

CHATBOTHON

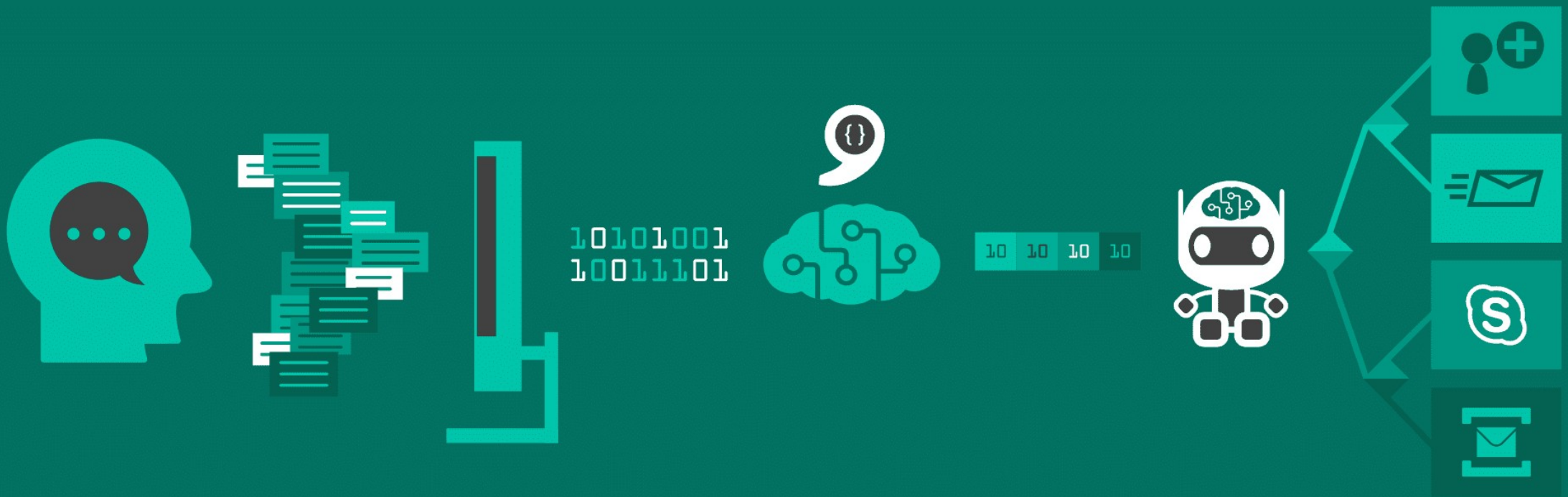
[행사개요](#) [아이디어](#) [시상금](#) [분야](#) [후원사](#)

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)
05 버튼봇	12:00 – 12:30(30분)
06 api봇	12:30 – 01:00(30분)

06 에코봇 + api 활용(주가봇)

가상화폐 종가 알림봇 만들기

6.1 가상화폐 : python 실행



탐색기

열려 있는 편집기

× kakao_bot.py 04.fi...

KAKAO-BOT-MASTER

▶ 00.hello-world

▶ 01.eco-bots

▶ 02.translate-bots

▶ 03.button-bots

▶ 04.finance-bots

🔗 kakao_bot.py

▶ 05.cognitive-bots

▶ 06.dialogflow-bots

≡ ngrok.exe

📘 README.md



▶ 개요

🔗 kakao_bot.py ×

```
21 @app.route("/message", methods=["POST"])
22 def message():
23     data = json.loads(request.data)
24     content = data["content"]
25
26     if content == str("코인").decode('utf-8'):
27         response = urlopen("https://api.bithumb.com/public/ticker/").read()
28         responsejson = json.loads(response)
29         coin = responsejson['data']['opening_price']
30         content = coin
31
32     response = {
33         "message": {
34             "text": content
35         }
36     }
37
38     response = json.dumps(response, ensure_ascii=False)
39     return response
40
41
42 if __name__ == "__main__":
43     app.run(host="0.0.0.0", port=5000)
```





탐색기

열려 있는 편집기

x kakao_bot.py 04.fi...

