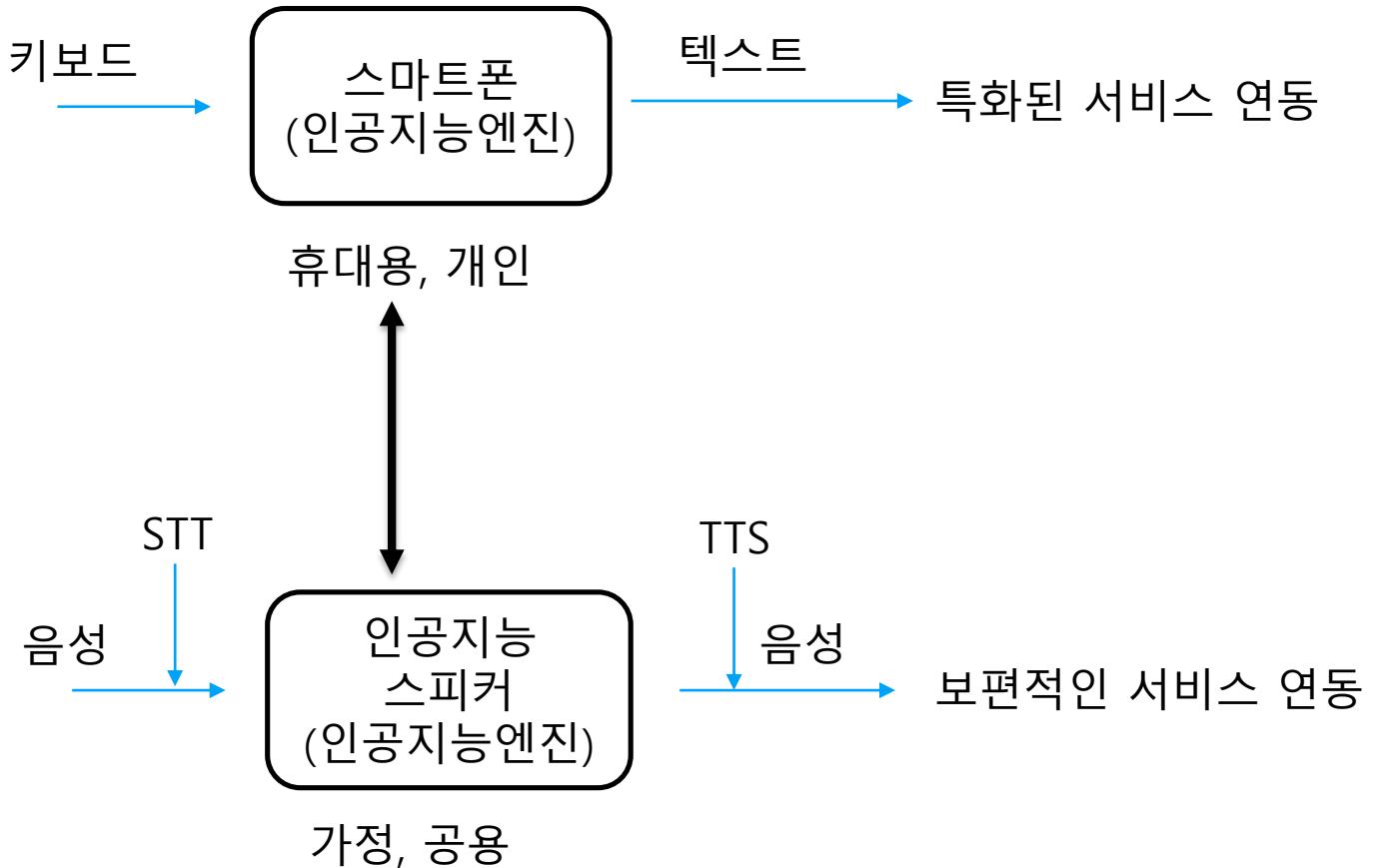


Google, AIY Project



KT, AI 메이커스

# 인공지능 스피커와 챗봇



# 챗봇 적용 분야 예측



Drift



Audience



salesforce



myclever

2018 State of Chatbots Report

# 챗봇 사용 이유

- 고객편의, 비용절감, 매출증대
- 비즈니스 환경의 변화
  - 모바일 메신저의 활성화
  - 브랜드 앱 설치 보다는 친숙한 메신저 플랫폼을 활용한 서비스 친밀도
- 자연어 처리 기술의 발전
- 관련 eco-system 발전
  - API플랫폼
  - Open Source/Big Data
- 24시간 비대면 서비스 강화

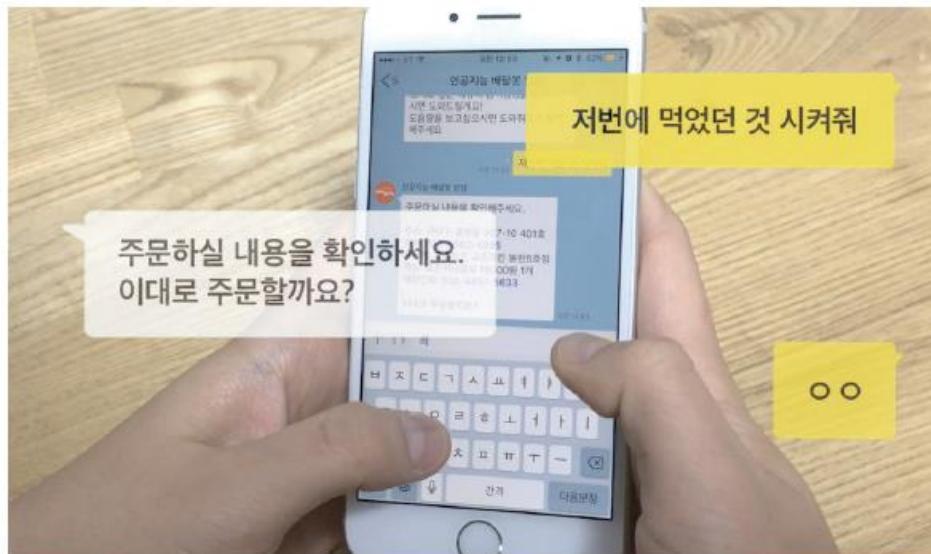


- 정량적 평가
  - 상담원수 줄이고 인건비 절감(\$3.7\$->\$0.3\$)
  - 콜센터 운영 비용 절감
- 정성적 평가
  - 양방향 소통 채널 확대
  - 24시간 365일 대응
- 마케팅 효과
  - 응답률 개선
  - 맞춤형 광고
  - 이벤트/광고 노출 시간 증가

# 챗봇이 웹사이트 보다 좋은 5가지 이유

- No need to learn a new user interface
- A truly personalized user experience
- Re-engage with users in a relevant way
- A better and faster experience
- Handoff to a Human Agent in a seamless way

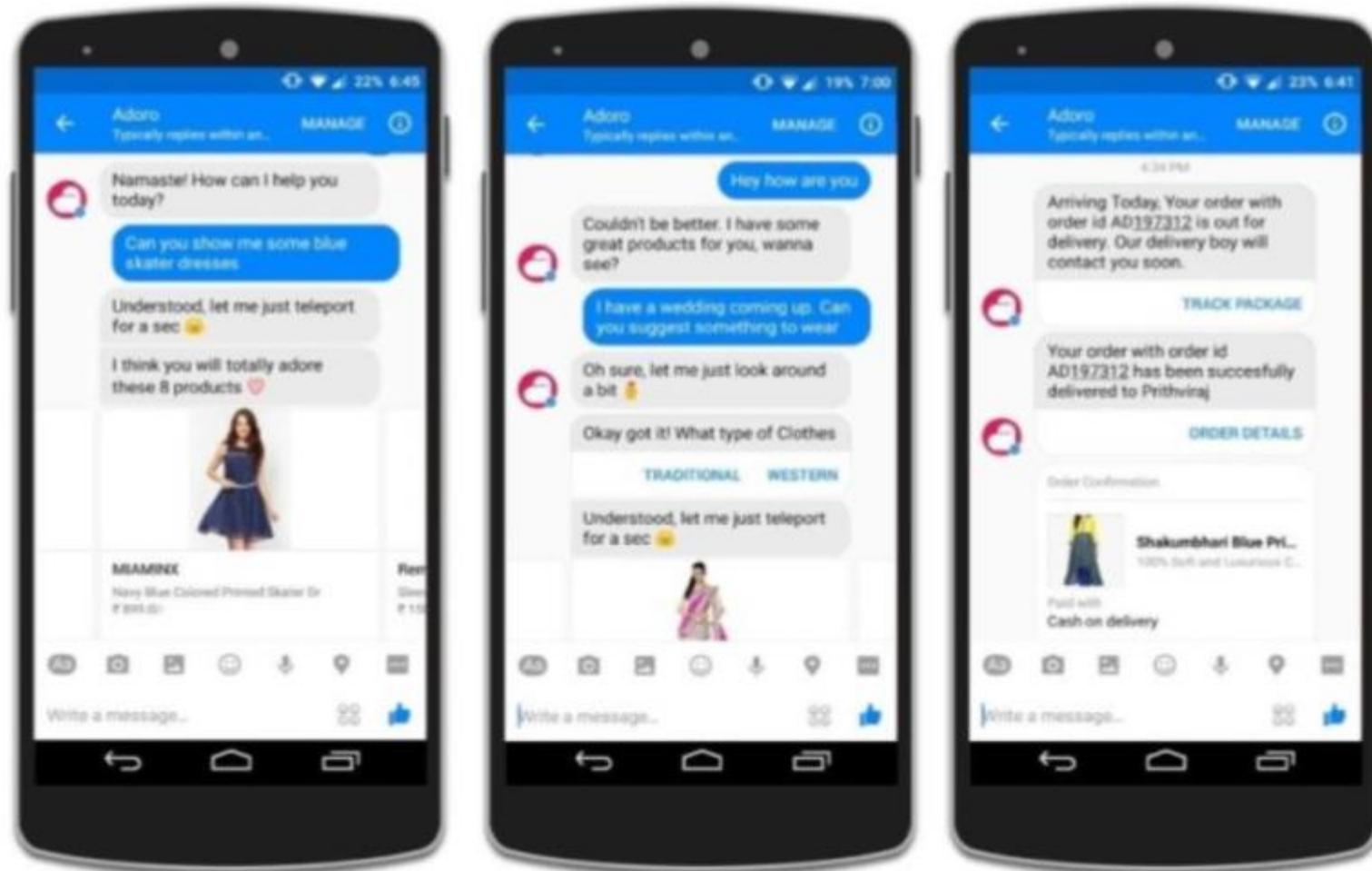
<https://medium.com/hijiffy/5-reasons-why-a-chatbot-is-better-than-a-website-75477cee7341>



[그림 9] 머니브레인은 인공지능 배달 첫봇 서비스인 '암암'을 출시했다. 암암은 카카오톡, 라인, 페이스북 메시지 앱에서 배달봇 암암을 친구추가하면 채팅을 통해 배달 음식을 주문할 수 있다. (출처: 머니브레인)

# 응용사례

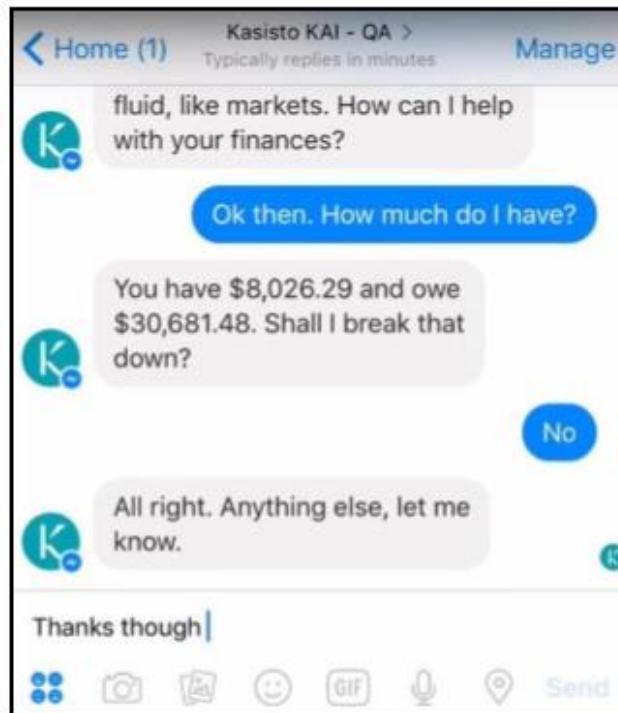
## ● Facebook 메신저 기반 챗봇 - 쇼핑



# 응용사례

- Kasisto KAI <http://kasisto.com>

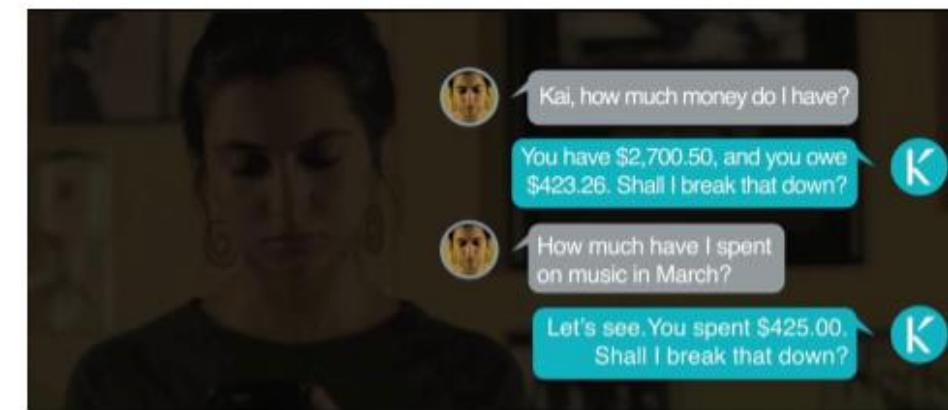
- ✓ 금융권의 챗봇을 전문적으로 개발 – 자연어처리에 강점



- ✓ 예

- <https://vimeo.com/169605404>
- <https://vimeo.com/207202830>

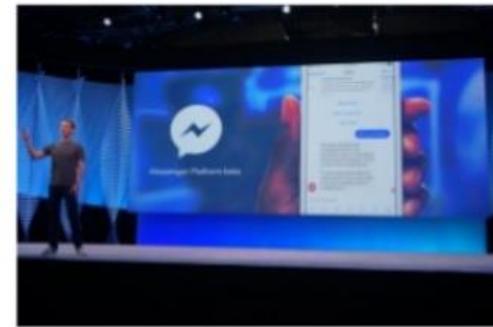
<https://vimeo.com/169605404>



# 챗봇 발전 방향



1996 : 문장변환 및 키워드 매칭



2016 : Face Book, 챗봇 API 공개



2016 : 카카오톡 친구 플러스 API 공개  
2019 : 챗봇 빌더 출시



앱기반 챗봇



인공지능 디스플레이

# 챗봇 세대 분류

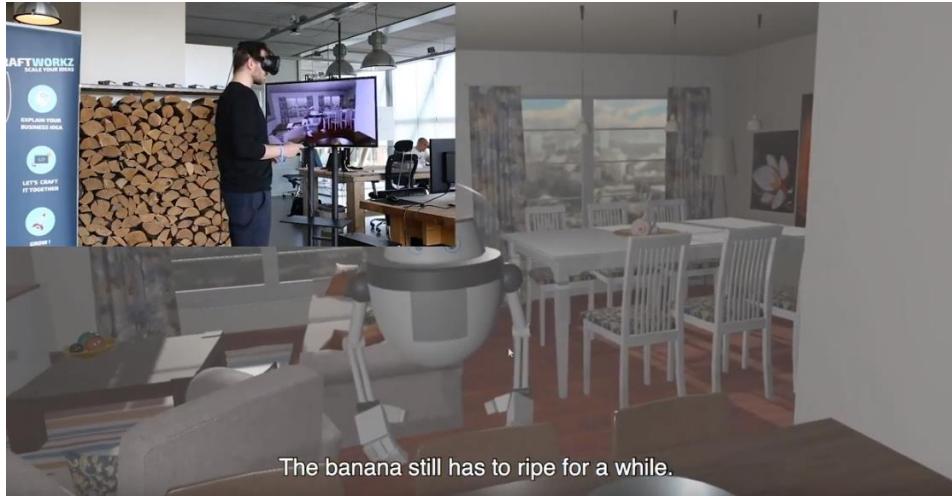
## 〈 챗봇 서비스 발전 단계 〉

구분 \ 단계	1단계	2단계	3단계
	챗봇 서비스(Chatbot)	지능형 비서(Intelligent Assistant)	감성 비서(Conscious Assistant)
제공방식	텍스트, 음성	텍스트, 음성, 시각자료	텍스트, 음성, 시각자료, 행동인지
입력방식	폐쇄형, 개방형 일부	폐쇄형 일부, 개방형	개방형·폐쇄형 복합사용
주요기술	패턴매칭, 키워드 및 연관어 추출 등	딥러닝·머신러닝, 자연어 처리, 기타 신기술 융합 등	감성인지기술, 데이터 정형화 기술
내용	<ul style="list-style-type: none"><li>- 학습된 내용에 대한 질의응답</li><li>- 사용자와 단순한 형태의 소통</li><li>- 검색을 통한 결과 제공</li></ul>	<ul style="list-style-type: none"><li>- 사용자의 패턴·상황을 고려한 개인 맞춤형 서비스 제공</li><li>- 간단한 업무 처리</li></ul>	<ul style="list-style-type: none"><li>- 감정 교류를 통한 서비스 및 각종 서비스에 대한 선제적 대응</li></ul>

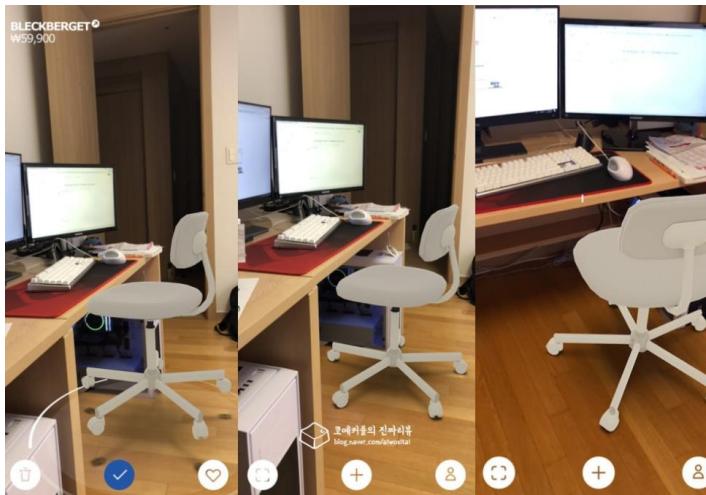
\* 일반적으로 입력방식은 챗봇이 제안하는 옵션 중 하나를 선택하는 폐쇄형 방식과 자연어를 입력하면 그 의미를 해석하고 대응하는 개방형 방식으로 구분<sup>60)</sup>

출처 : 서교리 외(2017), “인공지능 기반의 공공 지능형 가상 비서 서비스 발전모델 연구”, 한국통신학회, 수정 및 보완

# 가상현실 및 증강현실과 결합된 인공지능 스피커



<https://www.youtube.com/watch?v=WKcokBd5TJM>



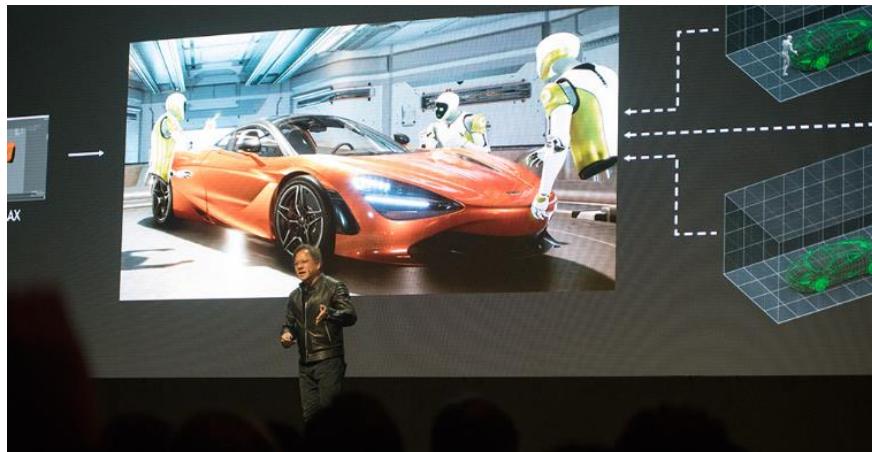
AR기반 가구 배치



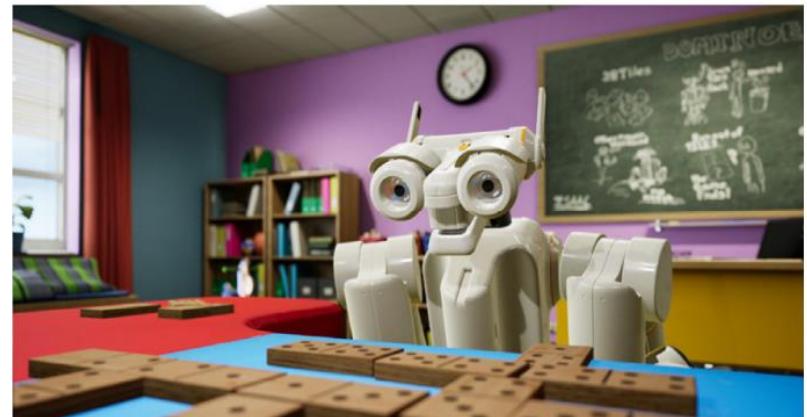
# 로봇과 대화를 통한 효율적인 시뮬레이션



- 3D 가상공간에서 다양한 캐릭터 모델을 공동으로 제작/시각화
- 현실세계에서 인간들과 안전하게 상호작용할 수 있는 로봇 개발
- 상호작용 관련 빅데이터를 이용하여 폭넓은 가상훈련과 테스트 진행
- 실제 환경에서 훈련하는 것보다 효율적으로 로봇을 훈련



