

CHATBOTHON

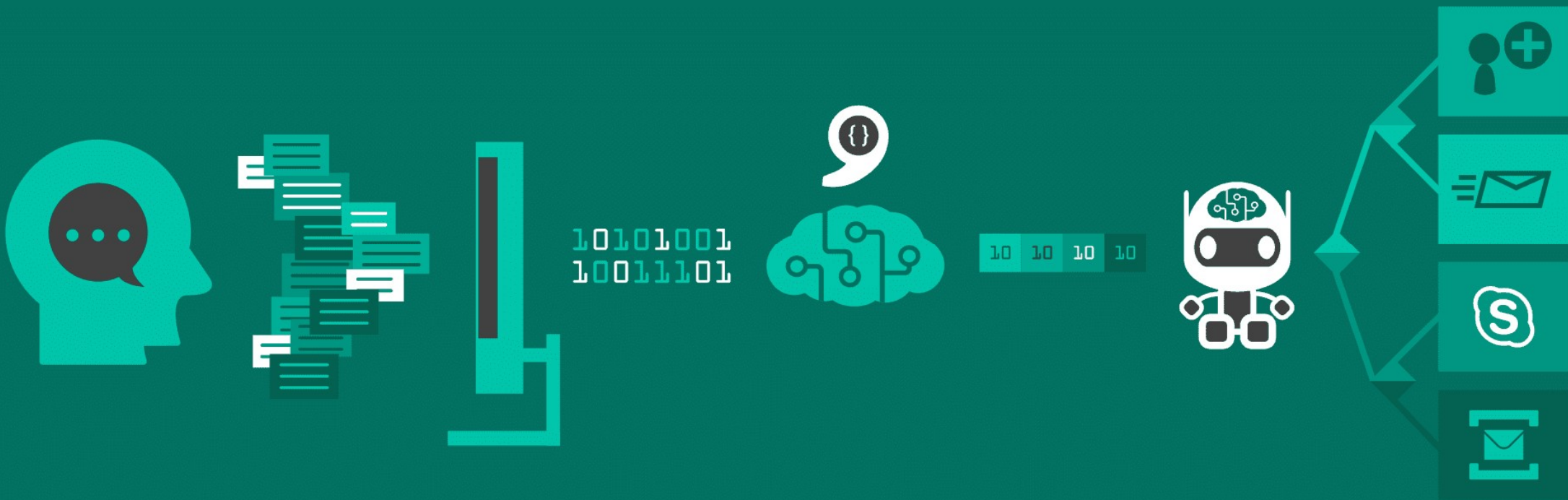
[행사개요](#) [아이디어](#) [시상금](#) [분야](#) [후원사](#)

# 2018 인공지능 챗봇 해커톤 AI CHATBOTHON 2018

개포디지털혁신파크, 2018.12.01(토)

주최: 한국인공지능연구소

[참가신청](#)





# 챗봇 해커톤 초급과정

이재석

주관 한국인공지능아카데미

후원 한국인공지능연구소

# 일정

# 오늘 일정

01 구름ide가입,설정	09:00 – 09:50(50분)
02 카카오톡친구플러스 설정	10:00 – 10:40(40분)
03 에코봇	11:00 – 11:20(20분)
04 번역봇	11:20 – 11:40(20분)
<b>05 버튼봇</b>	<b>12:00 – 12:30(30분)</b>
06 api봇	12:30 – 01:00(30분)

## 05 버튼 활용

# 5.1 버튼 활용: python 실행

- 에코봇에 버튼 달기



탐색기

열려 있는 편집기

kakao\_bot.py 03.bu...

KAKA...

00.hello-world

01.eco-bots

02.translate-bots

03.button-bots

kakao\_bot.py

04.finance-bots

05.cognitive-bots

06.dialogflow-bots

ngrok.exe

README.md



개요

kakao\_bot.py x

```
1  # -*- coding: utf-8 -*-
2
3  from flask import Flask
4  from flask import request
5  from flask import jsonify
6  from flask import json
7
8  app = Flask(__name__)
9
10 @app.route("/keyboard")
11 def keyboard():
12     response = {
13         "type" : "buttons",
14         "buttons" : ["처음으로", "챗봇등록", "취소하기"]
15     }
16     response = json.dumps(response, ensure_ascii=False)
17     return response
18
19
20 @app.route("/message", methods=["POST"])
21 def message():
22     data = json.loads(request.data)
23     content = data["content"]
```



탐색기

열려 있는 편집기

kakao\_bot.py 03.bu...

KAKA...

00.hello-world

01.eco-bots

02.translate-bots

03.button-bots

**kakao\_bot.py**

04.finance-bots

05.cognitive-bots

06.dialogflow-bots

ngrok.exe

README.md

측면에서 열기 Ctrl+Enter

탐색기에 표시 Alt+Shift+B

**명령 프롬프트에서 열기**

비교를 위해 선택

복사 Ctrl+C

경로 복사 Alt+Shift+C

이름 바꾸기 F2

삭제 Del

Debug All Unit Tests

Run All Unit Tests

Run Python File in Terminal

kakao\_bot.py x

```
1  # -*- coding: utf-8 -*-
2
3  from flask import Flask
4  from flask import request
5  from flask import jsonify
6  from flask import json
7
8  app = Flask(__name__)
9
10 @app.route("/keyboard")
11 def keyboard():
12     response = {
13         "type": "buttons",
14         "buttons": ["처음으로", "챗봇등록", "취소하기"]
15     }
16     response = json.dumps(response, ensure_ascii=False)
17     return response
18
19
20 @app.route("/message", methods=["POST"])
21 def message():
22     data = json.loads(request.data)
23     content = data["content"]
```



탐색기

열려 있는 편집기

kakao\_bot.py 03.bu..

KAKAO-BOT-MASTER

00.hello-world

01.eco-bots

02.translate-bots

03.button-bots

kakao\_bot.py

04.finance-bots

05.cognitive-bots

06.dialogflow-bots

ngrok.exe

README.md



kakao\_bot.py x

```
1  # -*- coding: utf-8 -*-
2
3  from flask import Flask
4  from flask import request
5  from flask import jsonify
6  from flask import json
7
8  app = Flask(__name__)
9
10 @app.route("/keyboard")
11 def keyboard():
12     response = {
```

문제 출력 디버그 콘솔 터미널

1: powershell

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\jaeseok\kakao-bot-master\03.button-bots&gt; python .\kakao\_bot.py



개요

10



탐색기

열려 있는 편집기

kakao\_bot.py 03.b...

KAKAO-BOT-MASTER

00.hello-world

01.eco-bots

02.translate-bots

03.button-bots

kakao\_bot.py

04.finance-bots

05.cognitive-bots

06.dialogflow-bots

ngrok.exe

README.md



개요

kakao\_bot.py x

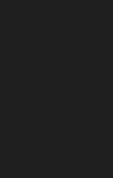
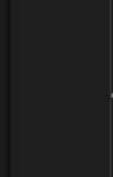
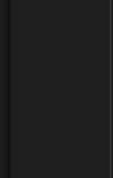
```
1  # -*- coding: utf-8 -*-
2
3  from flask import Flask
4  from flask import request
5  from flask import jsonify
6  from flask import json
7
8  app = Flask(__name__)
9
10 @app.route("/keyboard")
11 def keyboard():
12     response = {
```

문제 출력 디버그 콘솔 터미널

1: python

Copyright (C) Microsoft Corporation. All rights reserved.

```
PS C:\Users\jaeseok\kakao-bot-master\03.button-bots> python .\kakao_bot.py
* Serving Flask app "kakao_bot" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:80/ (Press CTRL+C to quit)
```



탐색기

- 열려 있는 편집기
  - kakao\_bot.py 03.b...
- KAKAO-BOT-MASTER
  - 00.hello-world
  - 01.eco-bots
  - 02.translate-bots
  - 03.button-bots
    - kakao\_bot.py
  - 04.finance-bots
  - 05.cognitive-bots
  - 06.dialogflow-bots
- ngrok.exe
- README.md

개요

```
1  # -*- coding: utf-8 -*-
2
3  from flask import Flask
4  from flask import request
5  from flask import jsonify
6  from flask import json
7
8  app = Flask(__name__)
9
10 @app.route("/keyboard")
11 def keyboard():
12     response = {
```

문제 출력 디버그 콘솔 터미널

2: ngrok

(Ctrl+C to quit)

ngrok by @inconshreveable

Session Status	online
Session Expires	7 hours, 59 minutes
Version	2.2.8
Region	Asia Pacific (ap)
Web Interface	http://127.0.0.1:4040
Forwarding	http://b0f7ac5f.ap.ngrok.io -> localhost:80
Forwarding	https://b0f7ac5f.ap.ngrok.io -> localhost:80

## 5.2 버튼 활용 : 카카오플러스 친구 설정

- 2.2 카카오플러스친구 설정은 ngrok 을 새로 실행한 경우에만, API 테스트를 하세요.
- ngrok 이 실행되고 있으면(8시간 이내), 2.3 카카오톡으로 바로 가세요.



무료 발송 메시지 1,000건 ?

다른 플러스친구 선택하기 ▼

홈

메시지 +

쿠폰 +

1:1채팅 +

**스마트채팅**

친구그룹 관리

홍보하기

통계 +

관리 +

공지사항 >

고객센터 챗봇 바로가기 >

이용가이드 ↓

고객센터  
1544-4293  
(평일 10시 ~ 18시 30분)

## API형

별도의 개발의 통해 특정 답변을 요구하는 형태의 질문들을 설계하는 타입입니다.

API Document

### 앱 등록

앱 이름 비개발자를 위한 챗봇

앱 URL http://3fa3505b.ap.ngrok.io

API 테스트

**Required\***

keyboard OK

{"type":"buttons","buttons":["처음으로","챗봇등록","취소하기"]}

앱 설명 비개발자를 위한 챗봇 세미나(0707)를 위한 플러스 친구 앱 등록입니다.

### 알림받을 전화번호 ?

☒ 플러스친구 API의 개인정보 수집 및 이용에 동의합니다.

전화번호 대한민국(82) ▼

카카오톡 이용중인 전화번호를 입력해주세요.

인증

대한민국(82)

✕ 삭제

무료 발송 메시지 1,000건

다른 플러스친구 선택하기

홈

메시지

쿠폰

1:1채팅

**스마트채팅**

친구그룹 관리

홍보하기

통계

관리

공지사항

고객센터 챗봇 바로가기

이용가이드

고객센터  
1544-4293  
(평일 10시 ~ 18시 30분)

앱 이름 비개발자를 위한 챗봇

앱 URL

API 테스트

**Required\***  
keyboard OK  
{ "type": "buttons", "buttons": ["처음으로", "챗봇등록", "취소하기"] }

앱 설명 비개발자를 위한 챗봇 세미나(0707)를 위한 플러스 친구 앱 등록입니다.

알림받을 전화번호

☒ 플러스친구 API의 개인정보 수집 및 이용에 동의합니다.

전화번호

대한민국(82)

카카오톡 이용중인 전화번호를 입력해주세요.