홀로데크



〈자료〉 © NVIDIA



- 코퍼스(말뭉치)- 한글 어휘와 어휘 특성을 정의한 자료
- 한글 자연어 처리 기술이 어려운 이유
- 학습데이터 수집 및 시나리오 개발이 중요함

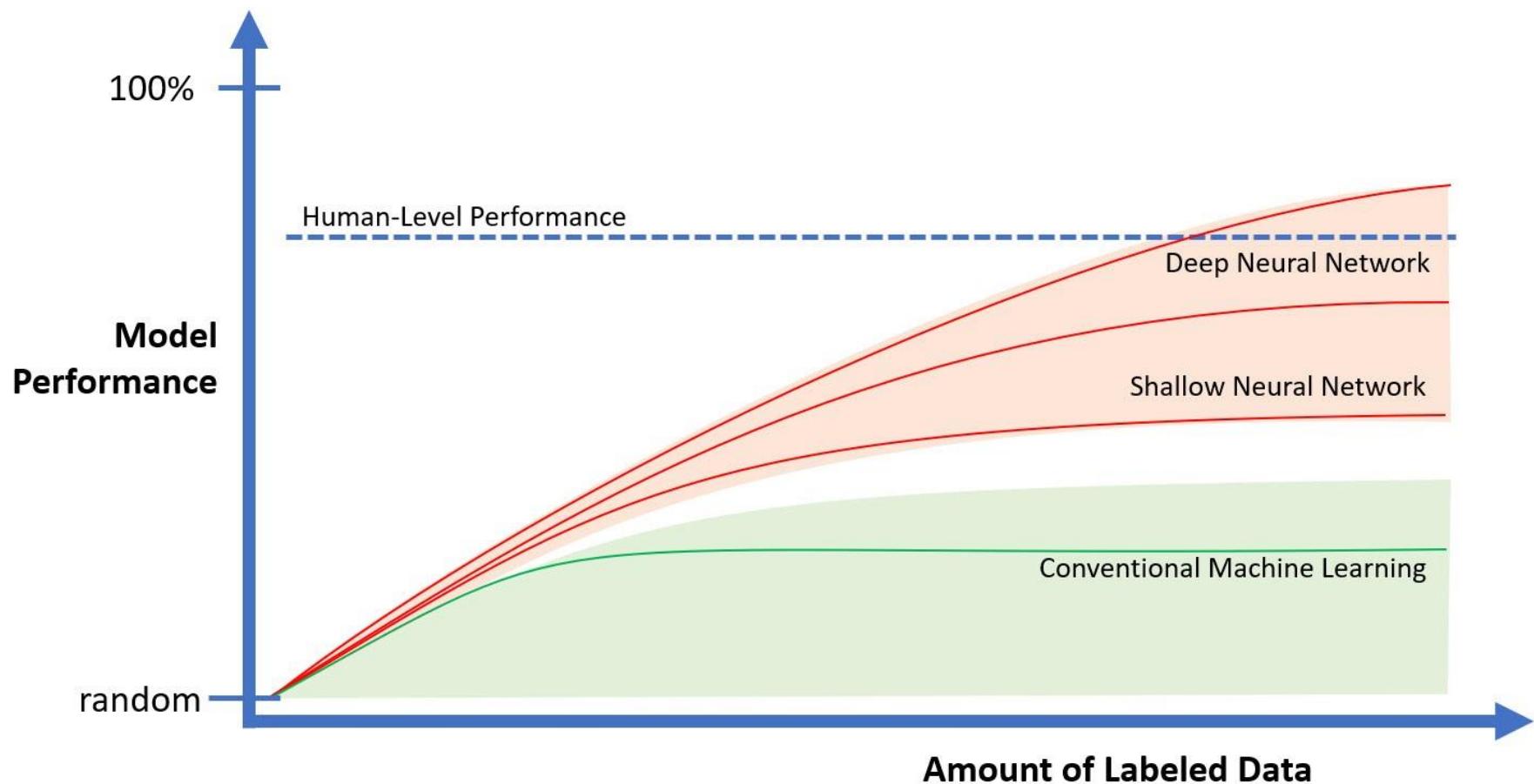


**What people think AI is about**



**The reality**

<https://hackernoon.com/%EF%88%8F-big-challenge-in-deep-learning-training-data-31a88b97b282>



<https://towardsdatascience.com/how-managers-should-prepare-for-deep-learning-new-values-f29a98b70bd8>

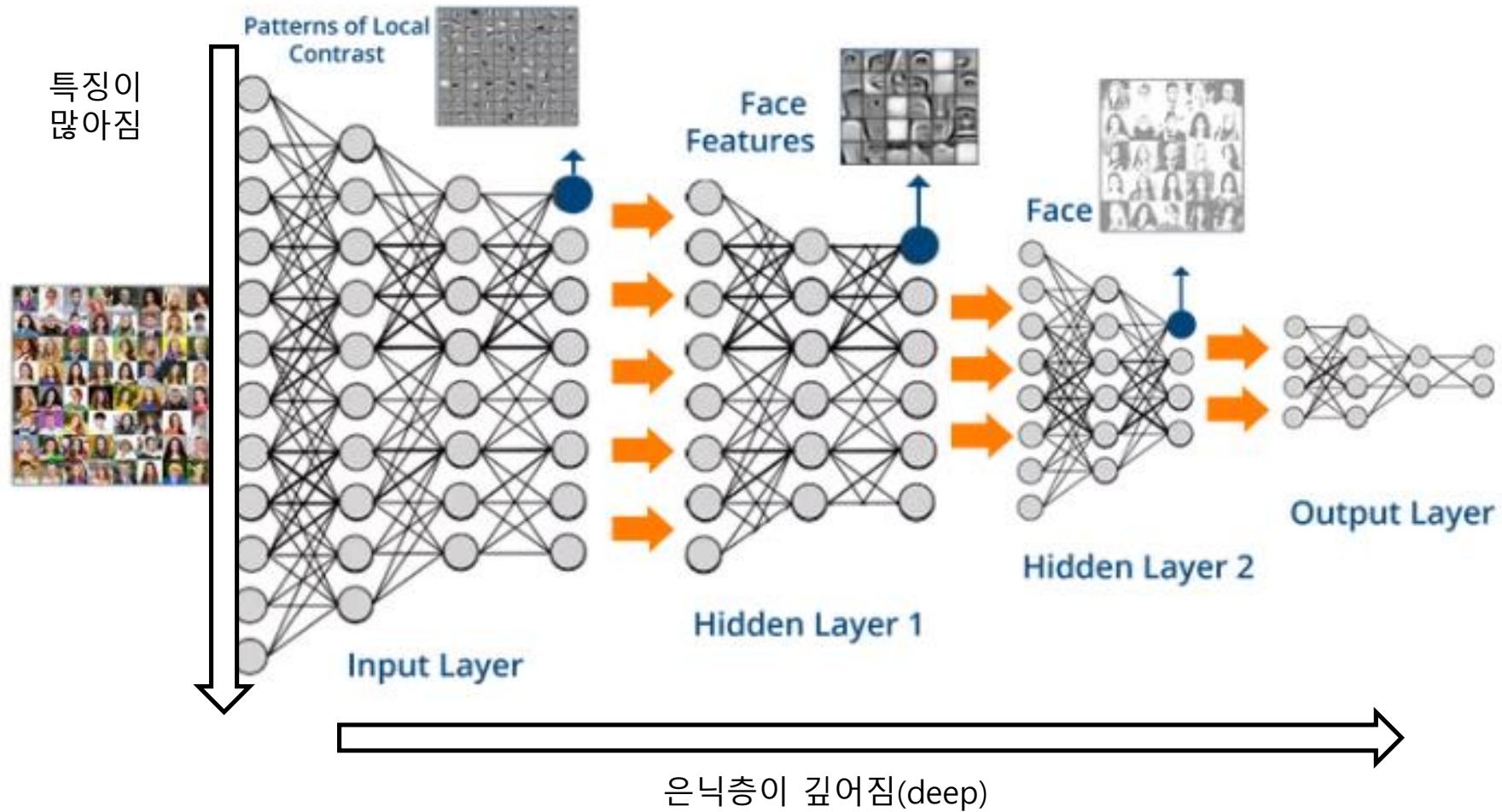


Image from: <https://cdn.edureka.co/blog/wp-content/uploads/2017/05/Deep-Neural-Network-What-is-Deep-Learning-Edureka.png>

# 패러다임의 전환

- 과거 인공지능
  - 전문가에 의한 전문가 시스템
  - 입력한 대로만 결과로 나옴
- 딥러닝 기반 인공지능
  - 가공되지 않은 방대한 data를 가지고 학습
  - 사람에 의존 하지 않음
  - 학습 결과를 해석할 수 없음
- 왜 다시 딥러닝인가?
  - 기존 인공신경망 모델의 단점 극복
  - 하드웨어의 발전(고성능GPU)
  - 빅데이터/다량의 복합 데이터

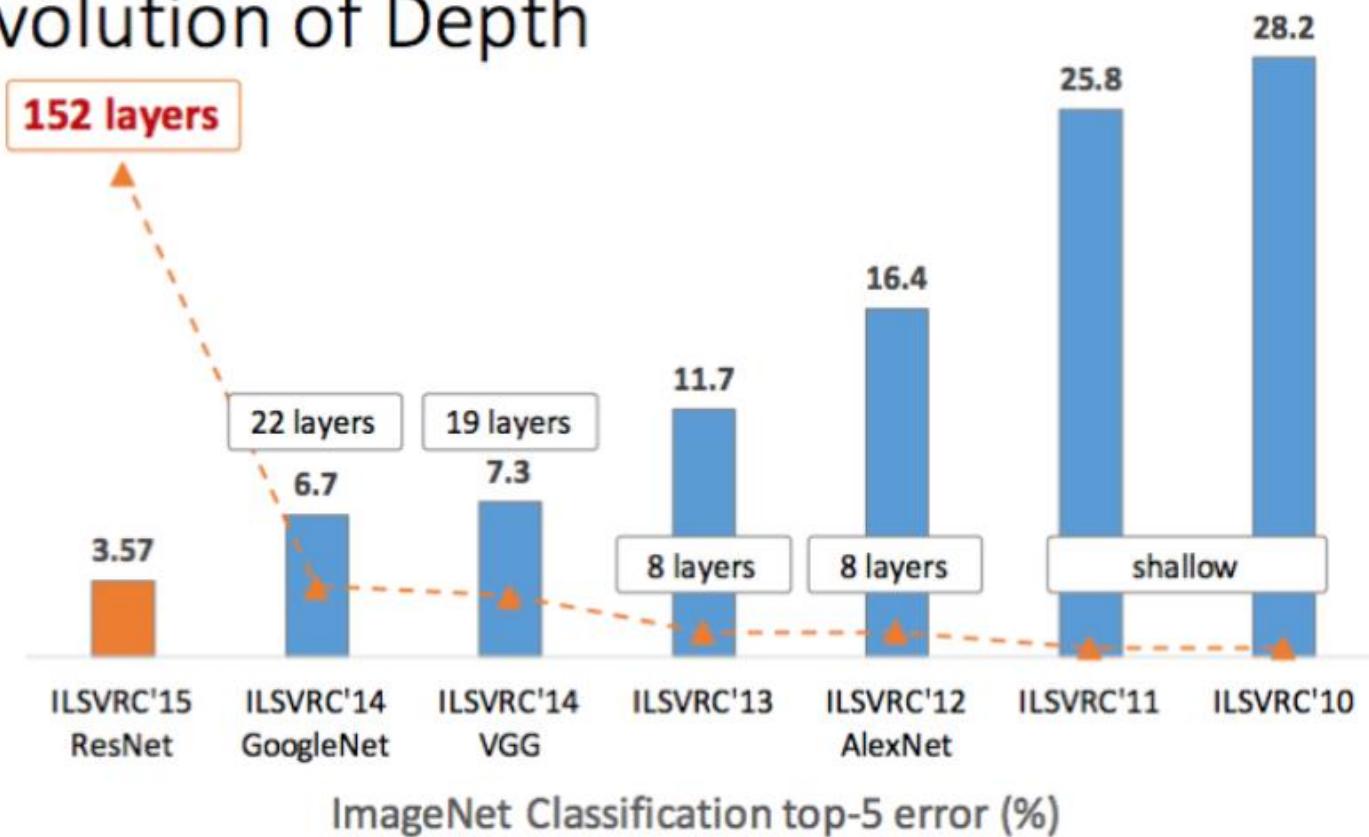


<https://www.youtube.com/watch?v=WmW6SD-EHVY>

# 딥러닝 얼마나 깊은가?

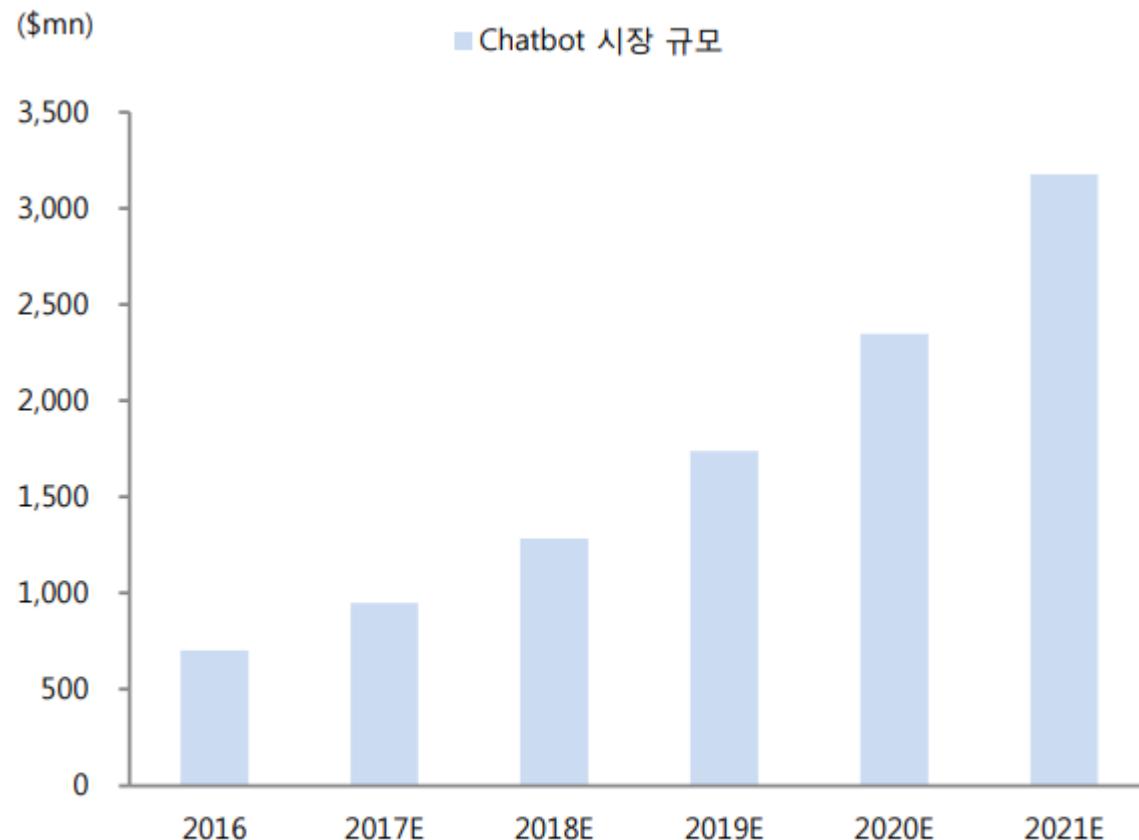
- Deep network

## Revolution of Depth



# 시장 규모

Chat Bot이 탑재된 AI 채팅 서비스 : 글로벌 시장은 2024년, 1조원 시장 예상



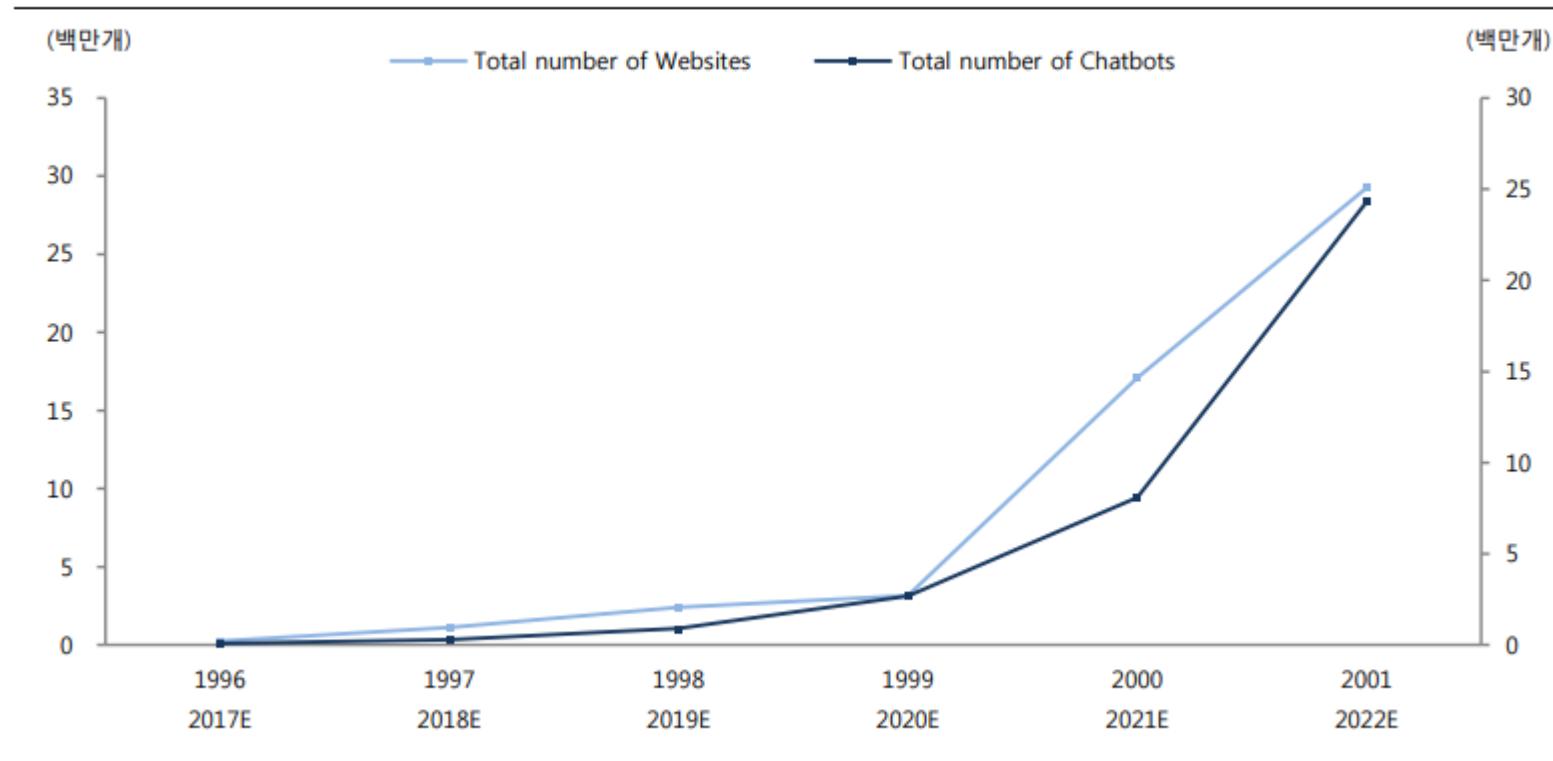
---

자료:technavio

# 시장 규모

챗봇 수는 2017년 15만개에서 2022년 2,430만개 수준 까지 증가 예상

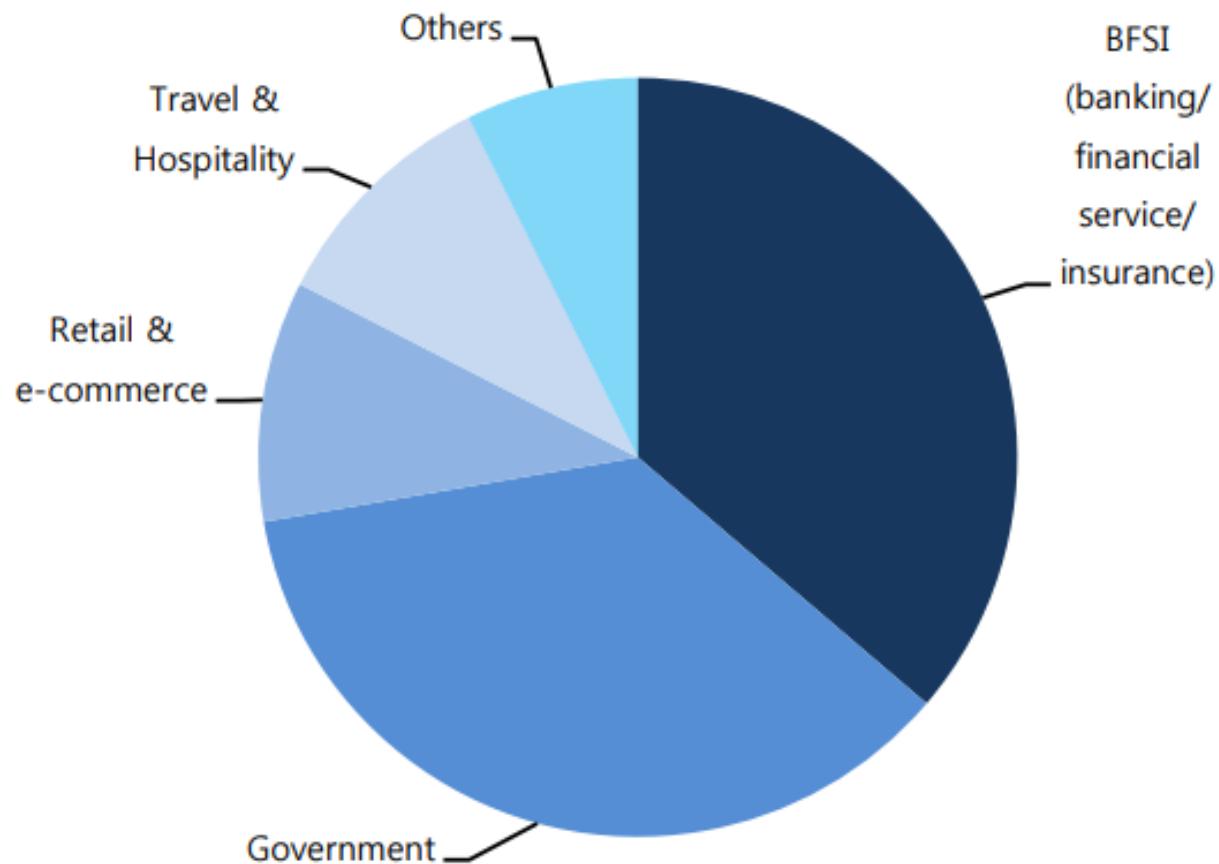
[그림 2] 글로벌 웹사이트 수(1996~2001년) 및 챗봇 수(2017~2022E) 전망 추이



자료:흥국증권 리서치센터

2015년 마이크로소프트(Microsoft: MS), 구글(Google), 페이스북(Facebook) 등 글로벌 기업이 챗봇 시장의 97.5%를 점유

# 챗봇 활용 분야



---

자료:technavio

| Chapter |

02

## Google Dialogflow

---

# 챗봇 동작 원리

## | 상담 대화 예시

(따르릉)

여행상담원 : 고객님과 함께하는 여행, 00여행입니다. 안녕하세요? 무엇을 도와드릴까요?

김단비씨 : 안녕하세요?

여행상담원 : 반갑습니다. 고객님~ 무엇을 도와드릴까요?

김단비씨 : 네, 내일 출발하는 뉴욕행 비행기표를 사려구요.

여행상담원 : 네 비행기표 예약하시려고 하시는군요?

출발지는 인천공항이신가요?

김단비씨 : 네. 인천이요.

여행상담원 : 잠시만요~

(잠시후...)

고객님, 확인해 봤는데 오후에는 비행기가 없고,

아침 5시 30분 A항공, 아침 7시 40분 B항공 2가지가 있네요!

(중략)

# 챗봇 동작 원리



## | 상담원과 챗봇 대화 비교

대화맥락	대화 예제	챗봇에서의 동작
Greeting Message (만남인사)	상담원 : 고객님과 함께하는 여행, OO여행입니다. 안녕하세요? 무엇을 도와드릴까요?	상담원이 준비된 멘트로 고객을 맞이하듯 서비스가 시작되면 챗봇이 준비된 멘트로 고객을 맞이합니다.
Intent Classification (의도분류) & Entity Extraction (핵심어 추출)	상황1 - 김단비씨 : 내일 출발하는 뉴욕행 비행기표를 사려구요. 상황2 - 김단비씨 : 비행기표를 사려구요.	상황1에서의 김단비씨의 의도는 "비행기표 예매"로 분류될 수 있고, 핵심어는 "도착지 : 뉴욕", "출발일 : 내일"이라는 것을 추출할 수 있습니다. 상황2에서의 김단비씨의 의도는 역시 "비행기표 예매"입니다. 그러나 핵심어는 나와있지 않네요.

# 챗봇 동작 원리



Slot Filling Prompt (부족한 정보 재질의)	상담원 : 네, 고객님. 출발지는 인천공항이신가요?	상황1에서는 비행기표 예매를 위한 필수값인 출발지를 되물어 봅니다. 상황2에서는 출발지외에 도착지와 출발일도 함께 물어보게 됩니다.
Slot Filling Answer (재질의 응답)	김단비씨 : 네, 인천이요	출발지가 인천이라는 것을 확인합니다. 상황2에서는 하나씩 차례대로 물어보고 필수 값을 받아낼 수 있습니다.
API Query (API 질의)	상담원 : 잠시만요~	상담원이 업무시스템을 통해 잔여 항공권을 조회합니다. 챗봇의 경우, 시스템의 API를 통해 상기 대화에서 입력받은 질의 조건을 기반으로 쿼리를 날립니다.
Result Speaking	상담원 : 고객님, 확인해 봤는데 오후에는 비행기가 없고, 아침 5시 30분 A항공, 아침 7시 40분 B항공 2가지가 있네요!	챗봇은 API를 통해 알아낸 여러가지 내용을 고객에게 제시합니다.



# Dialogflow

- Intent
- Entities
- Training
- Integrations
- Fulfillment
- Event



## Intent

Based on customer input

What does the user says/want?

What action should be taken?

Get Balance

Transfer Funds

Last Transactions

"How much money do I have in my saving account?"



## Entity

Variable/parameter for intent

Important word in an input

Adds relevance to intent

Possibly maps to domain object

Get Balance  
Entity:AccountType

Checking

Savings

Credit Card

"How much money do I have in my checking account?"



add training data that the agent learns from and uses to improve its performance.

provides an interface for incorporating both external and internal customer interaction logs into an agent's training phrases.

## Conversation list

The Training page shows a list of conversations. Logs for your agent's conversations automatically appear in the conversation list as customers chat with your agent. You can also upload log data captured outside of Dialogflow.

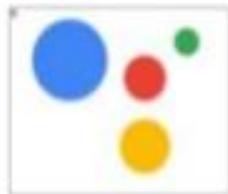
Conversation	Requests	No match	Date	C
I need to repair my bike	27	6	Aug 25	>
when are you open?	4	0	Aug 25	✓
Do you even have pizza?	17	0	Aug 25	>

- In order to serve the actual information the user is requesting, you'll need to setup fulfillment.
- Fulfillment is a method or a process which takes the user queries and replies the user appropriate response.

The screenshot shows the Dialogflow interface with the 'Fulfillment' tab selected in the sidebar. The main section is titled 'Fulfillment' and contains a 'Webhook' configuration. The 'Webhook' section is enabled, indicated by a blue toggle switch. Below it, a descriptive text explains that the web service will receive a POST request from Dialogflow. A red-bordered input field labeled 'URL\*' is present, with a placeholder 'Enter URL'. A note below the URL field states: 'This url will receive a POST request from Dialogflow every time the webhook is triggered by an intent.' The sidebar on the left lists other agent components: TestAgent, ko, Intents, Entities, Knowledge [beta], and Fulfillment, with 'Fulfillment' also being highlighted.

# Integrations

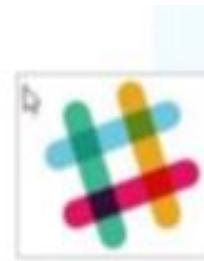
- Build once and deploy to multiple platforms using our one-click integrations.
- There are several social networking applications which supports for creating ChatBots



Google Assistant



Facebook Messenger



Slack



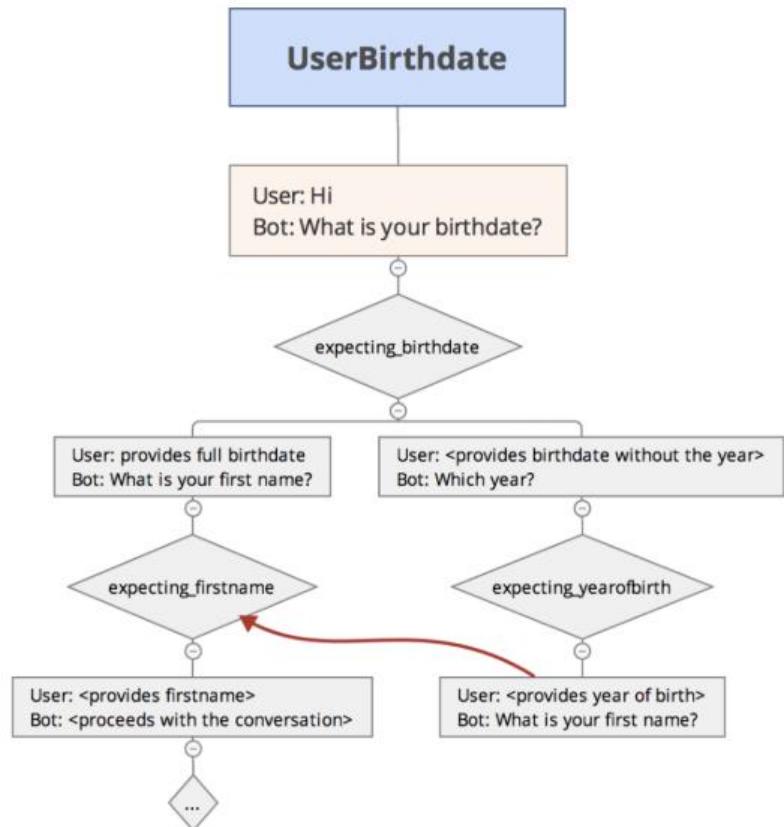
Skype



Telegram

# Event

- invoke intents based on something that has happened instead of what a user communicates.
- create your own custom events that can be triggered via fulfillment



# Agent 만들기



https://assistant.google.com/agent/editor/agents

FirstAgent WORKING... ⋮

DEFAULT LANGUAGE ⓘ

Korean (South Korea) – ko

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT+9:00) Asia/Tokyo

Date and time requests are resolved using this timezone.

GOOGLE PROJECT

New GCP project will be automatically linked to the agent after saving

# Default Intent



Intents

CREATE INTENT

Search intents

Default Fallback Intent

Default Welcome Intent

Add follow-up intent  

No regular intents yet. [Create the first one.](#)

# Fallback Intent



## Default Fallback Intent

SAVE



DEFAULT



### Text response



- 1 마지막에 말씀을 잘 못 이해 한것 같아요.
- 2 방금하신 말씀을 잘 못 알아들었어요.
- 3 죄송해요. 다시 들려 주실래요?
- 4 제가 제대로 이해하고 있는지 잘 모르겠어요.
- 5 제가 제대로 이해하지 못한것 같아요. 죄송해요.
- 6 조금 헷갈리는데요.
- 7 Enter a text response variant



ADD RESPONSES



Set this intent as end of conversation



# Welcome Intent



## Default Welcome Intent

SAVE



” Add user expression

Example mode is a quick way to test your intent without creating a full dialog. It's useful for quickly testing how your intent handles different user inputs.

” 오랜만이야



” 안녕

” 저기

” 잘 있었어

” 반가워

” 오랜만

” 어떻게 지냈어

# Welcome Intent



## Responses ?



DEFAULT



### Text response



- 1     안녕!
- 2     안녕하세요!
- 3     Enter a text response variant



**ADD RESPONSES**

# Welcome Intent



Try it now 

---

Agent

---

USER SAYS  COPY CURL

안녕?

---

 DEFAULT RESPONSE 

안녕하세요!

---

INTENT

Default Welcome Intent

# Web Demo

 Web Demo

Test the agent on its own page. Share the link to the page or embed the ` widget in other websites to get more conversations going. [More in documentation](#).

<https://bot.dialogflow.com/602bd859-5ffa-43bf-bce4-5f8b20040124> 

 Seems that your agent info is not filled yet. Set icon and description for better end-user experience. 

Add this agent to your website by copying the code below:

```
<iframe>
```

 CLOSE

3:41 / 4:58 https://bot.dialogflow.com/602bd859-5ffa-43bf-bce4-5f8b20040124 messaging) HD

# Web Demo

The screenshot shows a web browser window with the following details:

- Title Bar:** Dialogflow - FirstAgent
- URL Bar:** https://bot.dialogflow.com/602bd859-5ffa-43bf-bce4-5f8b20040124
- Header:** Dialogflow logo, API & DOCS, PRICING, BLOG, GO TO CONSOLE
- Main Content:** A large "FirstAgent" heading. Below it are three buttons: Embed, Email, and Share. To the right is a dark rectangular box containing the Dialogflow logo and the text "FirstAgent". Below this box is the text "Dialogflow 에 의해 작동됩니다".
- Bottom Navigation:** A red progress bar indicating 3:45 / 4:58. To the right are video control icons: back, forward, volume, and other standard media controls.

# 고장 접수 챗봇 시나리오



U : 고장 나서 접수 하려고 합니다.

B : 어떤 기능이 고장인가요?

U : 무슨 기능이 있는지 모르겠습니다.

B : 그럼 어느 기능이 문제인지 먼저 확인해주세요.

U : 모니터요 | 소리가 안나와요

B : 그럼 모니터기능에 대해 고장 접수 할까요?

U : 네 | 접수 할께요

B : 모니터 기능 고장 접수가 완료되었습니다.

# 음식 주문 시나리오

U : 사장님, 주문할께요

B : 메뉴를 말씀해 주세요.

U : 짜장면요

B : 몇 개요?

U : 2개요

B : 네. 짜장면 2개 주문 하셨습니다.

U : 보통이요.

B : 네, 짜장면 보통으로 주문 하셨습니다.



# 고장 접수 Intent



- failureReception

SAVE

Contexts



Events



Training phrases

Search training phrases



Add user expression

고장 접수 절차가 어떻게 되나요

고장 접수 가능한가요

고장접수 어떻게 하나요

고장 나서 접수 하려고 합니다.

# 고장 접수 Intent



## Responses ?



DEFAULT +

Text response		<span>?</span>	<span>trash</span>
1	어떤 기능이 고장났나요?		
2	Enter a text response variant		<span>▼</span>

ADD RESPONSES

Set this intent as end of conversation ?



## Intents

Search intents

Default Fallback Intent

Default Welcome Intent ▾

failureReception ^

↳ failureReception - custom

# 접수 실패 Follow-up Intent

- failureReception - unknown

SAVE



failureReception-followup X Add input context

Add output context X

Events ? ▼

Training phrases ?

Search training phrases Q ^

“” Add user expression

“” 몰라요

“” 잘모르겠습니다

“” 무슨 기능인지 모르겠습니다.

# 접수 실패 Follow-up Intent

## Responses ?

DEFAULT +

### Text response



- 1 그럼 어느 기능이 문제인지 확인해주세요
- 2 Enter a text response variant

ADD RESPONSES

# 증상 Follow-up Intent

- failureReception - failure

SAVE



## Contexts

failureReception-followup Add input context

Add output context

## Events

## Training phrases

Search training phrases

” Add user expression

” 전원이 안들어와요

” 소리가 안나와요

” 모니터요

# 증상 Follow-up Intent

- failureReception - failure

SAVE

Enter name

Enter entity

Enter value

+ New parameter

## Responses



DEFAULT



### Text response



- 그럼 기능에 대해 고장 접수를 할까요?
- 기능에 대한 고장으로 접수해도 될까요?
- 기능에 대한 고장이 맞으시죠?
- Enter a text response variant



ADD RESPONSES



Set this intent as end of conversation

# 접수 확인 Follow-up Intent

- failureReception - failure - confirm

SAVE



## Contexts



failureReception-failure-followup Add input context

Add output context

## Events



## Training phrases

Search training phrases



Add user expression

접수할께요

신청할께요

네.

# 접수 확인 Follow-up Intent

## Responses

DEFAULT 

### Text response



- 1 #failureReception-failure-followup.FailureName 고장접수 완료되었습니다.
- 2 Enter a text response variant  

[ADD RESPONSES](#)

# Entity 만들기

**i** Dialogflow API V1 will be deprecated on October 23rd, 2019. Learn how to migrate to API V2 [here](#).

The screenshot shows the Dialogflow interface with the 'Entities' tab selected in the sidebar. The main area displays a search bar with 'Search entities' placeholder text and a magnifying glass icon. Below the search bar is a list containing a single entity entry: '@ food'. The sidebar also includes links for 'TestAgent', 'ko', 'Intents', 'Entities' (which is highlighted in blue), and 'Knowledge [beta]'. A 'CREATE ENTITY' button is located in the top right corner of the main content area.