인증

대한민국(82)

삭제

이전

**API형 저장하기**

무료 발송 메시지 1,000건

다른 플러스친구 선택하기 ▾

홈

메시지 +

쿠폰 +

1:1채팅 +

**스마트채팅**

친구그룹 관리

홍보하기

통계 +

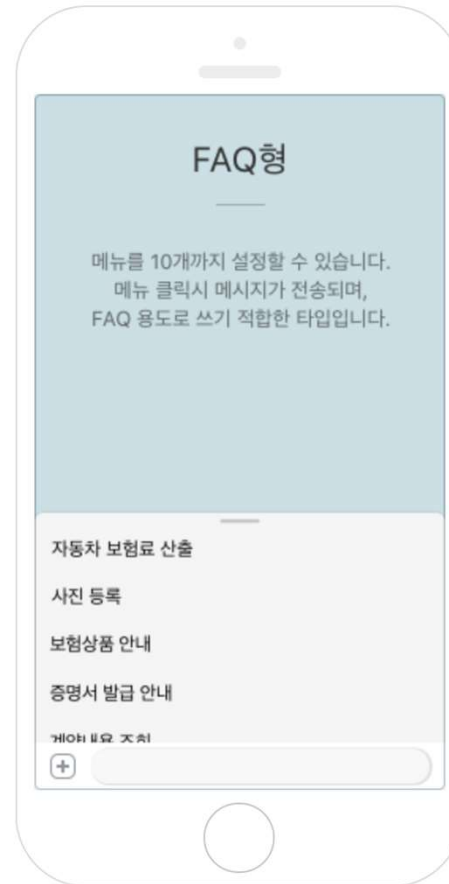
관리 +

공지사항 >

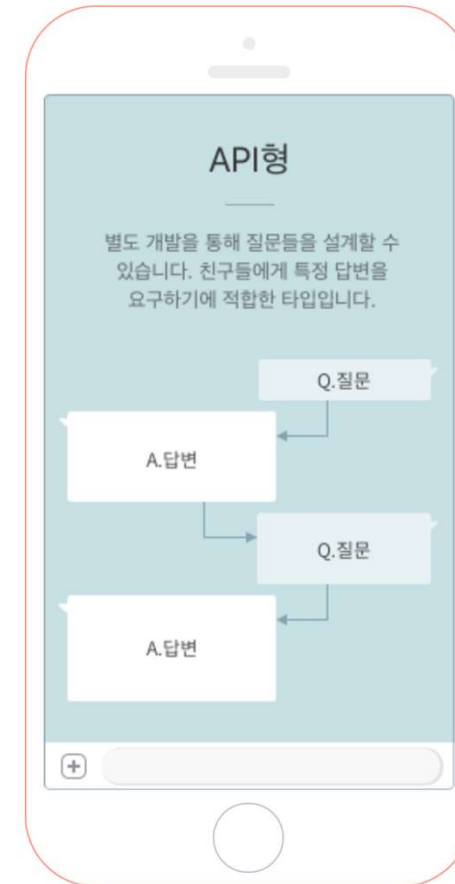
고객센터 챗봇 바로가기 >

이용가이드 ⬇

고객센터  
1544-4293  
(평일 10시 ~ 18시 30분)



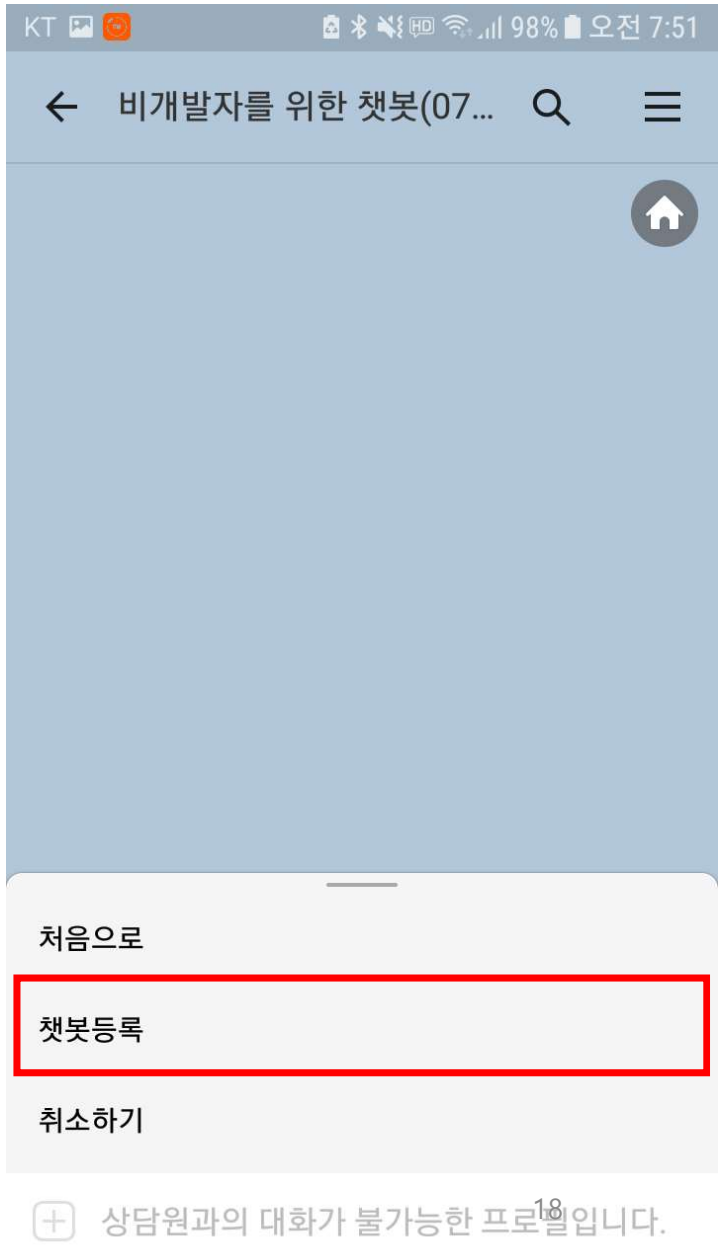
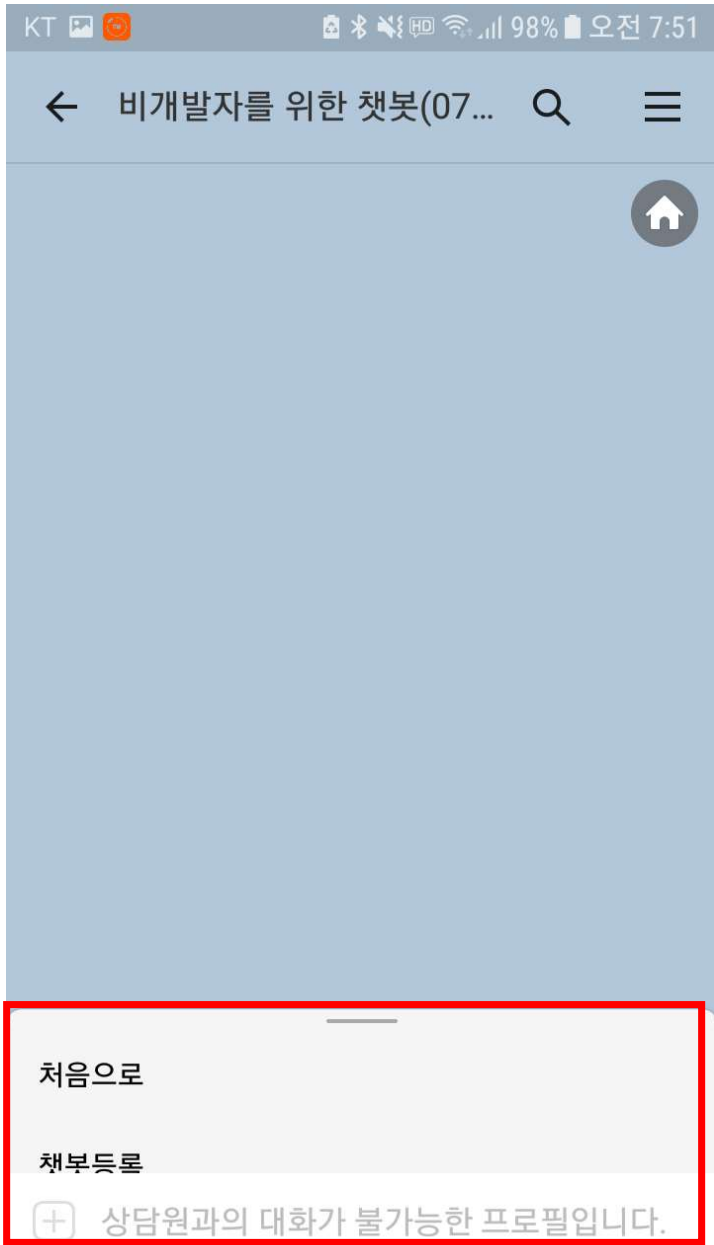
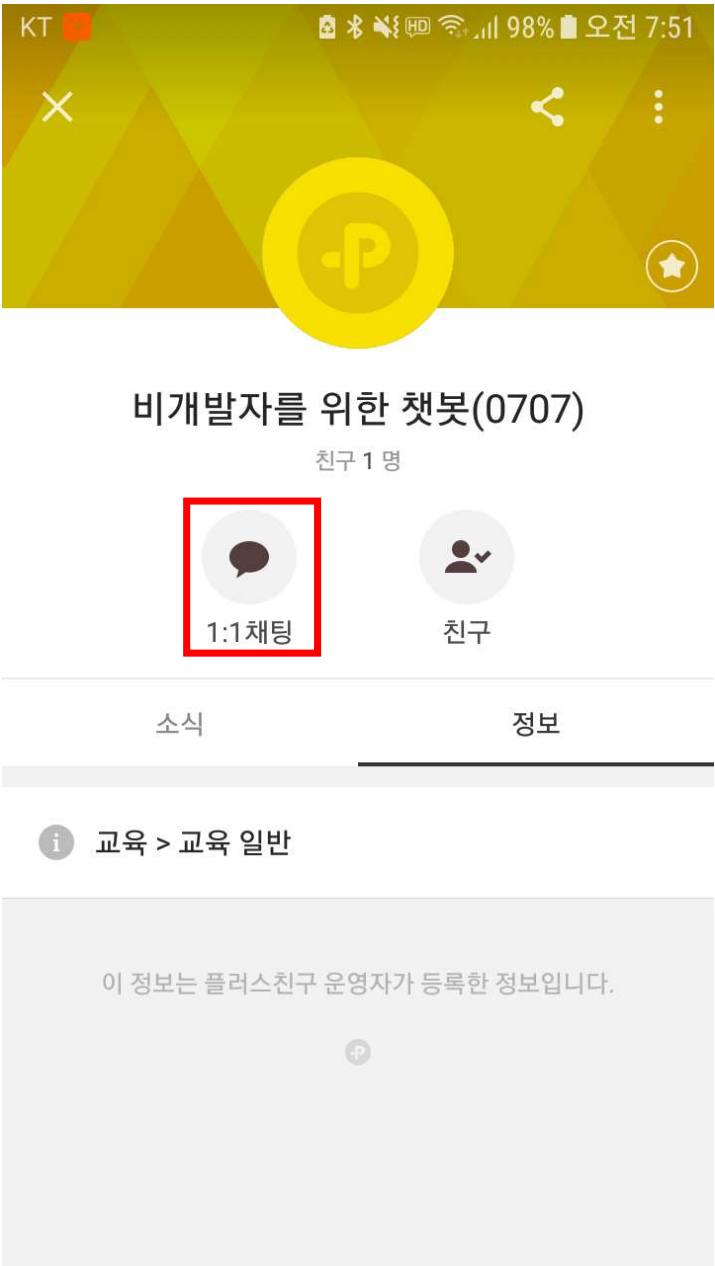
설정하기