# 고장 증상 entity

FailureName

DONE



Define synonyms   Allow automated expansion

모니터	모니터
소리	소리
전원	전원
Click here to edit entry	

# 고장 증상 entity



Dialogflow

TestAgent

ko

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

History

Analytics

Prebuilt Agents

- failureReception - failure

Training phrases

” Add user expression

” 전문가의 조언을 듣기

” 소리

” 모니터링

@FailureName

@sys.number-integer

Action

” @sys.temperature

” @sys.geo-country

failureReception

” @sys.unit-information

REQUIRE

” @sys.geo-capital

# 고장 증상 entity 설정

- failureReception - failure

SAVE

## Training phrases ?

Search training phrases 🔍



” Add user expression

” 전원이 안들어와요

” 소리가 안나와요

” 모니터요

PARAMETER NAME	ENTITY	RESOLVED VALUE	
FailureName	@FailureName	모니터	×

# 고장 증상 entity 설정

- failureReception - failure

SAVE



<input type="checkbox"/>	FailureName	@FailureName	\$FailureName	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

## Responses



DEFAULT +

Text response			
1	그럼 \$FailureName 기능에 대해 고장 접수를 할까요?		
2	\$FailureName 기능에 대한 고장으로 접수해도 될까요?		
3	\$FailureName 기능에 대한 고장이 맞으시죠?		
4	Enter a text response variant		



**Dialogflow**

TestAgent ko +

Intents +

Entities +

Knowledge [beta]

Fulfillment

**Integrations**

Training

History

Analytics

Prebuilt Agents

**Web Demo** toggle

Test the agent on its own page. Share the link to the page or embed the ` widget in other websites to get more conversations going. [More in documentation.](#)

<https://bot.dialogflow.com/614f424e-e212-424c-9d77-2376e29e86e5> edit

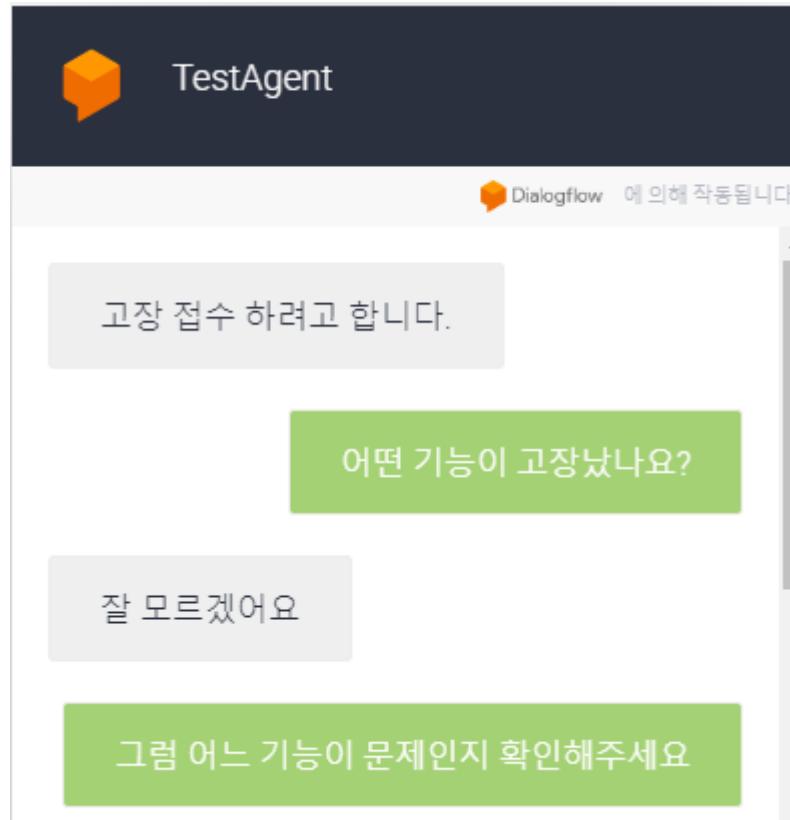
i Seems that your agent info is not filled yet. Set icon and description for better end-user experience. gear

Add this agent to your website by copying the code below:

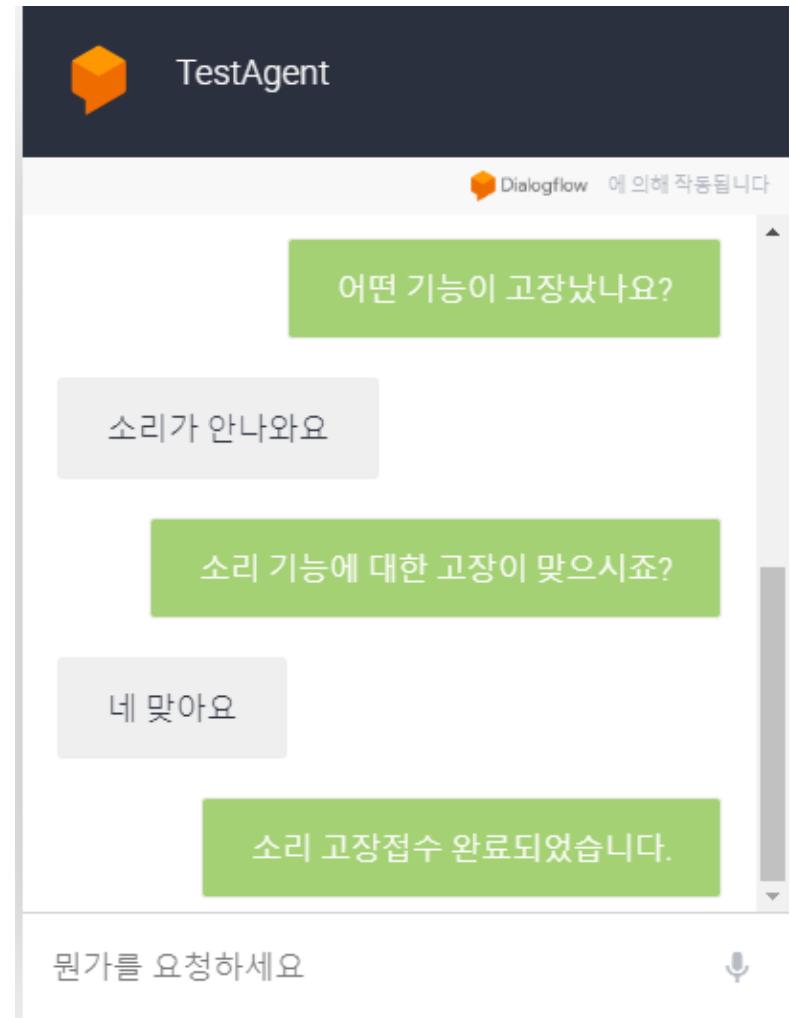
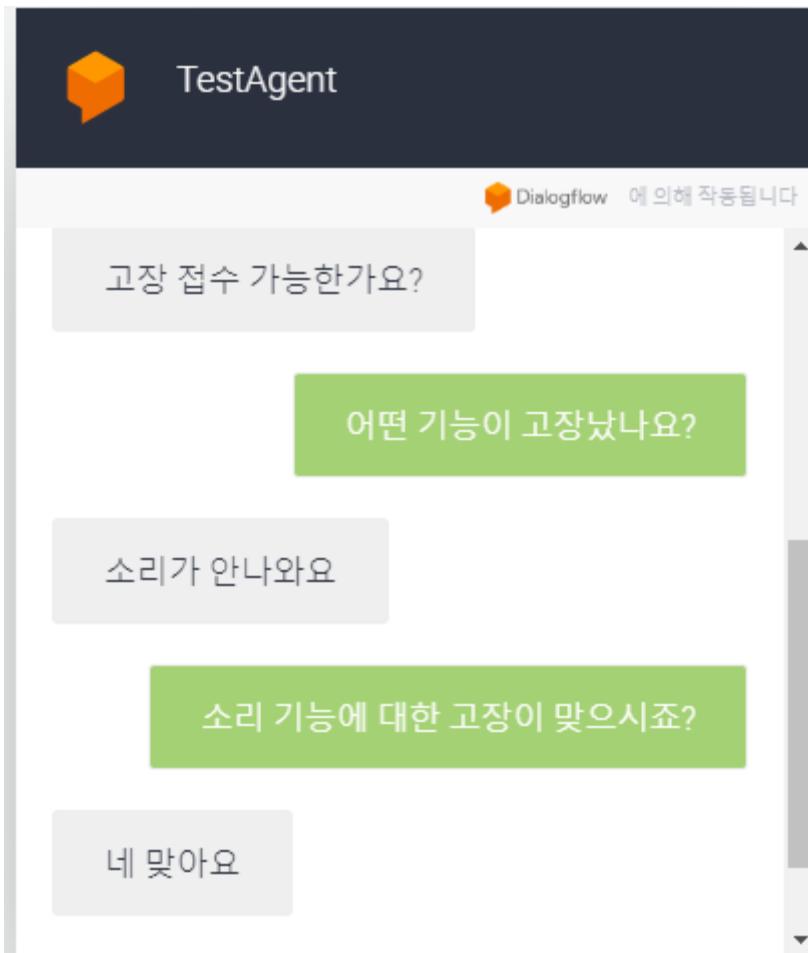
```
<iframe  
    allow="microphone;"  
    width="350"  
    height="430"  
    src="https://console.dialogflow.com/api-client/demo/embedded/614f424e-e212-424c-9d77-2376e29e86e  
5">  
</iframe>
```

copy

**CLOSE**



# 웹 테스트





## <대화주제>

- 인천의 날씨는 어때?
- 오늘 부산의 날씨는 어때?
- 내일 서울의 날씨는 어때?
- 다음주 대전의 날씨는?



## Weather

SAVE

### Contexts



### Events



### Training phrases

Search training phrases

” 오늘 서울 날씨 어때?

PARAMETER NAME	ENTITY	RESOLVED VALUE	
date	@sys.date	오늘	
geo-city	@sys.geo-city	서울	

” 서울 날씨 좋아?

” 서울 날씨는 어때?

” 서울 날씨는?



## • Weather

SAVE

⋮

” 다음주 서울 날씨 어때?

PARAMETER NAME	ENTITY	RESOLVED VALUE	X
date-time	@sys.date-time	다음주	×
geo-city	@sys.geo-city	서울	×

” 오늘 서울 날씨 어때?

” 서울 날씨 좋아?

” 서울 날씨는 어때?

” 서울 날씨는?



- Weather

SAVE+ New parameter

## Responses

DEFAULT

### Text response

- 1 \$geo-city 날씨는 항상 좋아요~
- 2 \$date의 \$geo-city 날씨는 맑아요
- 3 \$date-time \$geo-city 날씨 비와 끄끅
- 4 Enter a text response variant



# 날씨 챗봇

 TestAgent

Dialogflow 에 의해 작동됩니다

서울 날씨 어때?

서울특별시 날씨는 항상 좋아요~

부산 날씨는?

부산광역시 날씨는 항상 좋아요~

뭔가를 요청하세요

microphone icon

 TestAgent

Dialogflow 에 의해 작동됩니다

서울특별시 날씨는 항상 좋아요~

부산 날씨는?

부산광역시 날씨는 항상 좋아요~

오늘 광주 날씨는 어때?

2018-12-18의 광주광역시 날씨는 맑아요

뭔가를 요청하세요

microphone icon

# 날씨 챗봇



 TestAgent

 Dialogflow 에 의해 작동됩니다

다음주 서울 날씨 어때?

2018-12-24/2018-12-30 서울특별시 날씨  
비와ㅠㅠ

다음주 인천 날씨는 어떨까?

2018-12-24/2018-12-30 인천광역시 날씨  
비와ㅠㅠ

|뭔가를 요청하세요





## <대화주제>

- 12월 25일 7시에 예약해주세요.
- 다음 주 7시에서 9시까지 예약해주세요.

# 날짜 및 시간 예약 챗봇



- reservation

SAVE



〃 Add user expression

〃 내일 오후 7시에서 10시까지 예약할께요

〃 다음주 7시 예약해 주세요

## Action and parameters



Enter action name

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
<input type="checkbox"/>	date-time	@sys.date-time	\$date-time	<input checked="" type="checkbox"/>
<input type="checkbox"/>	time-period	@sys.time-period	\$time-period	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

# 날짜 및 시간 예약 챗봇

## Responses



DEFAULT 

### Text response

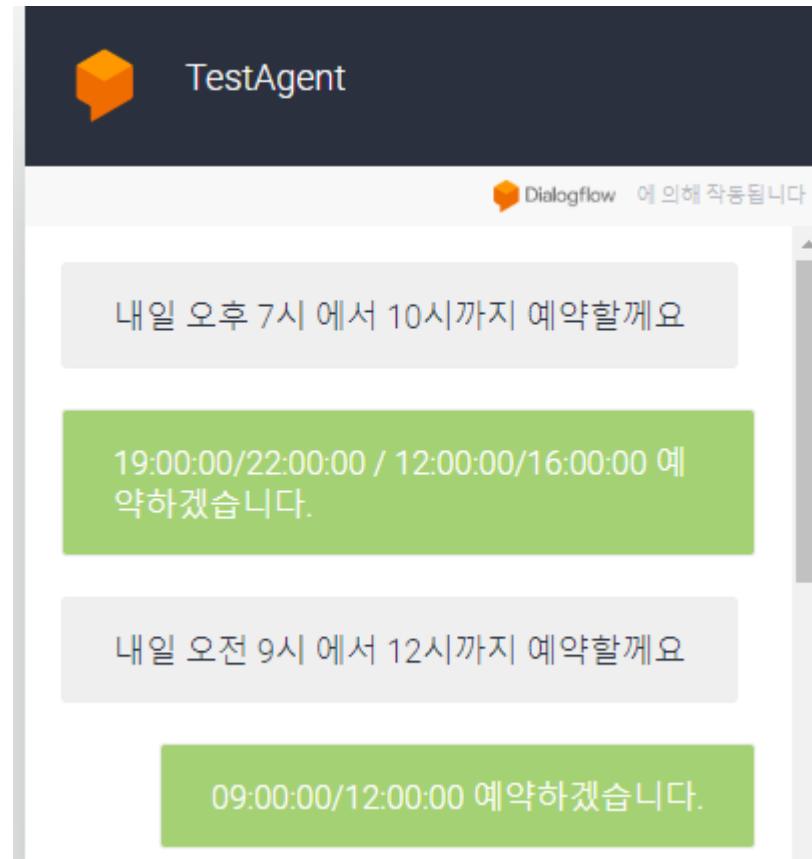
 

- 1 Sdate-time / Stime-period 예약하겠습니다.
- 2 Sdate-time 예약하겠습니다.
- 3 Stime-period 예약하겠습니다
- 4 Enter a text response variant



 ADD RESPONSES

# 날짜 및 시간 예약 챗봇



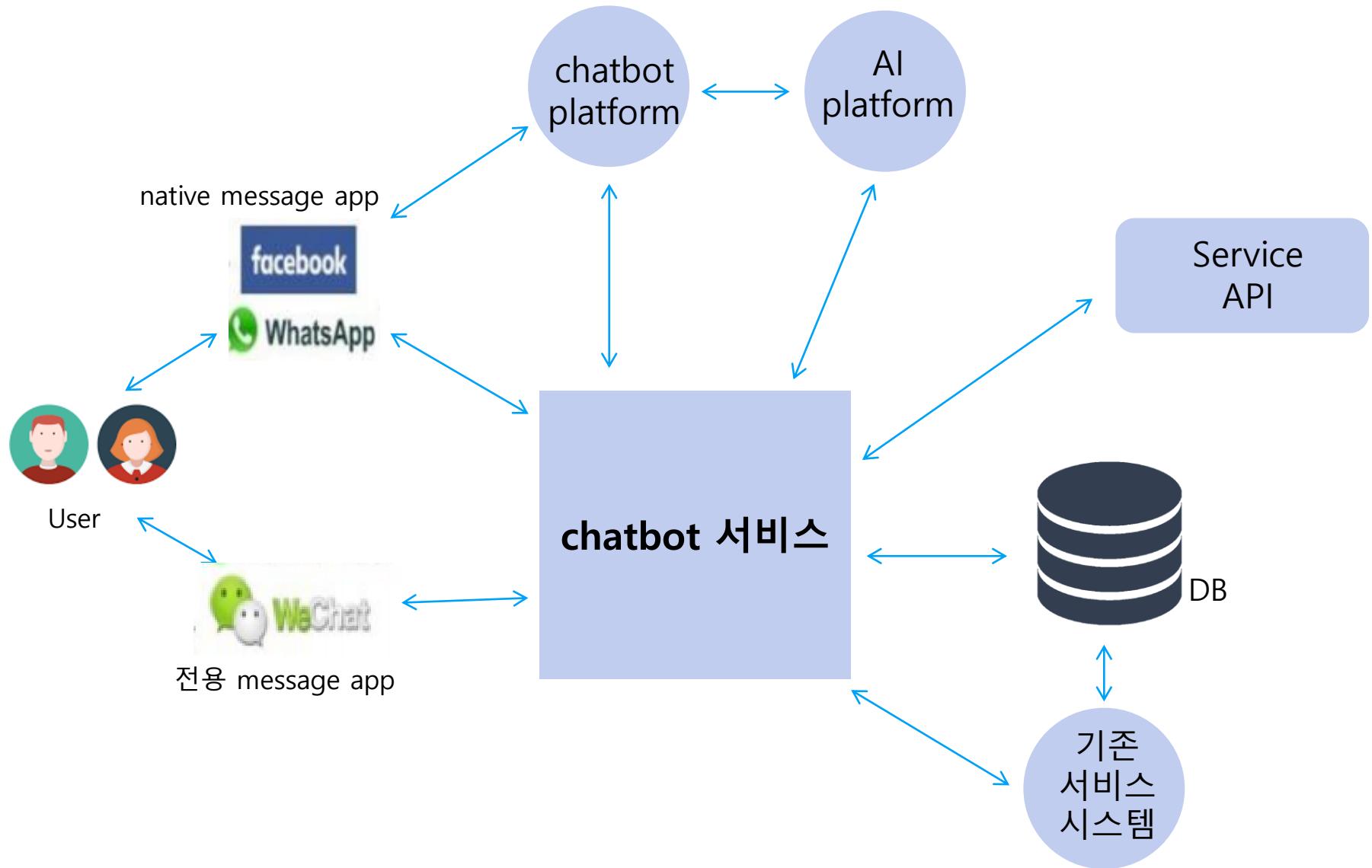
| Chapter |

# 03

## Web Platform 이해

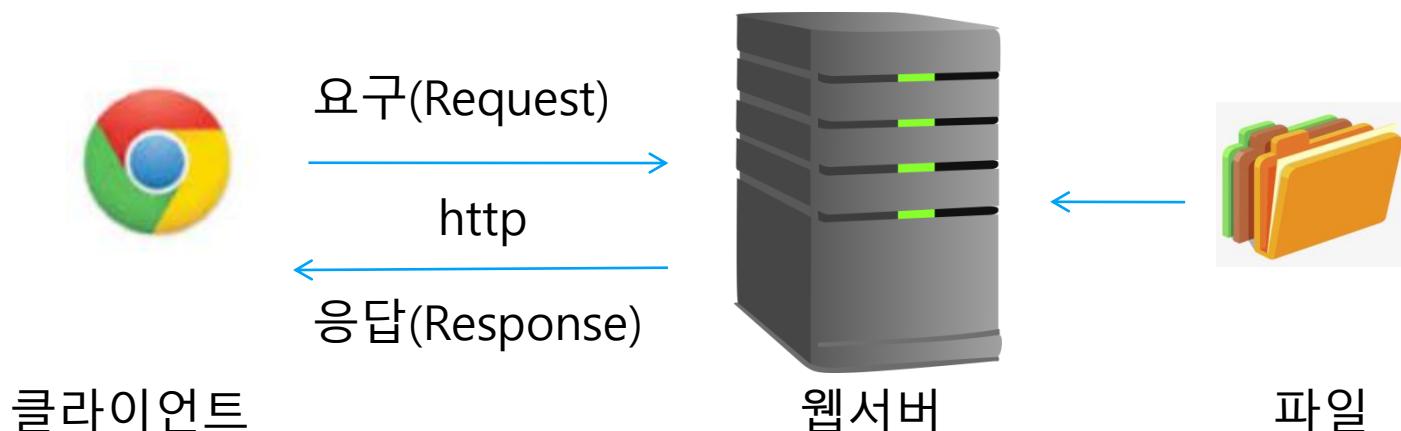
---

# 챗봇 platform



# 웹 서비스

- 웹 서버는 웹 클라이언트로부터 전송되어 오는 서비스 요청에 대한 결과인 HTML문서를 해당 웹 클라이언트에 제공
- 웹 브라우저 : HTML로 작성된 하이퍼텍스트 문서를 해석하여 화면에 출력
- 가장 대표적인 웹서버로 아파치 사용
  - 안정성, 확장성을 고려한 OpenSource 프로젝트



- Python에서는 경량 web server를 위해 세가지 방법 제공
  - 기본 웹서버(SimpleHTTPServer)
    - python -m http.server 9000
  - 스크립트 기반 웹 서버

```
import SimpleHTTPServer  
import SocketServer
```

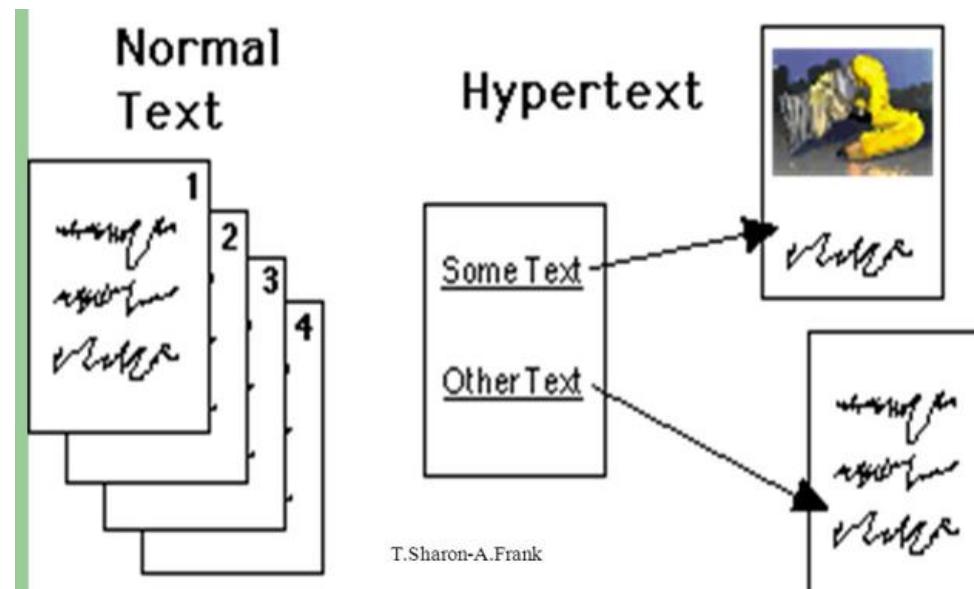
```
Handler = SimpleHTTPServer.SimpleHTTPRequestHandler  
httpd = SocketServer.TCPServer(("", 9000), Handler)  
httpd.serve_forever()
```