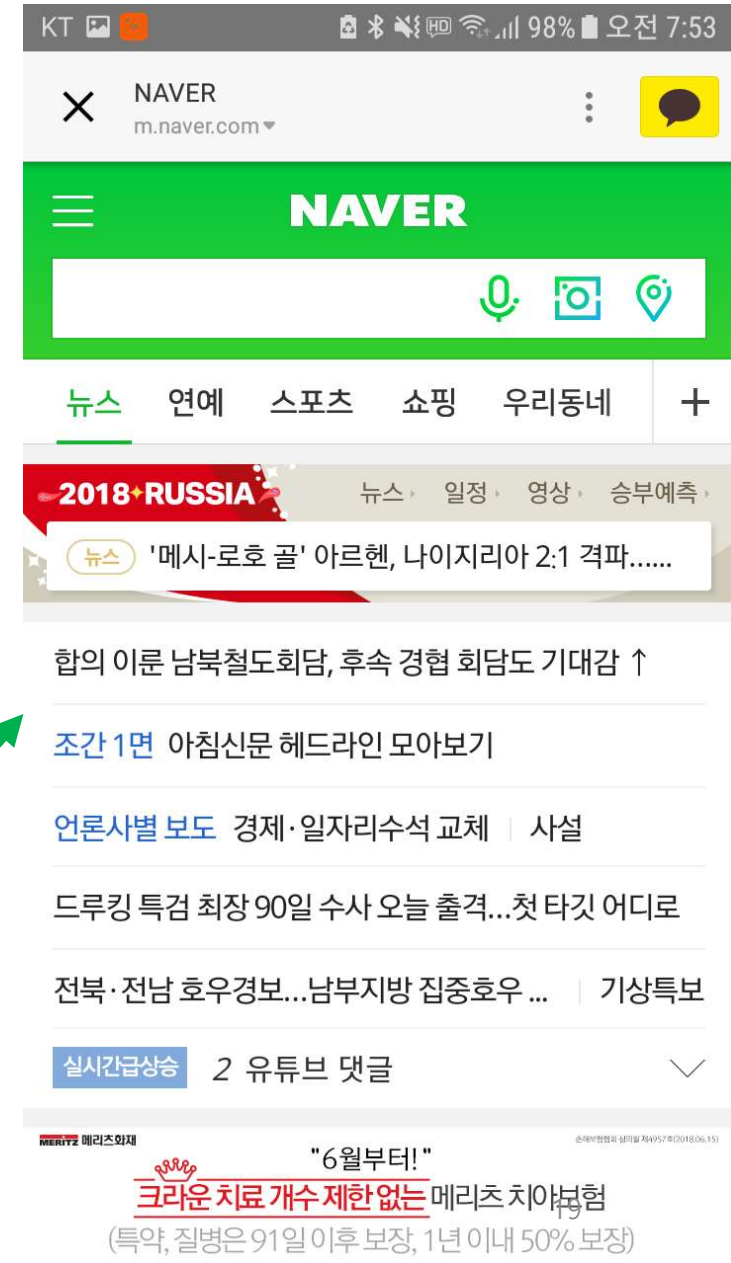
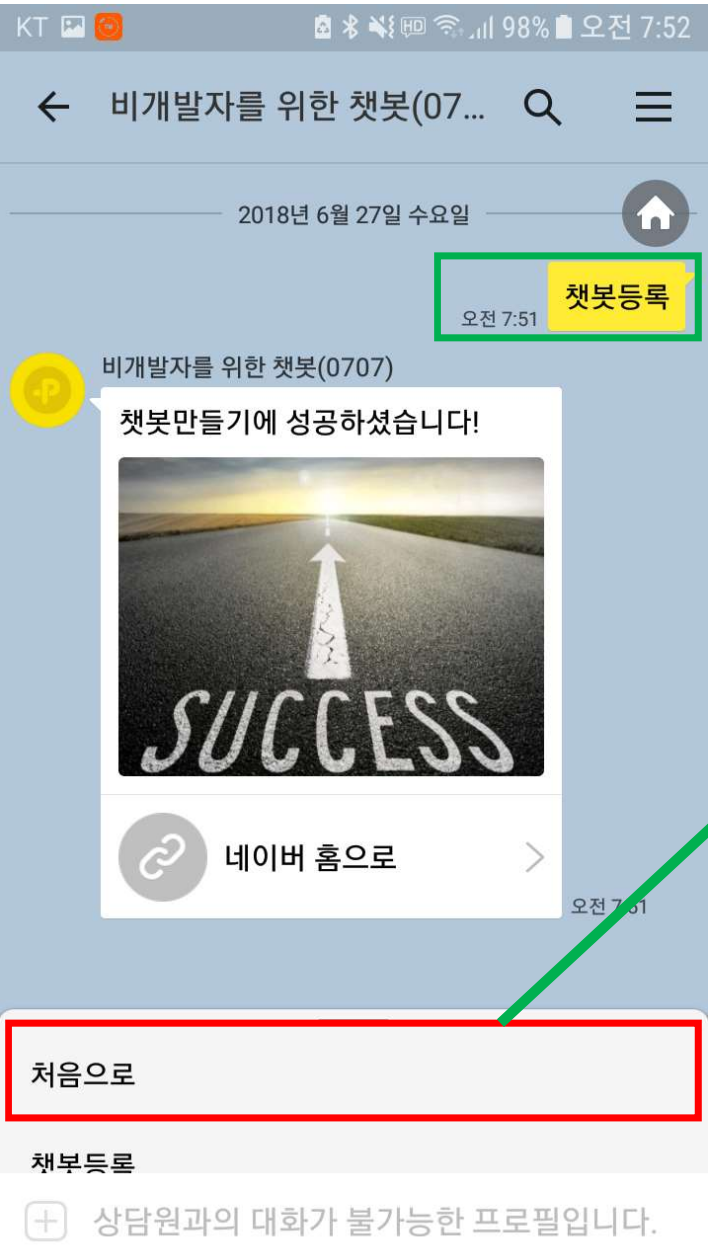