- Web Framework기반 웹서버
  - Flask : 경량 웹 서버
  - Django : MVC 기반의 웹 서버

# HTML 이란



- 많은 수의 문서를 하이퍼 링크로 서로 연결시킬 수 있는 문서의 구조
- 참조([하이퍼링크](#))를 통해 한 문서에서 다른 문서로 즉시 접근할 수 있는 텍스트
- 하이パーテქ스트는 1960년대 컴퓨터 개척자이자 철학자인 [ted nelson](#)이 처음 고안

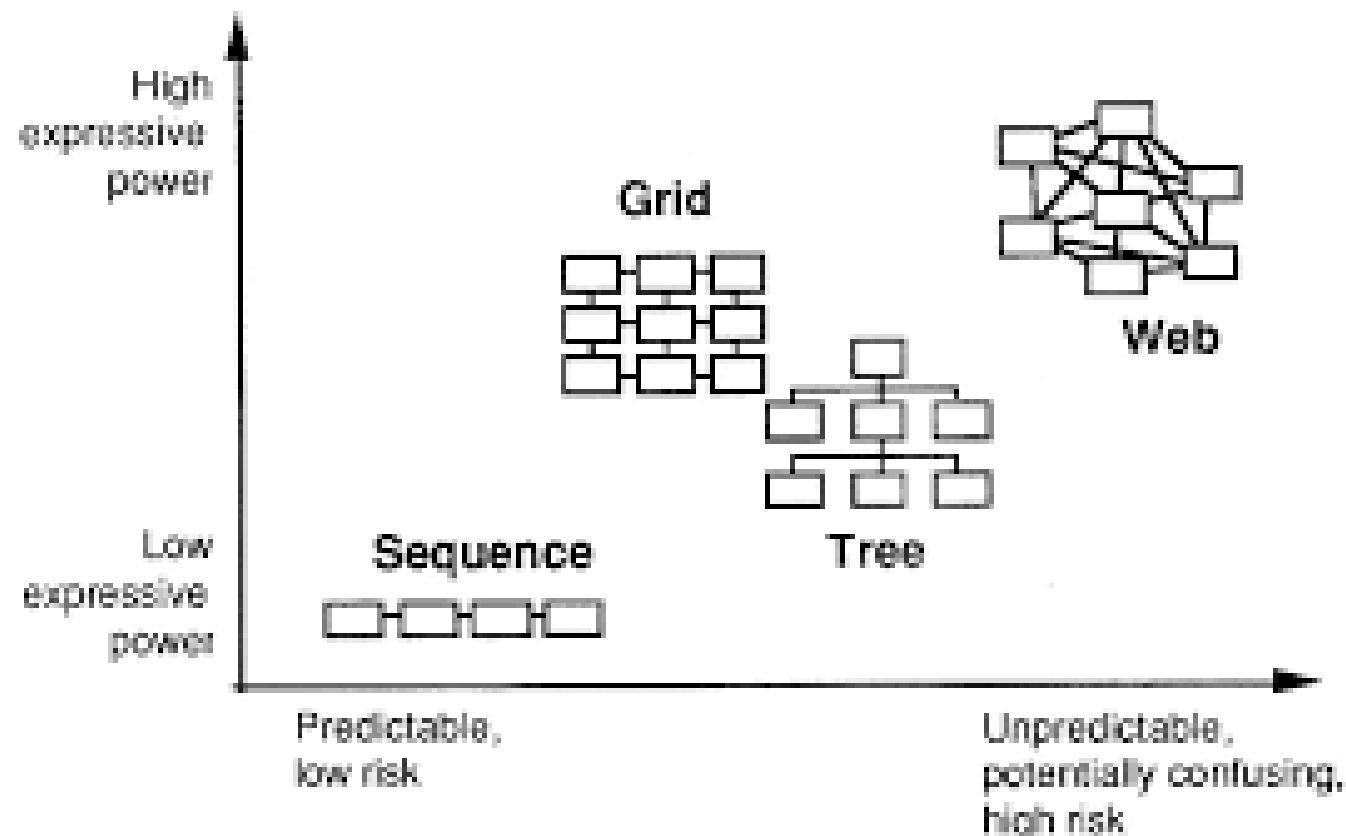


hyper(건너 편의, 초월, 과도한)

<A HREF="page.html"> label </A>

# 문서의 구조

- 기존 문서 : 순차적, 서열형 구조
- 하이퍼텍스트 : 링크에 따라 그 차례가 바뀌는 임의적이면서 나열형 구조
- 출판된 책처럼 작가의 의도대로 사용자가 따라가는 것이 아닌, 하이퍼링크로 연결된 문서들을 어떠한 행위(클릭)에 따라 자유롭게 이동
- 검색엔진과 더불어 정보습득의 새로운 장을 인류에게 가져다 줌



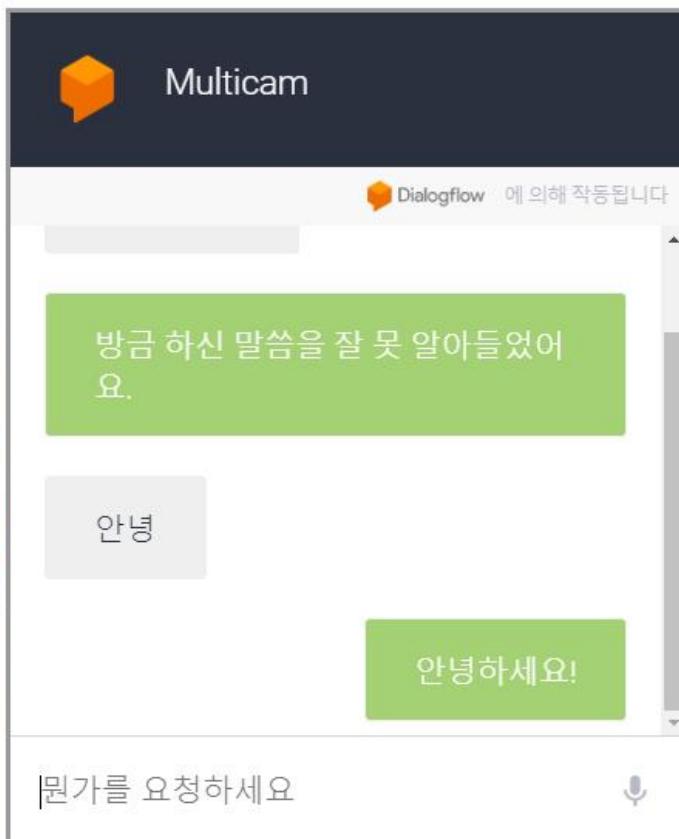
- tag 는 하나의 키워드
- opening 과 closing tags 이루어짐
- XHTML에서는 <img >가 아닌 <img /> 사용
- 중첩태그
  - <H1> <|> The Nation </|> </H1>
- 속성을 값을 설정하기 위해 “와 ‘ 모두 사용
- 주석
  - <!-- .... -->

# HTML을 이용한 챗봇 연동

Hello

Hello2

안녕하세요 google



# HTML을 이용한 챗봇 연동

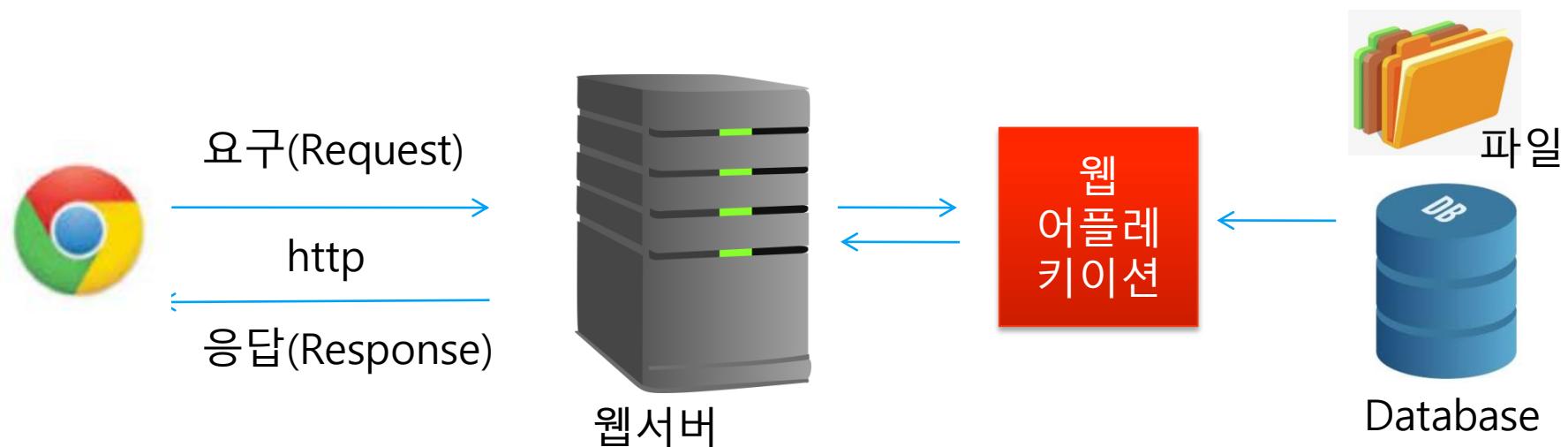


```
<h1>Hello </h1>
<h2>Hello2 </h2>

<font color=blue>안녕하세요</font>
<a href=http://www.google.co.kr>google</a><br>
<img src=image.jpg>
<iframe
    allow="microphone;"
    width="350"
    height="430"
    src="https://console.dialogflow.com/api-client/demo/embedded/efab931c-cb03-406d-ae09-da3af220b751">
</iframe>
```

# 웹 서버 어플리케이션

- 웹을 기반으로 서버실행되는 프로그램
- 동적 HTML 생성 : 웹 서버 애플리케이션은 HTML을 리턴
- 웹 서버 애플리케이션은 파일, 데이터베이스 등 기존 서버 자원 활동





- **H**yper**T**ext **T**ransfer **P- 웹에서 정보를 주고 받을 수 있는 가장 기본적인 프로토콜(1996)
- HTML 포맷을 사용한 text 및 멀티미디어 자료 공유
- TCP(80번 port) 프로토콜을 사용한 응용 계층
- http:로 시작하는 URL을 통해 접속
- 대규모 접속을 지원하기 위한 접속을 유지 하지 않는 stateless 프로토콜
- 클라이언트와 서버 사이에 이루어지는 요청/응답 프로토콜**

# HTTP Method(GET/POST)

- 웹서버에 요청하는 방식
  - 웹서버에 클라이언트의 data를 전송하기 위한 전송 방법
  - GET과 POST는 전달하는 방식에서 차이가 있음
- GET 방식 : URL을 통해 data 전송
  - GET방식으로 전송할수 있는 data크기는 한계가 있음
  - 전송 URL이 노출됨
  - <http://xxx/service?name1=value1&name2=value2>
  - GET /service?name1=value1&name2=value2
  - Host : xxx
- POST 방식 : Body를 통해 data 전송
  - 길이 제한이 없음
  - POST /service HTTP/1.1
  - Host : xxx
  - name1=value1&name2=value2

# 검색 서비스 URL 분석

- Google 검색
  - <https://www.google.co.kr/search?q=python>
- Naver 검색
  - <https://search.naver.com/search.naver?&query=%EC%9D%8C%EA%B3%B5%EC%A7%80%EB%8A%A5>
- Naver 음악차트
  - <https://search.naver.com/search.naver?query=%EC%9D%8C%EC%95%85%EC%B0%A8%ED%8A%B8>
- Daum 영화 검색
  - <http://movie.daum.net/premovie/released>
  - <http://movie.daum.net/moviedb/main?movield=108035>



- 누구나 사용할 수 있도록 공개된 API
- 주로 웹 서비스와 연동
  - REST(Representational State Transfer)
  - SOAP(Simple Object Access Protocol)
  - 사용자 요청 data : GET/POST 사용
  - 출력 : HTML보다는 데이터 처리가 가능한 XML, JSON 포맷
- 누구에게나 공개되어 있기 때문에 보안을 위해 인증 필요
  - 공개키를 사용해 사용자 인증 및 추적
  - 사용량에 따라 무료/유료 운영



- Extensible Markup Language
- 사람과 기계가 모두 읽을 수 있는 데이터를 기술하는 데 사용
- 태그/속성/text로 이루어짐

rss 태그

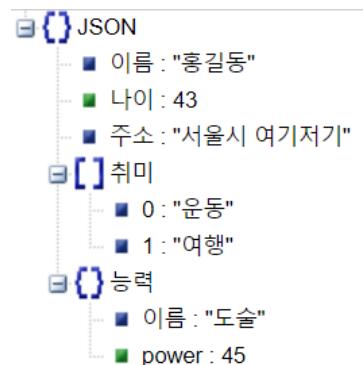
version 속성

text

```
<rss version="2.0">
  <channel>
    <title>Naver Open API - blog :: 'Hello'</title>
    <link>http://search.naver.com</link>
    <description>Naver Search Result</description>
    <lastBuildDate>Wed, 13 Jun 2018 17:17:03 +0900</lastBuildDate>
    <total>625263</total>
    <start>1</start>
    <display>10</display>
    <item>
      <title><b>Hello</b> HERO, 게이밍 마우스 로지텍 G304</title>
      <link>http://blog.naver.com/PostView?PostId=10251002182792</link>
```



- JavaScript Object Notation
- 경량의 DATA 교환 형식
- 자바스크립트 언어로부터 파생되어 자바스크립트의 구문 형식을 따르지만 언어 독립형 데이터 포맷
- Name/Value 형식으로 데이터 정의
  - 객체와 array 개념 제공



name : value 표현

```
{
  "이름": "홍길동",
  "나이": 43,
  "주소": "서울시 여기저기",
  "취미": ["운동", "여행"],
  "능력": { "이름": "도술", "power": 45 }
}
```

array

객체

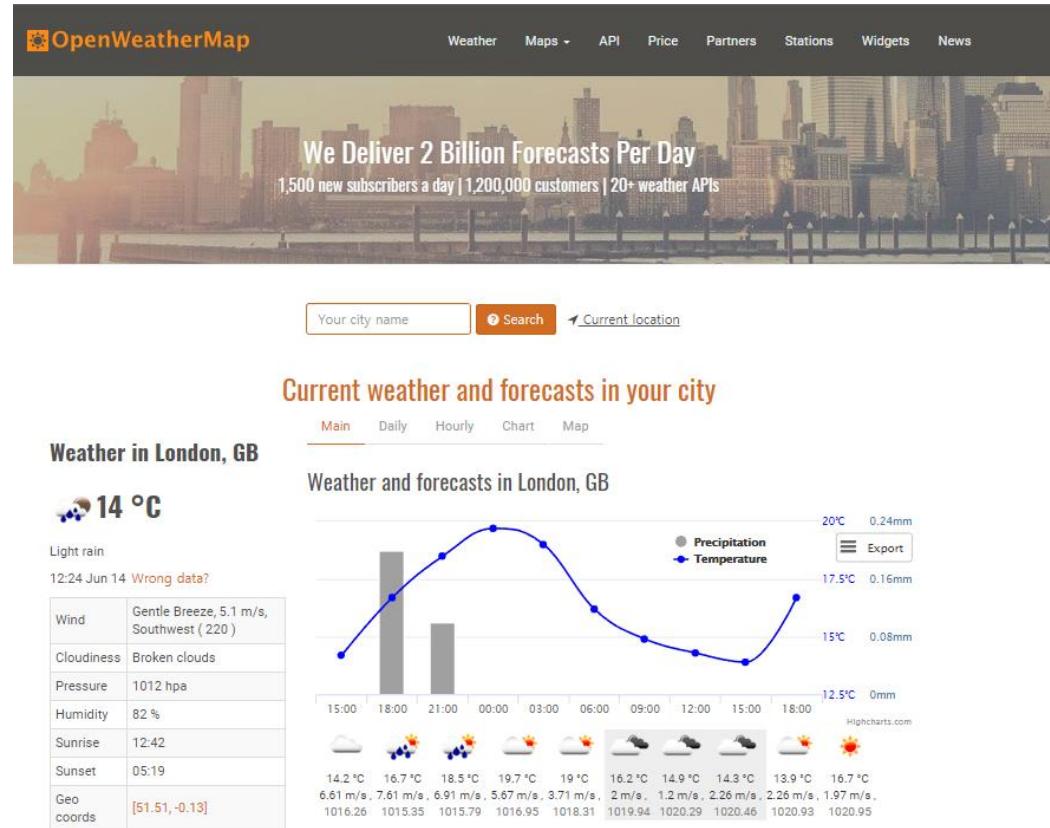
# Naver 검색 OpenAPI

## • \_로그인\_ 바시가 \_비로그인\_ 바시 \_지의

요청 URL	메서드	응답 형식	설명
<a href="https://openapi.naver.com/v1/search/news">https://openapi.naver.com/v1/search/news</a>	GET	JSON, XML	네이버 검색의 뉴스 검색 결과를 반환합니다.
<a href="https://openapi.naver.com/v1/search/encyc">https://openapi.naver.com/v1/search/encyc</a>	GET	JSON, XML	네이버 검색의 백과사전 검색 결과를 반환합니다.
<a href="https://openapi.naver.com/v1/search/blog">https://openapi.naver.com/v1/search/blog</a>	GET	JSON, XML	네이버 검색의 블로그 검색 결과를 반환합니다.
<a href="https://openapi.naver.com/v1/search/shop">https://openapi.naver.com/v1/search/shop</a>	GET	JSON, XML	네이버 검색의 쇼핑 검색 결과를 반환합니다.
<a href="https://openapi.naver.com/v1/search/movie">https://openapi.naver.com/v1/search/movie</a>	GET	JSON, XML	네이버 검색의 영화 검색 결과를 반환합니다.
<a href="https://openapi.naver.com/v1/search/webkr">https://openapi.naver.com/v1/search/webkr</a>	GET	JSON, XML	네이버 검색의 웹 문서 검색 결과를 반환합니다.
<a href="https://openapi.naver.com/v1/search/image">https://openapi.naver.com/v1/search/image</a>	GET	JSON, XML	네이버 검색의 이미지 검색 결과를 반환합니다.
<a href="https://openapi.naver.com/v1/search/doc">https://openapi.naver.com/v1/search/doc</a>	GET	JSON, XML	네이버 검색의 전문정보 검색 결과를 반환합니다.

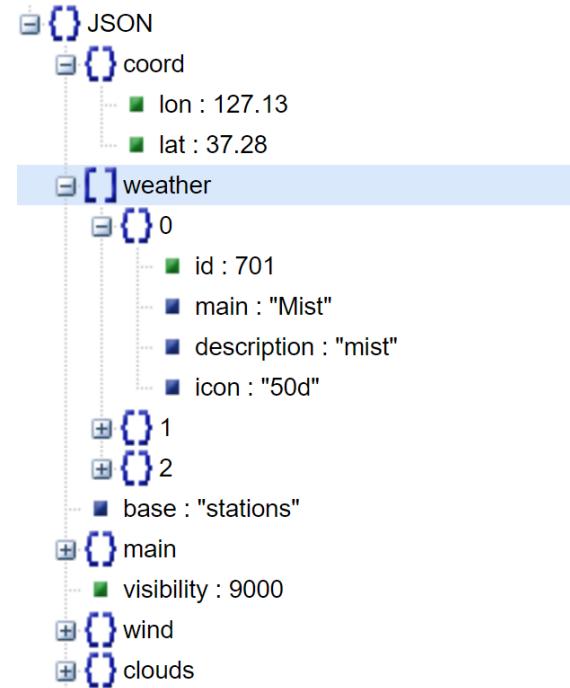
# 날씨 OpenAPI

- <https://openweathermap.org/>
- 날씨 정보 제공



# 날씨 OpenAPI

- 다양한 OpenAPI 제공
- 위도, 경도 좌표에 대한 날씨 정보 제공
- JSON 포맷 사용
- 날씨 아이콘 제공



<http://api.openweathermap.org/data/2.5/weather?lat=37.276101&lon=127.130824&APPID=342bd9672f19bbc63b...>

| Chapter |

# 04

## Open API 기반 챗봇

---

# Dialogflow API 호출 인증키(V1)

## TestAgent

SAVE

General Languages ML Settings Export and Import Speech Share

DEFAULT TIME ZONE  
(GMT+9:00) Asia/Tokyo  
Date and time requests are resolved using this timezone.

GOOGLE PROJECT  
Project ID testagent-cf7f4

API VERSION  
 V2 API Use Cloud API as default for the agent. Your webhook will receive V2 format requests and should return V2 format responses.  
 V1 API Legacy APIs

API KEYS (V1)  
Client access token [REDACTED]  
Developer access token [REDACTED]

TestAgent ko

Intents Entities Knowledge [beta] Fulfillment Integrations Training History Analytics Prebuilt Agents Docs Standard Free Upgrade Support

# Web Response 형식

```
JSON
  - id : "6d7e8bfa-a521-4ea5-8645-38f733b65b8e"
  - timestamp : "2018-12-15T05:05:59.014Z"
  - lang : "ko"
  - result
    - source : "agent"
    - resolvedQuery : "음식"
      -----
    - speech : ""
    - action : ""
    - actionIncomplete : false
  - parameters
    - msg : "test"
  - contexts
    - 0
      - name : "orderfood-followup"
      - parameters
        - msg : "test"
        - msg.original : ""
        - lifespan : 2
  - metadata
  - fulfillment
    - speech : "어떤 음식을 드릴까요?"
      -----
  - messages
    - 0
  - score : 0.7200000286102295
  - status
    - sessionId : "9eb90172-babc-fae3-6f3a-0ca0876a82ef"
```

Query

Answer

# Dialogflow API 호출

```
import requests
import json
import sys

text = sys.argv[1]

sessionId = "user01"

data_send = {
    'query': text,      'sessionId': sessionId,
    'lang': 'ko' ,      'timezone' : 'Asia/Seoul'
}

data_header = {
    'Authorization': 'Bearer a5eb16526d534a9fba2210249c580de3',
    'Content-Type': 'application/json; charset=utf-8'
}

dialogflow_url = 'https://api.dialogflow.com/v1/query?v=20150910'
res = requests.post(dialogflow_url, data=json.dumps(data_send), headers=data_header)

print(res.status_code) # 정상이면 requests.codes.ok == 200

dict = res.json()

answer = dict['result']['fulfillment']['speech']
print(answer)
```

```
python openapi.py 안녕
200
안녕하세요!
```

# 콘솔 기반 챗봇 구현

```
import requests
import json

def get_answer(text, sessionId):
    data_send = {
        'query': text, 'sessionId': sessionId,
        'lang': 'ko', 'timezone' : 'Asia/Seoul'
    }
    data_header = {
        'Authorization': 'Bearer a5eb16526d534a9fba2210249c580de3',
        'Content-Type': 'application/json; charset=utf-8'
    }

    dialogflow_url = 'https://api.dialogflow.com/v1/query?v=20150910'
    res = requests.post(dialogflow_url, data=json.dumps(data_send), headers=data_header)
    if res.status_code == requests.codes.ok:
        return res.json()
    return {}

while True :
    txt = input("->")
    dict = get_answer(txt, 'user01')
    answer = dict['result']['fulfillment']['speech']
    print("Bot:" + answer)

    python chatbot.py
->안녕
Bot:안녕하세요!
->머해?
Bot:제가 제대로 이해하고 있는지 잘 모르겠어요.
```

| Chapter |

# 05

## 웹 데이터 분석

---





- 구조화된 문서를 효율적으로 파싱
- 필요한 정보를 손쉽게 접근 가능
- Html 문서와 xml 문서 파싱
- Unicode 텍스트 처리
- 설치
  - pip install beautifulsoup4

# selector를 이용한 html 분석

```
|: import requests
|: from bs4 import BeautifulSoup
|: import re

|:
|: html = """
|:
|:     <html>
|:         <head>
|:             <title>    Title~~~~</title>
|:         </head>
|:
|:         <div class="txt1"> hello
|:             <p class="txt1"> hello2 </p>
|:         </div>
|:         <p class="txt1"> hello3 </p>
|:
|:         <div>
|:             <img src=test.jpg id=img />
|:         </div>
|:
|:     </html>
|: """
|:

bs = BeautifulSoup(html, "html.parser")
```

# selector를 이용한 html 분석 : tag selector

```
In [47]: title = bs.select('title')

print(type(title))
print(title)
print(title[0].text)

<class 'list'>
[<title> Title~~~~ </title>]
    Title~~~~
```

```
In [29]: title = bs.select('html head title')
print(title)

title = bs.select('html title')
print(title)
```

```
[<title> Title~~~~ </title>]
[<title> Title~~~~ </title>]
```

```
In [30]: title = bs.select('html > title')
print(title)

title = bs.select('head > title')
print(title)
```

```
[]
[<title> Title~~~~ </title>]
```

# selector를 이용한 html 분석 : 클래스 및 ID selector

```
In [33]: div = bs.select('.txt1')  
  
print(len(div))  
print(div)
```

```
3  
[<div class="txt1"> hello  
     <p class="txt1"> hello2 </p>  
</div>, <p class="txt1"> hello2 </p>, <p class="txt1"> hello3 </p>]
```

```
In [41]: p = bs.select('div > p.txt1')  
print(p)
```

```
[<p class="txt1"> hello2 </p>]
```

```
In [49]: img = bs.select('#img')  
print (img)
```

```
[]
```

```
In [58]: img = bs.select('div > img#img')  
print (img)
```

```
[]
```

# 날씨 정보 추출



날씨정보



```
▼<div class="main_info">
  <span class="ico_state ws2"></span>
  ▼<div class="info_data">
    ▼<p class="info_temperature"> == $0
      <span class="todaytemp">23</span>
    ▼<span class="tempmark">
      <span class="blind">도씨</span>
      "°C"
    </span>
  </p>
```

```
▼<li>
  <p class="cast_txt">맑음, 어제보다 1° 낮아요</p> == $0
</li>
```

```
▼<dd class="lv1"> == $0
  <span class="num">23㎍/㎥</span>
  "좋음"
  <span class="ico"></span>
</dd>
```

```
▼<dd class="lv1"> == $0
  <span class="num">12㎍/㎥</span>
  "좋음"
  <span class="ico"></span>
</dd>
```

# 날씨 정보 추출



```
url = "https://search.naver.com/search.naver?sm=top_hty&fbm=1&ie=utf8&query="
url = url + urllib.parse.quote_plus(city + "날씨")

soup = BeautifulSoup(urllib.request.urlopen(url).read(), "html.parser")

temp = soup.select("span.todaytemp")
desc = soup.select("p.cast_txt")
dust = soup.select("dd.lv1")

dict = {"temp": temp[0].text, "desc": desc[0].text, "dust": dust[0].text }
```

# 네이버 사전 정보 추출

[https://search.naver.com/search.naver?sm=tab\\_hty.top&where=kdic&query=%EC%9D%BC%EA%B3%B5%EC%A7%80%EB%8A%A5](https://search.naver.com/search.naver?sm=tab_hty.top&where=kdic&query=%EC%9D%BC%EA%B3%B5%EC%A7%80%EB%8A%A5)

The screenshot shows a browser window with developer tools open, specifically the Elements tab. The page content is from a search result for '인공지능' on Naver. The selected element is a `<div class="kindic">`. The expanded tree shows nested elements: `<div class="info_area">`, `<div class="tit_box">`, and `<p class="txt_box">`. The `<p>` block contains the definition of AI and its sub-sections.

Page content (Visible in the browser):

NAVER 인공지능

통합검색 이미지 뉴스 어학사

지식백과 1-10 / 1,911건

인공지능 인간의 학습능력과 추론능력, 지각능력, 자연언어의 이해능력 등을 컴퓨터 프로그램으로 실현한 기술. 인간의 지능으로 할 수 있는 사고, 학습, 자기 개발 등을 컴퓨터가 할 수 있도록 하는 방법을 연구하는...

인공지능 [artificial intelligence, 人工知能].

인간의 학습능력과 추론능력, 지각능력, 자연언어의 이해능력 등을 컴퓨터 프로그램으로 실현한 기술. 인간의 지능으로 할 수 있는 사고, 학습, 자기 개발 등을 컴퓨터가 할 수 있도록 하는 방법을 연구하는...

두산백과 | 소프트웨어 > 인공지능

# 위키피디아 정보 추출

<https://ko.wikipedia.org/wiki/채터봇>

The screenshot shows a web browser displaying the Korean Wikipedia page for "채터봇" (Chatbot). The page content is annotated with red text, likely representing extracted information or entities. The annotations include:

- 챗봇** (chatbot) 혹은 **채터봇** (Chatterbot)은 **음성**이나 **문자**를 통한 **인간**과의 대화를 통해 특정한 작업을 수행하도록 제작된 **컴퓨터 프로그램**이다.
- 토크봇** (talkbot), **채터박스** (chatterbox) 혹은 그냥 **봇** (bot)라고도 한다.
- 인간이 사용하는 언어를 이해해서 대화를 진행하는 **자연 언어 처리** 기술을 심각하게 적용하는 언어미해 방식, 입력받은 말에서 특정 단어나 어구를 검출하여 그에 맞는 준비된 응답을 출력하는 검색 방식,

The browser interface includes a back button, forward button, refresh button, and a search bar with the URL https://ko.wikipedia.org/wiki/채터봇. The page has tabs for 문서 (Document) and 토론 (Discussion).

# 최신 영화 정보 추출

<https://movie.naver.com/movie/running/current.nhn>

The screenshot shows the Naver Movie website's interface. On the left, there's a sidebar with links like 'Movie Home', 'Upcoming &预定' (Upcoming &预定), 'Recent Movies', 'New Releases', 'Movie Ranking', 'Reviews', 'Download', and 'Indie Cinema'. The main content area is titled 'Current Running Movies' and displays three movie entries:

- 장사리 : 잊혀진 영웅들**  
네이버 ★★★★☆ 8.48 | 참여 3,402명 | 예매율 26.74%  
개요 전쟁 | 104분 | 2019.09.25 개봉  
감독 팔경택, 김태훈  
출연 김명민, 최민호, 김성철, 김인권, 곽시양  
Buttons: 예매하기, 포토보기, 예고편, 메이킹
- 나쁜 녀석들: 더 무비**  
네이버 ★★★★☆ 6.60 | 참여 8,761명 | 예매율 24.39%  
개요 범죄, 액션 | 114분 | 2019.09.11 개봉  
감독 손용호  
출연 마동석, 김상중, 김아중, 장기용  
Buttons: 예매하기, 포토보기, 예고편, 메이킹, 인터뷰
- 양자물리학**  
네이버 ★★★★☆ 8.61 | 참여 2,433명 | 예매율 14.16%  
개요 범죄 | 119분 | 2019.09.25 개봉  
감독 이성태  
Buttons: 예매하기, 포토보기, 예고편, 메이킹

```
<ul class="lst_detail_t1">
  <li>
    <div class="thumb">...</div>
    <dl class="lst_dsc">
      <dt class="tit">
        <span class="ico_rating_12">12세 관람가</span>
        <a href="/movie/bi/mi/basic.nhn?code=159806">장사리 : 잊혀진 영웅들</a>
        <!-- N=a:nol.title,r:1,i:159806 -->
      </dt>
      <dd class="star">...</dd>
      <dd>...</dd>
      <dd class="info_t1">...</dd>
    </dl>
  </li>
```

# 최신 음악 차트 정보 추출

```
▼<div class="music_info"> == $0
  ▼<div class="title">
    <a href="https://vibe.naver.com/track/28487815" target=
      "_blank" onclick="return goOtherCR(this,
      'a=mus_sen*s.tit&r=1&i=080003B5_000028487815&u='+urlencode(
      this.href));">어떻게 이별까지 사랑하겠어, 널 사랑하는 거지</a>
  </div>
  ▼<div class="info">
    <a href="https://vibe.naver.com/artist/243100" onclick="return
      goOtherCR(this,
      'a=mus_sen*s.artist&r=1&i=080003B5_000028487815&u='+urlencode(
      this.href));" target="_blank" class="singer">AKMU (악동뮤지션)
    </a>
    <span class="bar"></span>
    <a href="https://vibe.naver.com/album/3199865" onclick="return
      goOtherCR(this,
      'a=mus_sen*s.album&r=1&i=080003B5_000028487815&u='+urlencode(
      this.href));" target="_blank" class="album">황해</a>
  </div>
</div>
```

뮤직 ⓘ

종합 TOP 100

순위	곡명	아티스트
1 - 0	 어떻게 이별까지	AKMU (악동뮤지션)
2 - 0	 흔들리는 꽃들 속	장범준   멜토
3 - 0	 워커힐릭	볼빨간사춘기   Two Five
4 - 0	 안녕	풀킴   호텔 델루나 OST Part 10

 뮤비     가사

가사

# 로또 번호 추출

[https://search.naver.com/search.naver?sm=top\\_hty&fbm=0&ie=utf8&query=%EB%A1%9C%EB%98%90%EB%B3%B5%EA%B6%8C](https://search.naver.com/search.naver?sm=top_hty&fbm=0&ie=utf8&query=%EB%A1%9C%EB%98%90%EB%B3%B5%EA%B6%8C)

로또6/45 당첨번호 정보



라인 설치하고 매주 당첨번호를 알림받



878회차 당첨번호 2019.09.28 ▾

2

(i) 본 정보에는 오류

```
<div class="num_box" style="border: 1px solid #ccc; padding: 10px; border-radius: 5px; background-color: #f9f9f9; margin-bottom: 10px;>
    <span class="num ball2">2</span>
    <span class="num ball6">6</span>
    <span class="num ball11">11</span>
    <span class="num ball16">16</span>
    <span class="num ball25">25</span>
    <span class="num ball31">31</span>
    <span class="bonus">보너스번호</span>
    <span class="num ball3">3</span>
    <a href="#" onclick="return goOtherCR(this,
        'a=nco_x5e*1.contents&r=1&i=0011AD9E_0000009C2012&u=' +
        urlencode(this.href));" target="_blank" class="btn_num" href="https://www.dhlottery.co.kr/gameResult.do?method=myWin">내 번호 당첨조회</a>
</div>
```

| Chapter |

# 06

## 자연어 처리

---

# 자연어처리 (NLP)

컴퓨터과학, 인공지능과 언어학이 합쳐진 분야

컴퓨터 과학

인공지능

언어학

자연어처리



## 자연어 처리 업무

수|움

중간

## 어려움

- 스펠링 체크
  - 키워드 검사
  - 유사어 감지
  - 웹사이트 및 서류의 형태 해석
  - 구문해석 etc.
  - 기계번역
  - 감정분석
  - 질의응답시스템

# 기계번역

문구기반 번역의 예



단어의 모호성

번역

한국어 일본어 영어 번역 갑자 ▾

The curry is hot.

한국어 영어 번역하기

카레는 뜨겁다.

수정 제안하기

즉석 번역 사용 안함 ★

# 감정분석

문장으로부터 감정을 판단

긍정적 「매우 재밌다. 아무리 놀아도 실증나지 않는다.」

→ **0.86**

부정적 「설치하지마. 데이터만 낭비한다.」

→ **-0.68**



# 질문응답시스템(QA 시스템)

closed-domain – 정해진 분야의 질문에 응답

「라마는 무슨 과?」 → 「낙타과」

open-domain – 어떠한 질문에도 응답

「왜 나는 결혼을 못하나?」 → 「...」

# 자연어 처리의 어려움

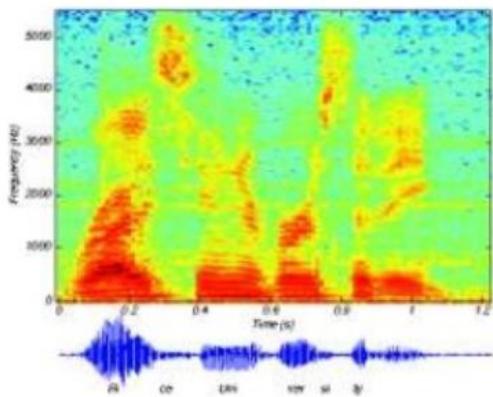
언어 · 상황 · 환경 · 지각 지식의 학습 및 표현의 복잡함  
→ Rule 기반만으로는 무리인가 ?

DNN은 분산표현의 장점으로 인해  
모호하지만 풍부한 정보를 얻을 수 있다.  
→ 단어의 벡터화로부터 시작

# 언어 데이터의 특성

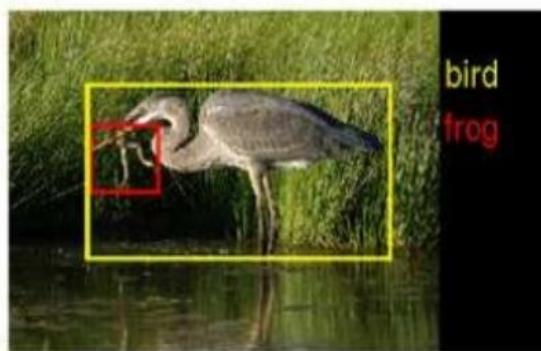


Audio



연속

Image



연속

Text

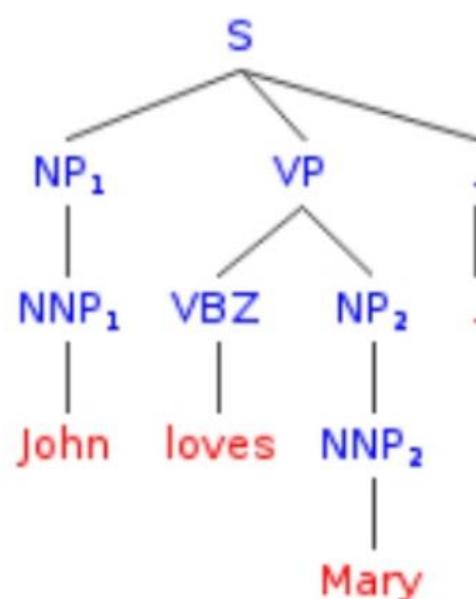


불연속

# 언어 데이터의 특성

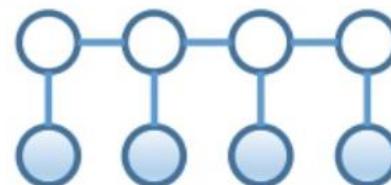
- 불연속적인 심볼의 계열(Sequence)
- 계열 길이가 샘플에 따라 다름
- 계층적, 재귀적(Recursive)인 구조를 지님  
(문자, 단어, 구, 문)

John loves Mary .



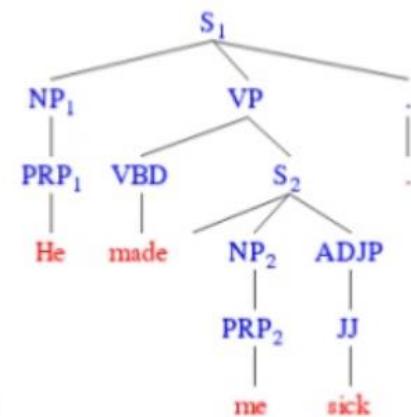
1. 계열 → 계열  $f_{\theta}(\text{seq} | \text{seq})$

- 형태소해석
- 기계번역, 자동요약
- 질의응답, chatbot



2. 계열 → Tree구조  $f_{\theta}(\text{tree} | \text{seq})$

- 구문해석



3. 계열 → 그래프구조  $f_{\theta}(\text{graph} | \text{seq})$

- 의미해석

# 단어의 국소표현

고양이 [1, 0, 0, 0, 0]

개 [0, 1, 0, 0, 0]

사람 [0, 0, 1, 0, 0]

:

:

이것만 가지고는 단어의 의미를 전혀 알 수 없다.

→ 단어의 **의미를 파악하는 벡터**를 갖고 싶다.



# 분포가설 [Harris 1954, Firth 1957]

*“You shall know a word by the company it keeps”*  
- J. R. Firth

비슷한 문맥을 가진 단어는 비슷한 의미를 갖는다.

현대의 통계적 자연어 처리에서 획기적인 발상

# Count-based vs Predictive methods

분포가설에 기반한 방법은 크게 2종류로 나눈다.