중지하기



## 5.3 버튼 : 카카오톡 테스트





## 5.4 버튼 : 에코봇에서 달라진 점

- 소스코드 분석

```
from flask import Flask
from flask import request
from flask import jsonify
from flask import json

app = Flask(__name__)

@app.route("/keyboard")
def keyboard():
    return jsonify(type="text")
```

```
@app.route("/message", methods=["POST"])
def message():
    data = json.loads(request.data)
    content = data["content"]

    response = {
        "message": {
            "text": content
        }
    }

    response = json.dumps(response, ensure_ascii=False)
    return response

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

```

from flask import Flask
from flask import request
from flask import jsonify
from flask import json

app = Flask(__name__)

@app.route("/keyboard")
def keyboard():
    response = {
        "type": "buttons",
        "buttons": ["처음으로", "챗봇등록", "취소하기"]
    }
    response = json.dumps(response, ensure_ascii=False)
    return response

@app.route("/message", methods=["POST"])
def message():
    data = json.loads(request.data)
    content = data["content"]
    if content == "처음으로":
        text = "처음으로 돌아왔습니다."
        url = "http://cfile237.uf.daum.net/image/2265F24B5603A3D704F183"
    elif content == "챗봇등록":
        text = "챗봇만들기에 성공하셨습니다!"
        url = "http://ppss.kr/wp-content/uploads/2017/05/3-51-540x303.jpg"
    else:
        text = "취소되었습니다."
        url = "http://thumbs.dreamstime.com/x/cancel-sign-button-3053748.jpg"

```

```

response = {
    "message": {
        "text": text,
        "photo": {
            "url": url,
            "width": 640,
            "height": 480
        },
        "message_button": {
            "label": "네이버 홈으로",
            "url": "https://naver.com"
        }
    },
    "keyboard": {
        "type": "buttons",
        "buttons": [
            "처음으로",
            "챗봇등록",
            "취소하기"
        ]
    }
}

response = json.dumps(response, ensure_ascii=False)
return response

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)

```

# @app.route("/keyboard")

- 키보드 : GET /keyboard
- 이용자가 최초로 채팅방에 들어올 때, 기본으로 키보드 영역에 표시될 자동응답 명령어의 목록을 호출하는 API입니다.

```
@app.route("/keyboard")
def keyboard():
    return jsonify(type="text")
```



```
@app.route("/keyboard")
def keyboard():
    response = {
        "type": "buttons",
        "buttons": ["처음으로", "챗봇등록", "취소하기"]
    }
    response = json.dumps(response, ensure_ascii=False)
    return response
```

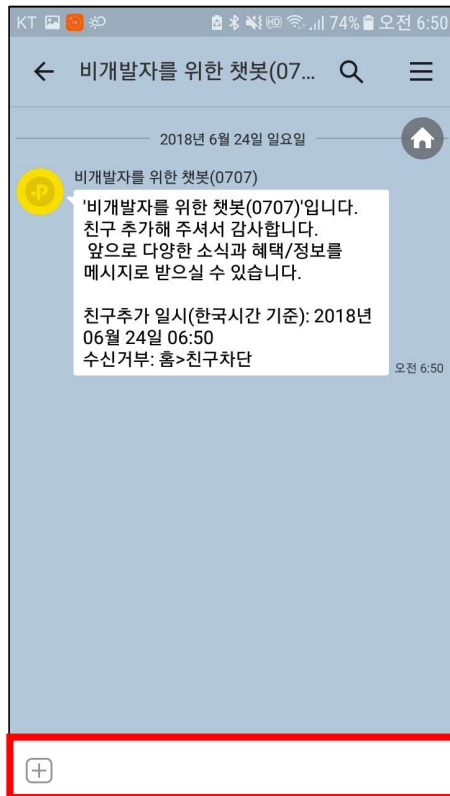
텍스트 봇

버튼 봇

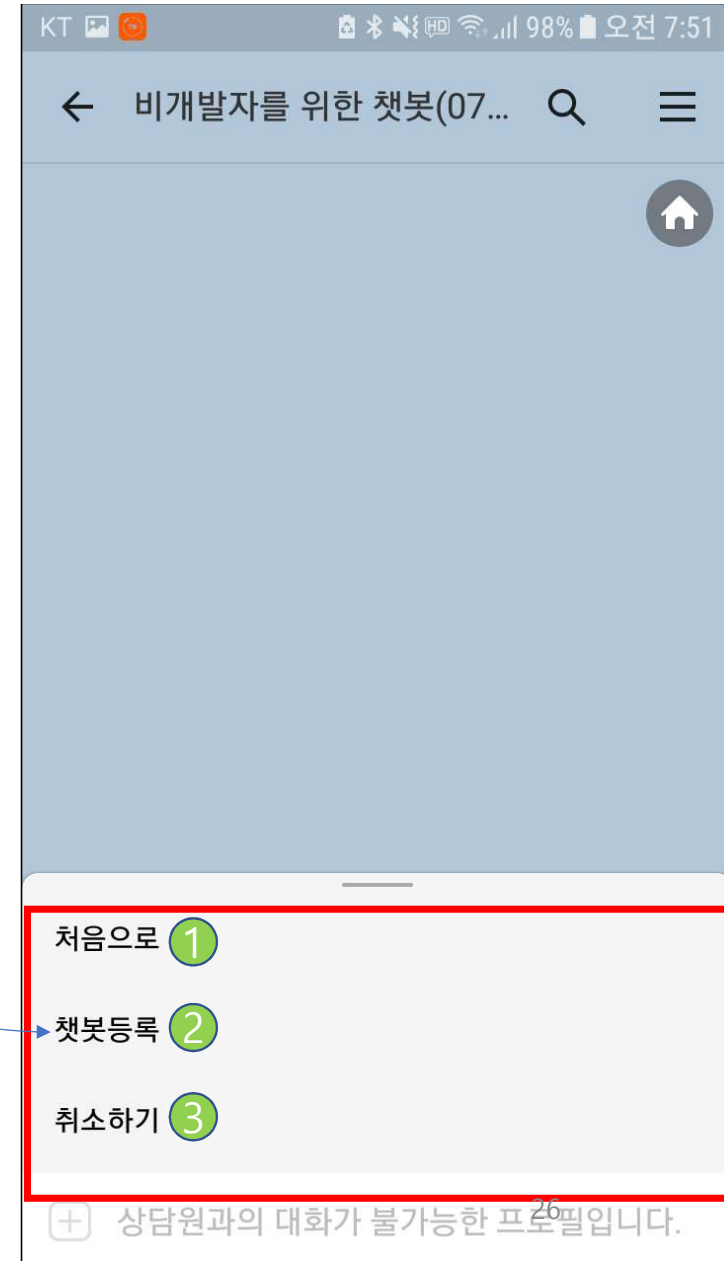


```
@app.route("/keyboard")
def keyboard():
    return jsonify(type="text")
```

```
@app.route("/keyboard")
def keyboard():
    response = {
        "type": "buttons",
        "buttons": ["처음으로", "챗봇등록", "취소하기"]
    }
    response = json.dumps(response, ensure_ascii=False)
    return response
```



```
@app.route("/keyboard")
def keyboard():
    response = {
        "type" : "buttons",
        "buttons" : ["처음으로", "챗봇등록", "취소하기"]
    }
    response = json.dumps(response, ensure_ascii=False)
    return response
```



```
@app.route("/message",  
methods=["POST"])
```

- 메시지 : POST /message
- 자동응답 API에서 제일 핵심인 내용입니다.
- 사용자가 보내는 메세지/사진/동영상을 이 API를 통해서 보내고 서버가 처리해서 그에 맞는 응답을 해주면 됩니다.

```

from flask import Flask
from flask import request
from flask import jsonify
from flask import json

```

```

app = Flask(__name__)

```

```

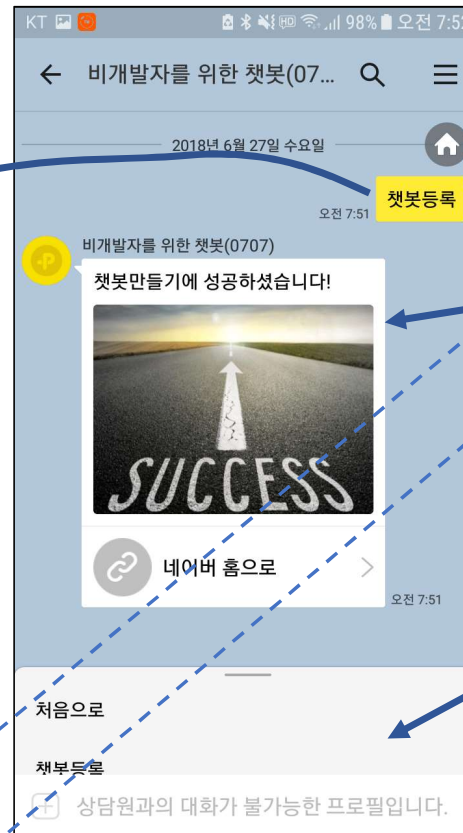
@app.route("/keyboard")
def keyboard():
    response = {
        "type": "buttons",
        "buttons": ["처음으로", "챗봇등록", "취소하기"]
    }
    response = json.dumps(response, ensure_ascii=False)
    return response

```

```

@app.route("/message", methods=["POST"])
def message():
    data = json.loads(request.data)
    content = data["content"]
    if content == "처음으로":
        text = "처음으로 돌아왔습니다."
        url = "http://cfil2v7.uf.daum.net/image/2265F24B5603A3D704F183"
    elif content == "챗봇등록":
        text = "챗봇만들기에 성공하셨습니다!"
        url = "http://ppss.kr/wp-content/uploads/2017/05/3-51-540x303.jpg"
    else:
        text = "취소되었습니다."
        url = "http://thumbs.dreamstime.com/x/cancel-sign-button-3053748.jpg"

```



```

response = {
    "message": {
        "text": text,
        "photo": {
            "url": url,
            "width": 640,
            "height": 480
        },
        "message_button": {
            "label": "네이버 홈으로",
            "url": "https://naver.com"
        }
    },
    "keyboard": {
        "type": "buttons",
        "buttons": [
            "처음으로",
            "챗봇등록",
            "취소하기"
        ]
    }
}

```