- count-based methods
  - 예 : SVD (LSA)、HAL、etc.
  - 단어, 문맥 출현횟수를 세는 방법
- predictive methods
  - 예 : NPLM、word2vec、etc.
  - 단어에서 문맥 또는 문맥에서 단어를 예측하는 방법



- Intent

Q : 안녕~

A : 안녕하세요

```
If ['안녕' , '하이'] in Q then  
    print A
```

- 0 :가위, 1:바위, 2:보

- If my == 0 and computer == 0 then 비김
- If my == 0 and computer == 1 then 짐
- If my == 0 and computer == 2 then 이김
- If my == 1 and computer == 0 then 이김
- If my == 1 and computer == 1 then 비김
- If my == 1 and computer == 2 then 짐
- If my == 2 and computer == 0 then 짐
- If my == 2 and computer == 1 then 이김
- If my == 2 and computer == 2 then 비김

# 공식을 통한 학습

- 규칙 기반 모델링
  - 0 :가위, 1:바위, 2:보
  - $(\text{my} - \text{computer} + 3) \% 3$ 
    - 0 : 비김
    - 1 : 이김
    - 2 : 짐

		computer		
		0:가위	1:바위	2:보
my	0:가위	비김 $0-0=0$	짐 $0-1+3=2$	이김 $0-2+3=1$
	1:바위	이김 $1-0+3=4(1)$	비김 $1-1=0$	짐 $1-2+3=2$
	2:보	짐 $2-0+3=5(2)$	이김 $2-1+3=4(1)$	비김 $2-2=0$

| Chapter |

07



## 서버 프로그래밍

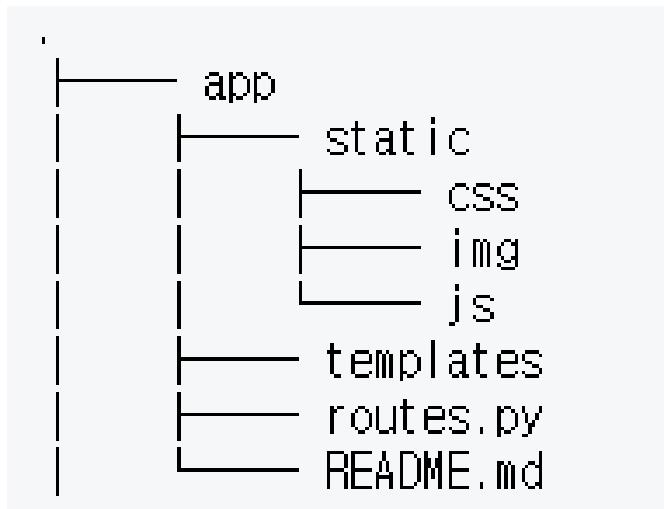
---

- 파이썬으로 작성된 마이크로 웹 프레임워크
- 최소한의 기능만 제공하여 유연한 어플리케이션 제작 가능
- 설치 : pip install Flask



# Flask 폴더 기본 구조

- 폴더 기본 구조
  - 루트 폴더 : app
  - 정적파일 폴더 : app/static
    - 하위에 css, img, js 폴더 생성
  - 템플릿 폴더 : app/templates
- python 코드는 루트 폴더에 작성



# Hello Flask

- 실행 : python home.py

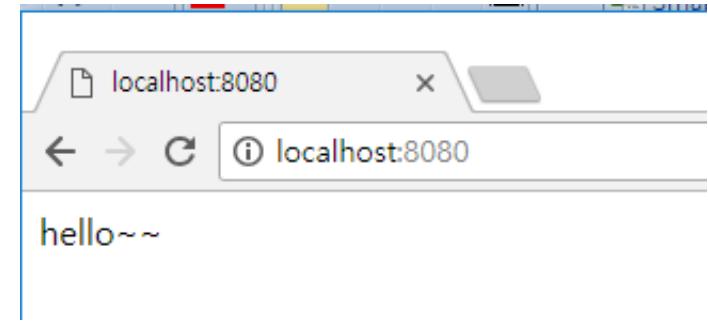
```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "hello~~"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

```
(base) D:\Wckt\2018\WT\IoT\flask>python home.py
* Restarting with stat
* Debugger is active!
* Debugger PIN: 383-366-169
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
127.0.0.1 - - [14/Jun/2018 14:21:24] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Jun/2018 14:21:24] "GET /favicon.ico HTTP/1.1" 404 -
* Detected change in 'D:\Wckt\2018\WT\IoT\flask\home.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 383-366-169
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```



- 코드 수정 시 서버 자동 reload

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)
```

■ Anaconda Prompt - python home.py

```
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
127.0.0.1 - - [14/Jun/2018 16:56:27] "POST /form HTTP/1.1" 200 -
* Detected change in 'D:\www\ct\2018\www\IoT\flask\home.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 383-366-169
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
127.0.0.1 - - [14/Jun/2018 19:11:36] "POST /form HTTP/1.1" 200 -
* Detected change in 'D:\www\ct\2018\www\IoT\flask\home.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 383-366-169
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
* Detected change in 'D:\www\ct\2018\www\IoT\flask\home.py', reloading
* Restarting with stat
* Debugger is active!
```

# Flask App path

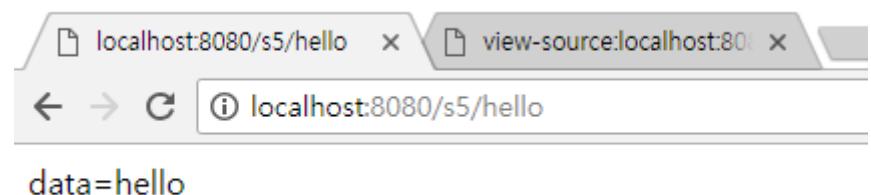
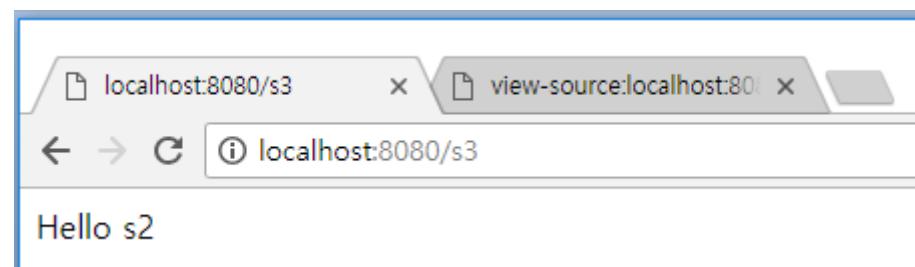
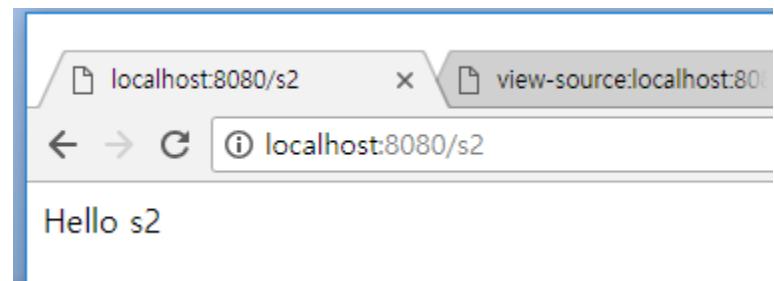
- 하나의 파일에서 다수의 URL 처리

```
@app.route('/')
def home():
    return render_template('home.html')

@app.route('/s1')
def s1():
    return "Hello s1"

@app.route('/s2')
@app.route('/s3')
def s2():
    return "Hello s2"

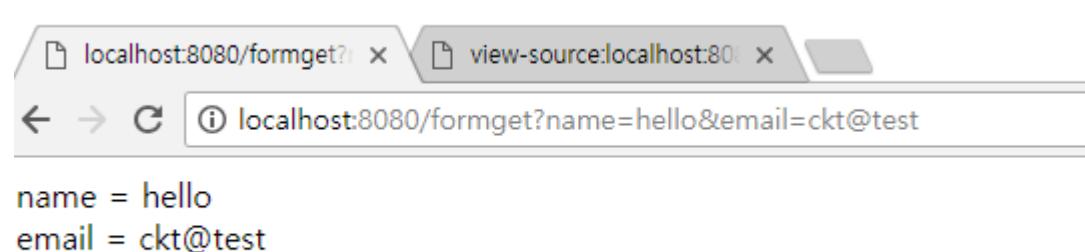
@app.route('/s5/<name>')
def s6(name):
    return "data=" + name
```



# Flask Form GET

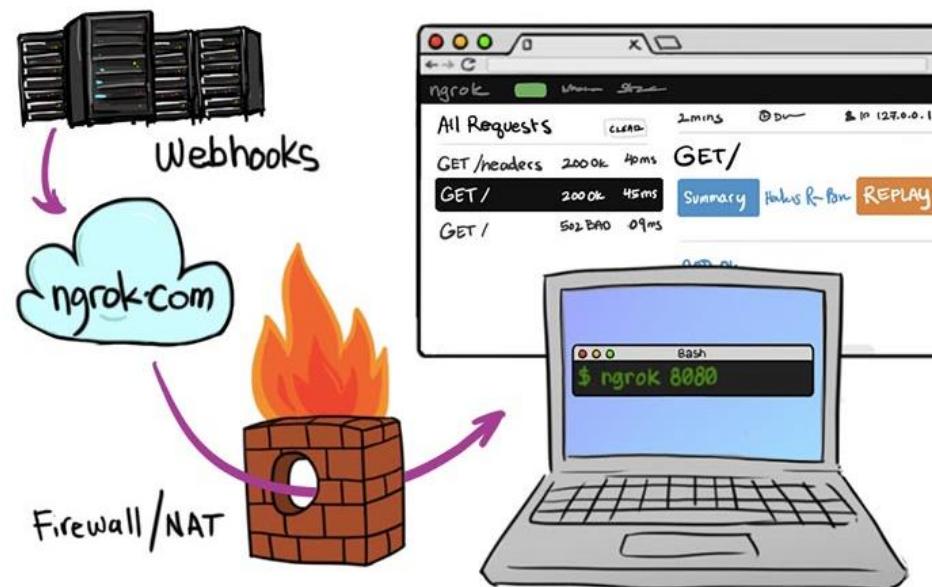
- Get 파라미터는 request.args.get() 함수로 값 접근

```
@app.route('/formget', methods=['GET'])  
def formget():  
  
    html = "name = " + request.args.get("name") + "<br>"  
    html += "email = " + request.args.get("email") + "<br>"  
    return html
```



# Secure Tunnels(NGROK)

- 공인 IP 아니여도 외부에서 접속 가능하게 하는 터널 프로그램
- 방화벽 내부 서버를 외부에서 접속 가능
- Mac, Linux, Windows, 모바일 지원
- 유료와 무료 버전 존재



<https://ngrok.com/>

# NGROK 사용방법



- 다운로드 : <https://ngrok.com/download>
- 실행 : ngrok http 8080

```
명령 프롬프트 - ngrok http 8080
ngrok by @inconshreveable

Session Status           online
Session Expires          7 hours, 59 minutes
Version                  2.2.8
Region                   United States (us)
Web Interface            http://127.0.0.1:4040
Forwarding               http://2107b72b.ngrok.io -> localhost:8080
Forwarding               https://2107b72b.ngrok.io -> localhost:8080

Connections              ttl     opn     rt1     rt5     p50     p90
                         3       0      0.04    0.01   0.31   0.46

HTTP Requests
-----
GET /book/java           200 OK
GET /favicon.ico         404 NOT FOUND
GET /                     404 NOT FOUND
```

| Chapter |

# 08



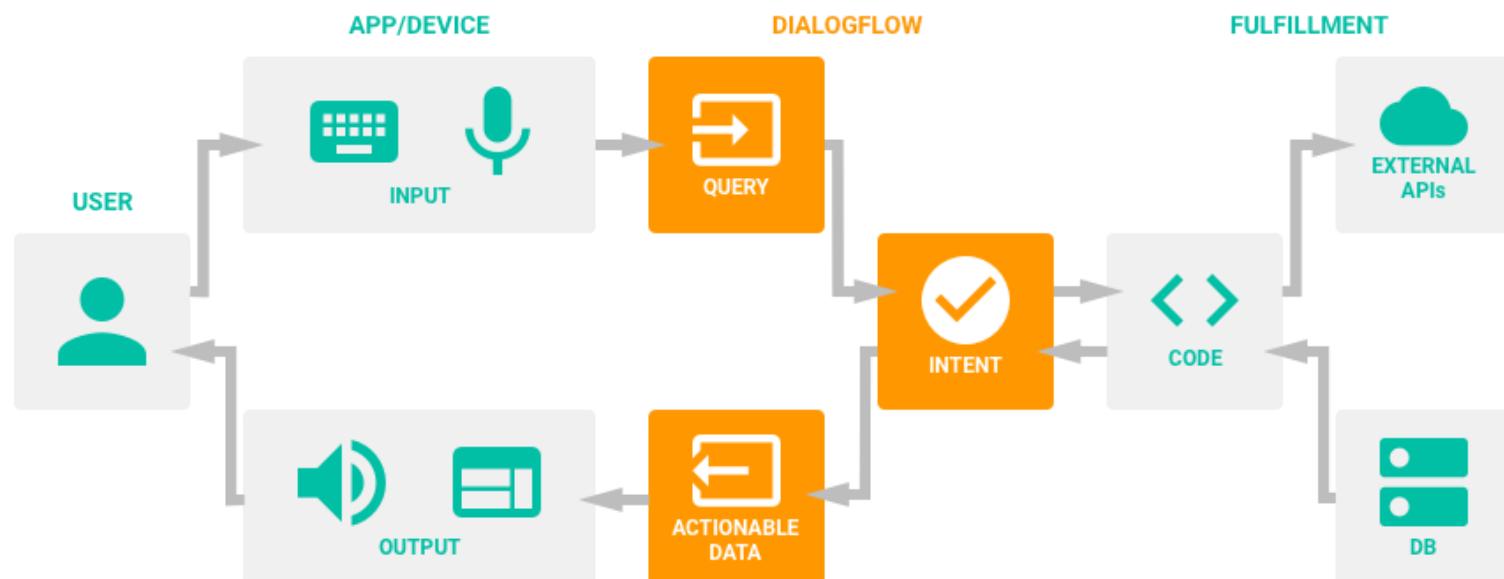
## WebHook 기반 챗봇 확장

---

# Fulfillment 개요

Dialogflow 에이전트가 의도에 따라 비즈니스 로직을 호출

Dialogflow의 자연어 처리로 추출한 정보를 사용하여 동적 응답 생성



# Webhook request

- Fulfillment를 사용하기 위해 webhook 설정 필요
- Webhook을 만들고 호스트하는 웹 서버 위치 지정
- Dialogflow는 일치 된 intent 확인
- intent 정보가 들어있는 JSON 객체를 WebHook에 HTTP POST 요청
- Webhook은 Dialogflow에 응답 JSON 객체 리턴

# Request 형식

```
V2 API V1 API

POST https://my-service.com/action

Headers:
//user defined headers
Content-type: application/json

POST body:

{
  "originalRequest": {},
  "id": "7811ac58-5bd5-4e44-8d06-6cd8c67f5406",
  "sessionId": "1515191296300",
  "timestamp": "2018-01-05T22:35:05.903Z",
  "timezone": "",
  "lang": "en-us",
  "result": {
    "source": "agent",
    "resolvedQuery": "user's original query to your agent",
    "speech": "Text defined in Dialogflow's console for the intent that was matched",
    "action": "Matched Dialogflow intent action name",
    "actionIncomplete": false,
    "parameters": {
      "param": "param value"
    },
    "contexts": [
      {
        "name": "incoming context name",
        "parameters": {},
        "lifespan": 0
      }
    ],
    "metadata": {
      "intentionId": "29bcd7f8-f717-4261-a8fd-2d3e451b8af8",
      "webhookUsed": "true",
      "webhookForSlotFillingUsed": "false",
      "nluResponseTime": 6,
      "intentName": "Name of Matched Dialogflow Intent"
    },
    "fulfillment": {
      "speech": "Text defined in Dialogflow's console for the intent that was matched",
      "messages": [
        {
          "type": 0,
          "speech": "Text defined in Dialogflow's console for the intent that was matched"
        }
      ],
      "score": 1
    }
  }
}
```

# Webhook response



- 웹 혹은 응답을 제공해야 함
- Dialogflow가 인식할 수 있는 json 포맷
- 응답은 Google 애플리케이션에서 작업 내에서 10 초 이내
- 다른 모든 애플리케이션에서는 5 초 이내
- 응답 크기가 64KB 이하

# Response 형식

```
V2 API V1 API

{
  "speech": "this text is spoken out loud if the platform supports voice interactions",
  "displayText": "this text is displayed visually",
  "messages": [
    {
      "type": 1,
      "title": "card title",
      "subtitle": "card text",
      "imageUrl": "https://assistant.google.com/static/images/molecule/Molecule-Formation-stop.png"
    },
    "data": {
      "google": {
        "expectUserResponse": true,
        "richResponse": {
          "items": [
            {
              "simpleResponse": {
                "textToSpeech": "this is a simple response"
              }
            }
          ]
        }
      },
      "facebook": {
        "text": "Hello, Facebook!"
      },
      "slack": {
        "text": "This is a text response for Slack."
      }
    },
    "contextOut": [
      {
        "name": "context name",
        "lifespan": 5,
        "parameters": {
          "param": "param value"
        }
      }
    ],
    "source": "example.com",
    "followupEvent": {
      "name": "event name",
      "parameters": {
        "param": "param value"
      }
    }
  }
}
```

# Agent 설정

Dialogflow

TestAgent

ko

Intents Entities Knowledge [beta] Fulfillment Integrations Training History Analytics Prebuilt Agents Docs Standard Free Upgrade Support

TestAgent

General Languages ML Settings Export and Import Speech Share

DEFAULT TIME ZONE  
(GMT+9:00) Asia/Tokyo  
Date and time requests are resolved using this timezone.

GOOGLE PROJECT  
Project ID testagent-cf7f4

API VERSION  
 V2 API Use Cloud API as default for the agent. Your webhook will receive V2 format requests and should return V2 format responses.  
 V1 API Legacy APIs

API KEYS (V1)  
Client access token [REDACTED]  
Developer access token [REDACTED]

SAVE

# Fulfillment 외부 URL 연결



TestAgent

ko

Intents

Entities

Knowledge [beta]

**Fulfillment**

Integrations

Training

History

Webhook

ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL\*

BASIC AUTH

HEADERS

**Add header**

DOMAINS

 Dialogflow API V1 will be deprecated on October 23rd, 2019. Learn how to migrate to API V2 [here](#).

# Dialogflow

Multicam  

ko 

**Intents** 

Entities 

Knowledge [beta] 

Fulfillment 

Integrations 

Training 

History 

Analytics 

Prebuilt Agents 

Docs 

Standard Free 

## weather

### Contexts

### Events

### Training phrases

Search training phrases  

PARAMETER NAME	ENTITY	RESOLVED VALUE	X
geo-city	@sys.geo-city	인천	
인천 날씨			
인천 날씨 어때?			
인천의 오늘 날씨어때?			
인천의 오늘 날씨?			
다음주 부산 날씨는			
오늘 서울 날씨는?			



## ● weather

SAVE

⋮

” 서울의 내일 날씨는 어떤가요?

## Action and parameters



Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	geo-city	@sys.geo-city	\$geo-city	<input type="checkbox"/>	지역명을 말씀해주세요 [1]
<input type="checkbox"/>	date-time	@sys.date-time	\$date-time	<input type="checkbox"/>	—
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

[New parameter](#)



## Responses

DEFAULT

GOOGLE ASSISTANT

FACEBOOK MESSENGER



## Text response

- 1 \$geo-city날씨 좋습니다.
- 2 Sdate-time 의 \$geo-city의 날씨는 흐려요
- 3 날씨는 쌩쌩
- 4 Enter a text response variant

[ADD RESPONSES](#)

Set this intent as end of conversation



## Fulfillment



Enable webhook call for this intent



Enable webhook call for slot filling

# 외부 Web 서버 연동

## Webhook

ENABLED



Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the webhook requirements specific to the API version enabled in this agent.

URL\*

http://130572a8.ngrok.io

```
@app.route('/' , methods=['POST', 'GET'])
def chatting():

    req = request.get_json(force=True)

    print(json.dumps(req, indent=4))

    res = {"speech" : "좋아~~~~~"}

    return jsonify(res)
```

# Request JSON 객체

```
{"id": "eefb5835-b715-4706-bb4c-e495411fee31", "timestamp": "2018-12-15T04:54:05.297Z", "lang": "ko", "result": {"source": "agent", "resolvedQuery": "음식", "speech": "", "action": "", "actionIncomplete": False, "parameters": {"msg": "test"}, "contexts": [{"name": "orderfood-followup", "parameters": {"msg": "test", "msg.original": ""}, "lifespan": 2}], "metadata": {"intentId": "14eefe7e-4706-4d19-9199-d54f61d78899", "webhookUsed": "true", "webhookForSlotFillingUsed": "false", "isFallbackIntent": "false", "intentName": "orderfood"}, "fulfillment": {"speech": "어떤 음식을 드릴까요?", "messages": [{"type": 0, "speech": "어떤 음식을 드릴까요?"}]}, "score": 0.7200000286102295}, "status": {"code": 200, "errorType": "success"}, "sessionId": "9eb90172-babc-fae3-6f3a-0ca0876a82ef"}
```

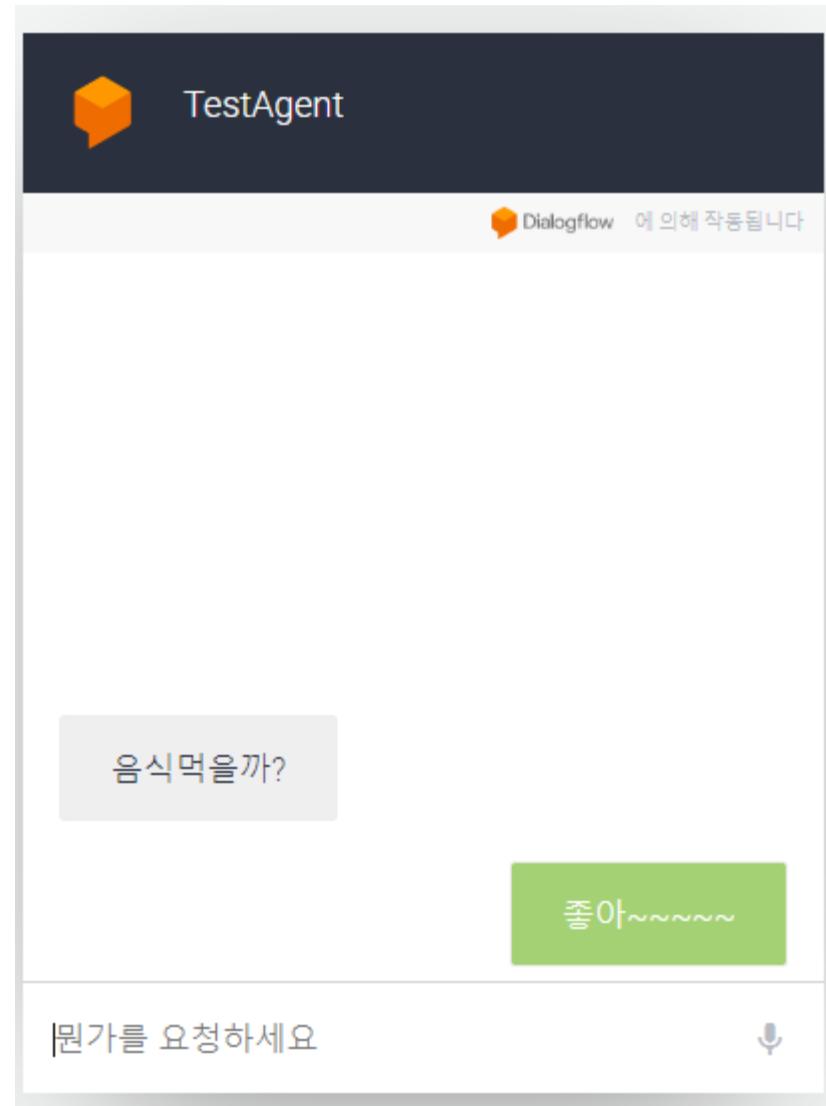
# Request JSON 객체

```
running on http://0.0.0.0:5000/ (press CTRL+C to quit)
{
  "id": "19c279a0-eaa6-487a-ac31-9b7957c4e63d",
  "timestamp": "2018-12-15T05:03:01.368Z",
  "lang": "ko",
  "result": {
    "source": "agent",
    "resolvedQuery": "\udc74\uc2ec\uc73c",
    "speech": "",
    "action": "",
    "actionIncomplete": false,
    "parameters": {
      "msg": "test"
    },
    "contexts": [
      {
        "name": "orderfood-followup",
        "parameters": {
          "msg": "test",
          "msg.original": ""
        },
        "lifespan": 2
      }
    ],
    "metadata": {
      "intentId": "14eefe7e-4706-4d19-9199-d54f61d78899",
      "webhookUsed": "true",
      "webhookForSlotFillingUsed": "false",
      "isFallbackIntent": "false",
      "intentName": "orderfood"
    },
    "fulfillment": {
      "speech": "\udc5b\udb5a4 \udc74\uc2dd\uc744 \udb4dc\udb9b4\udae4c\uc694?",
      "messages": [
        {
          "type": 0,
          "speech": "\udc5b\udb5a4 \udc74\uc2dd\uc744 \udb4dc\udb9b4\udae4c\uc694?"
        }
      ],
      "score": 0.5099999904632568
    },
    "status": {
      "code": 200,
      "errorType": "success"
    },
    "sessionId": "9eb90172-babc-fae3-8f3a-0ca0876a82ef"
  }
}
```

# Request JSON 객체

```
JSON
  id : "6d7e8bfa-a521-4ea5-8645-38f733b65b8e"
  timestamp : "2018-12-15T05:05:59.014Z"
  lang : "ko"
  result
    source : "agent"
    resolvedQuery : "음식"
    speech : ""
    action : ""
    actionIncomplete : false
  parameters
    msg : "test"
  contexts
    0
      name : "orderfood-followup"
      parameters
        msg : "test"
        msg.original : ""
        lifespan : 2
    metadata
    fulfillment
      speech : "어떤 음식을 드릴까요?"
      messages
        0
        score : 0.7200000286102295
    status
    sessionId : "9eb90172-babc-fae3-6f3a-0ca0876a82ef"
```

# Fulfillment 테스트



# Action and Parameter 전달

## Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	msg	Enter entity	test	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	JSON		

+ New parameter

The JSON tree view shows the following structure:

- JSON
  - id : "6d7e8bfa-a521-4ea5-8645-38f733b65b8e"
  - timestamp : "2018-12-15T05:05:59.014Z"
  - lang : "ko"
- result
  - source : "agent"
  - resolvedQuery : "음식"
  - speech : ""
  - action : ""
  - actionIncomplete : false
- parameters
  - msg : "test"

```
print("→", req['result']['parameters']['msg'])
```

# Entity 전달



Dialogflow

TestAgent ko

Intents

Entities

food

Define synonyms ?  Allow automated expansion

짜장	짜장
짬뽕	짬뽕
Click here to edit entry	

# Entity 전달



Dialogflow

TestAgent ▼ ⚙️

ko +

💬 **Intents** +

- whatfood

Contexts ?

Events ?

Training phrases ? Search

” ” Add user expression

” ” 짜장?

” ” 짬뽕 먹을까?

” ” 짜장면 먹을까?

PARAMETER NAME	ENTITY	RESOLVED VALUE
food	@food	짜장면

# Entity 전달



## Action and parameters



Enter action name



REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
<input type="checkbox"/>	food	@food	\$food	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

[+ New parameter](#)

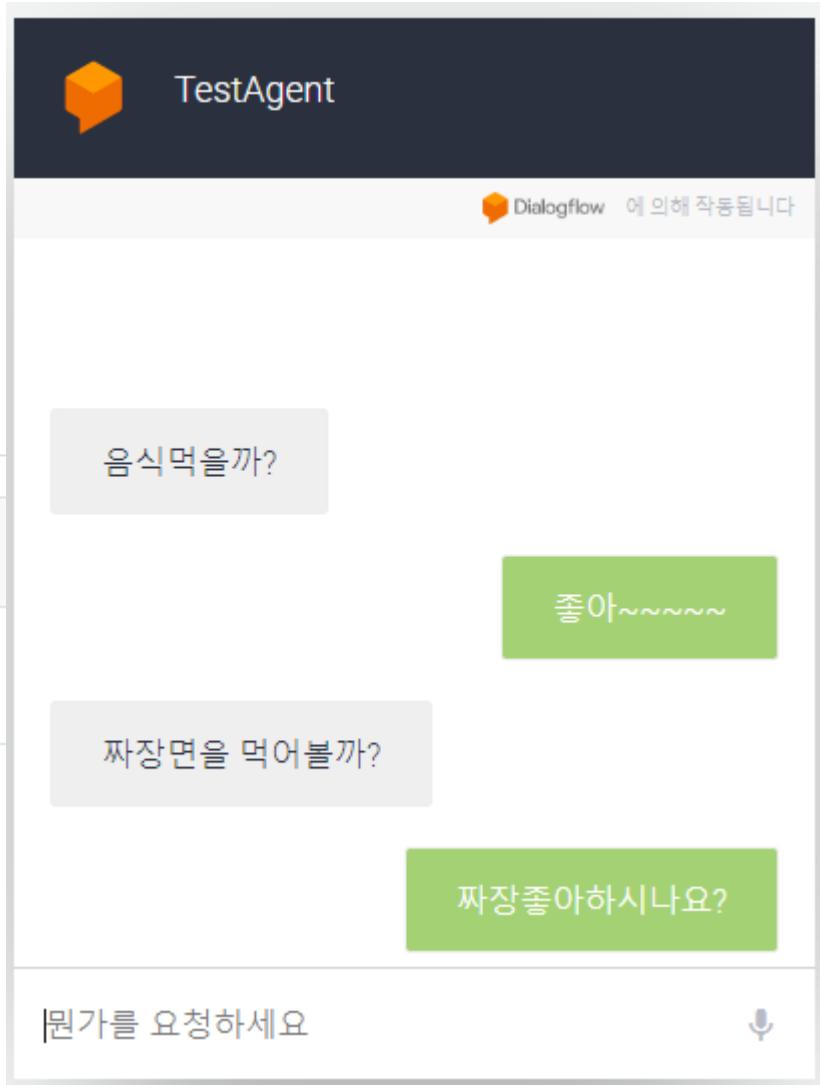
# Entity 정보를 포함한 Fulfillment 테스트

## Responses ?

DEFAULT +

### Text response

- 1 \$food 좋아하시나요?
- 2 Enter a text response variant



# Entity 정보를 포함한 Request

Viewer Text

JSON

- id : "c9b17ed5-92d8-47b4-8cae-3e35dd92c95e"
- timestamp : "2018-12-15T15:34:17.228Z"
- lang : "ko"

result

- source : "agent"
- resolvedQuery : "짜장"
- speech : ""
- action : ""
- actionIncomplete : false

parameters

- food : "짜장"

contexts

metadata

- intenId : "5ecddbff-8db9-440e-9273-bcc40f0923ad"
- webhookUsed : "true"
- webhookForSlotFillingUsed : "false"
- isFallbackIntent : "false"
- intentName : "whatfood"

fulfillment

- speech : "짜장 좋아하시나요?"

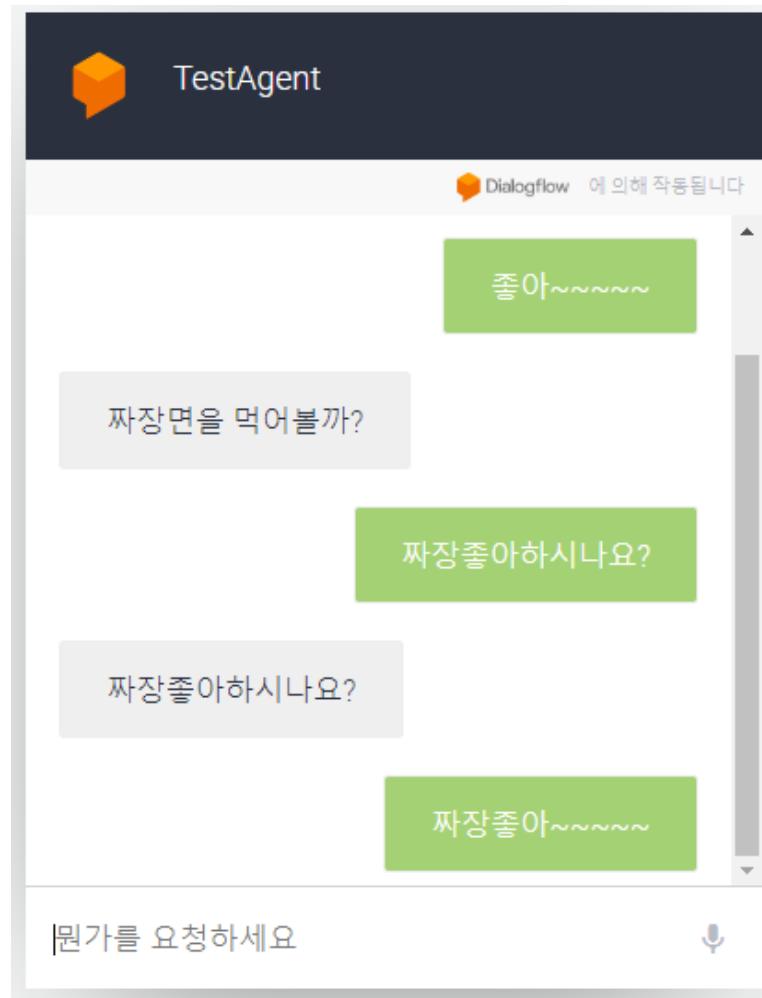
messages

- score : 1

status

- sessionId : "8a3a8b4f-5350-717c-7014-7910cf932ff1"

# Entity 정보를 포함한 Fulfillment 테스트

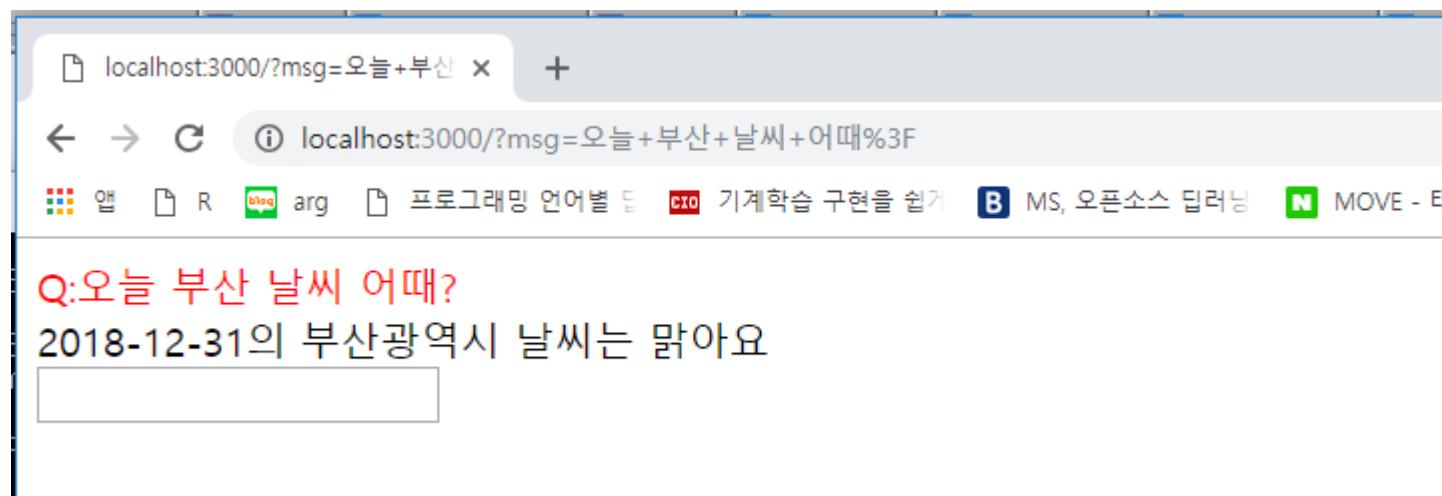
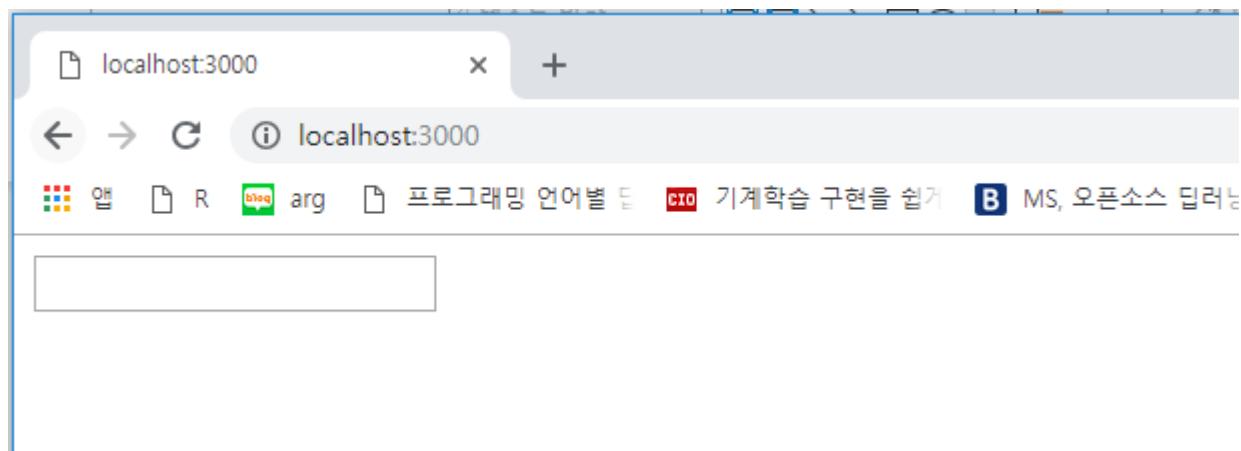


```
res = {"speech" : req['result']['parameters']['food'] + "좋아~~~~~"}  
return jsonify(res)
```

# 서버 프로그래밍과 Dialogflow 연동

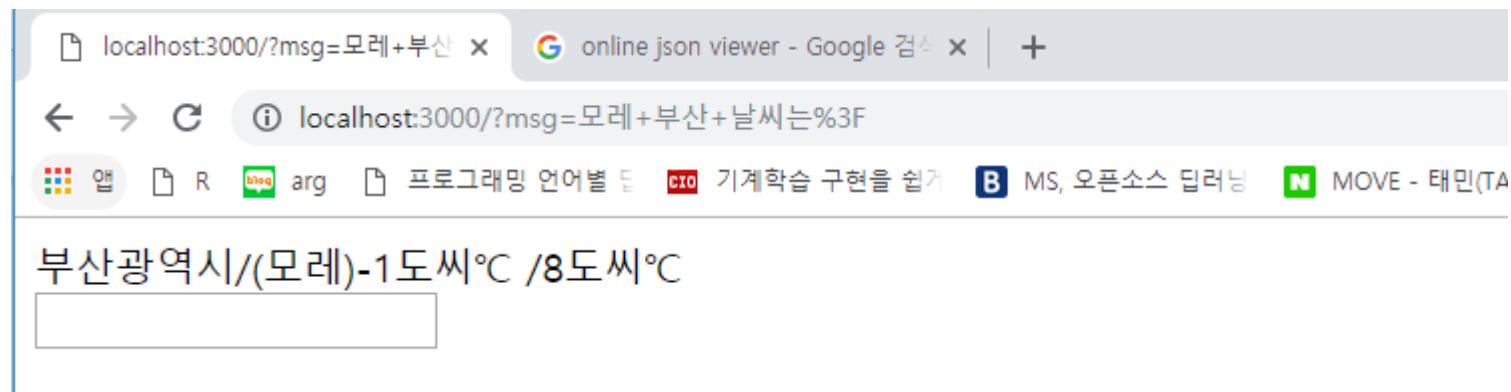
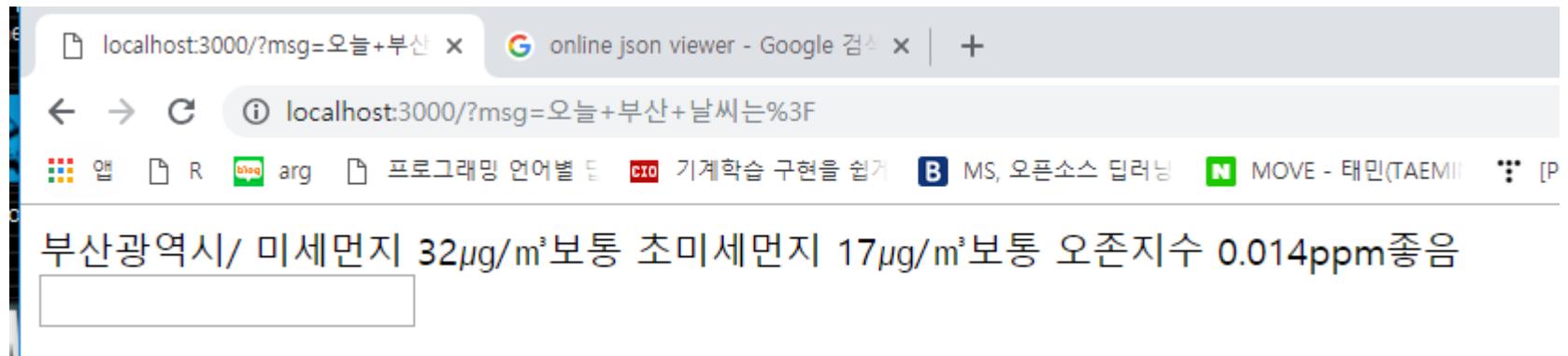


dialogflowapiform.py



# 웹 데이터 분석과 자연어 처리 연동

dialogflowapinlp.py



# 웹 데이터 분석과 자연어 처리 연동

dialogflowapinlp.py