```

response = json.dumps(response, ensure_ascii=False)
return response

```

```

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)

```

```
@app.route("/message", methods=["POST"])
def message():
```

```
    data = json.loads(request.data)
    content = data["content"]
```

```
    if content == "처음으로" :
```

```
        text = "처음으로 돌아왔습니다."
```

```
        url = "http://cfile237.uf.daum.net/image/2265F24B5603A3D704F183"
```

```
    elif content == "챗봇등록" :
```

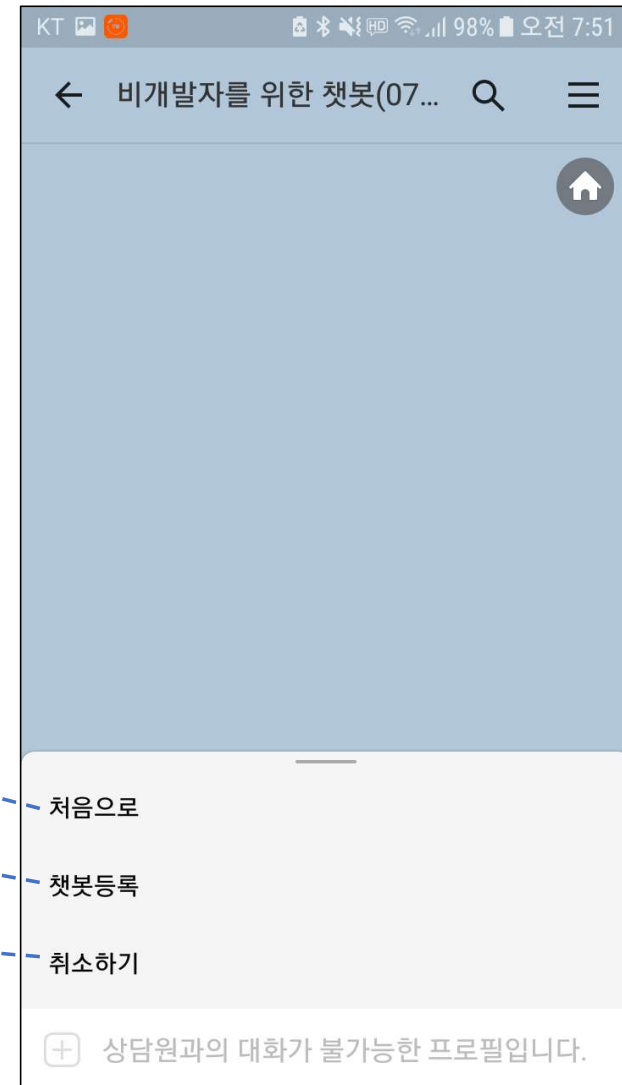
```
        text = "챗봇만들기에 성공하셨습니다!"
```

```
        url = "http://ppss.kr/wp-content/uploads/2017/05/3-51-540x303.jpg"
```

```
    else :
```

```
        text = "취소되었습니다."
```

```
        url = "http://thumbs.dreamstime.com/x/cancel-sign-button-3053748.jpg"
```



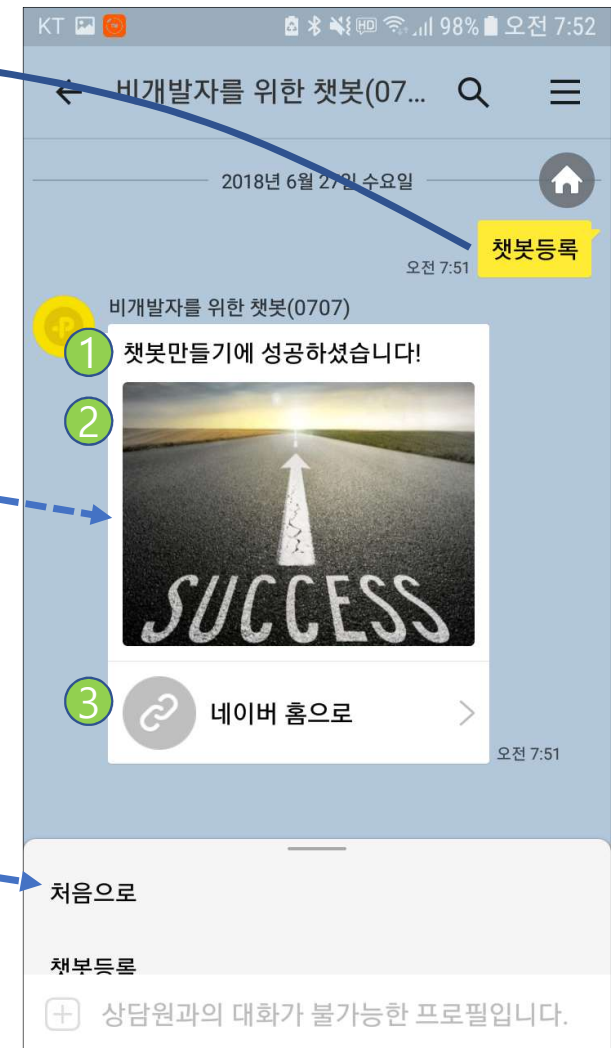
```
response = {
```

```
  "message" : {  
    1 "text": text,  
    2 "photo": {  
      "url": url,  
      "width": 640,  
      "height": 480  
    },  
    3 "message_button": {  
      "label": "네이버 홈으로",  
      "url": "https://naver.com"  
    }  
  },  
  "keyboard" : {  
    "type": "buttons",  
    "buttons": [  
      "처음으로",  
      "챗봇등록",  
      "취소하기"  
    ]  
  }  
}
```

```
elif content == "챗봇등록" :
```

```
  1 text = "챗봇만들기에 성공하셨습니다!"  
  2 url = "http://ppss.kr/~x303.jpg"
```

```
response = json.dumps(response, ensure_ascii=False)  
return response
```



# 버튼

대화형 인터페이스에서 가장 유용한 리치 컨트롤

1. 대화나 인터랙션의 방식을 **안내**
2. 사용자가 할 수 있는 대답을 **제한**
3. 미리 준비된 응답들을 제공 – 새 메시지 전송 – 클릭한 버튼에 따라 대화를 어떻게 이어 나갈 것인지 결정
4. 내비게이션 – 기존 메시지를 교체 – 강력한 기능 – 필터링 기능, 이전 다음 버튼 제공

# 버튼의 활용 용도

- 봇 첫 환영 화면에 시작하기 버튼
- URL 버튼 - 별도의 새로운 웹 페이지 열기
- 공유 버튼 - 대화 내용 공유
- 구매 버튼 - 대화의 일부로서 구매할 수 있는 항목들
- 전화걸기 버튼
- 로그인, 로그아웃 버튼
- (빠른 답변 - 올바른 대화 흐름 유도) + 자유로운 텍스트 처리



# 대화중 언제 버튼을 사용하나

## 버튼 사용 권장

- 내비게이션
- 더 나은 대화 흐름
- 예, 아니오 같은 구조화된 응답

## 버튼 사용 제한

- 선택항목이 많은 경우 - 버튼의 수 제한
- 이메일, 주소, 고객 이름
- 우리의 기분 묘사시
- 재무 타당성 제공시



take a break

JAESEOK LEE  
[www.ttearth.com](http://www.ttearth.com) / [www.facebook.com/EonNow](https://www.facebook.com/EonNow)