KAKAO-BOT-MASTER

00.hello-world

01.eco-bots

02.translate-bots

03.button-bots

04.finance-bots

kakao_bot.py

05.cognitive-bots

06.dialogflow-bot

ngrok.exe

i README.md



개요

kakao_bot.py x

```
21 @app.route("/message", methods=["POST"])
22 def message():
23     data = json.loads(request.data)
24     content = data["content"]
25
26     if content == str("코인").decode('utf-8'):
27         response = urlopen("https://api.bithumb.com/public/ticker/").read()
28         responsejson = json.loads(response)
29         coin = responsejson['data']['opening_price']
30         content = coin
31
32     response = {
33         "message": {
34             "text": content
35         }
36     }
37
38     response = json.dumps(response, ensure_ascii=False)
39     return response
40
41
42 if __name__ == "__main__":
43     app.run(host="127.0.0.1", port=8080)
```

측면에서 열기 Ctrl+Enter

탐색기에 표시 Alt+Shift+R

터미널에서 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



탐색기

열려 있는 편집기

× kakao_bot.py 04.fi...

KAKAO-BOT-MASTER

▶ 00.hello-world

▶ 01.eco-bots

▶ 02.translate-bots

▶ 03.button-bots

▶ 04.finance-bots

kakao_bot.py

▶ 05.cognitive-bots

▶ 06.dialogflow-bots

ngrok.exe

README.md



▶ 개요

kakao_bot.py ×

```
21 @app.route("/message", methods=["POST"])
22 def message():
23     data = json.loads(request.data)
24     content = data["content"]
25
26     if content == str("코인").decode('utf-8'):
27         response = urlopen("https://api.bithumb.com/public/ticker/").read()
28         responsejson = json.loads(response)
29         coin = responsejson['data']['opening_price']
30         content = coin
31
32     response = {
```

문제 출력 디버그 콘솔 터미널

1: powershell

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\jaeseok\kakao-bot-master\04.finance-bots> python .\kakao_bot.py



탐색기

열려 있는 편집기

× kakao_bot.py 04.fi...

KAKAO-BOT-MASTER

▶ 00.hello-world

▶ 01.eco-bots

▶ 02.translate-bots

▶ 03.button-bots

▶ 04.finance-bots

kakao_bot.py

▶ 05.cognitive-bots

▶ 06.dialogflow-bots

ngrok.exe

README.md



개요

kakao_bot.py ×

```
21 @app.route("/message", methods=["POST"])
22 def message():
23     data = json.loads(request.data)
24     content = data["content"]
25
26     if content == str("코인").decode('utf-8'):
27         response = urlopen("https://api.bithumb.com/public/ticker/").read()
28         responsejson = json.loads(response)
29         coin = responsejson['data']['opening_price']
30         content = coin
31
32     response = {
```

문제 출력 디버그 콘솔 터미널

1: python

Copyright (C) Microsoft Corporation. All rights reserved.

```
PS C:\Users\jaeseok\kakao-bot-master\04.finance-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)
```

11

kakao_bot.py - kakao-bot-master - Visual Studio Code

파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 디버그(D) 작업(T) 도움말(H)

탐색기

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

kakao_bot.py

```
21 @app.route("/message", methods=["POST"])
22 def message():
23     data = json.loads(request.data)
24     content = data["content"]
25
26     if content == str("코인").decode('utf-8'):
27         response = urlopen("https://api.bithumb.com/public/ticker/krw-usdt")
28         responsejson = json.loads(response.read())
29         coin = responsejson['data']['opening_price']
30         content = coin
31
32     response = {
```

문제 출력 디버그 콘솔 터미널

2: ngrok

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://27518c7e.ap.ngrok.io -> localhost:80
Forwarding	https://27518c7e.ap.ngrok.io -> localhost:80

줄 43, 열 39

Python 3.6.5 :: Anaconda, Inc.

Ngrok 의
세션이 만료
되지 않은
경우

python 만
재실행하면
됩니다.



탐색기

열려 있는 편집기

× kakao_bot.py 04.fi...

KAKAO-BOT-MASTER

▶ 00.hello-world

▶ 01.eco-bots

▶ 02.translate-bots

▶ 03.button-bots

▶ 04.finance-bots

kakao_bot.py

▶ 05.cognitive-bots

▶ 06.dialogflow-bots

ngrok.exe

README.md



▶ 개요

kakao_bot.py ×

```
21 @app.route("/message", methods=["POST"])
22 def message():
23     data = json.loads(request.data)
24     content = data["content"]
25
26     if content == str("코인").decode('utf-8'):
27         response = urlopen("https://api.bithumb.com/public/ticker/").read()
28         responsejson = json.loads(response)
29         coin = responsejson['data']['opening_price']
30         content = coin
31
32     response = {
```

문제 출력 디버그 콘솔 터미널

2: powershell

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\jaeseok\kakao-bot-master> .\ngrok http 80 --region ap

6.2 카카오플러스친구 설정

- 6.2 카카오플러스친구 설정은 ngrok 을 새로 실행한 경우에만, API 테스트를 하세요.
- ngrok 이 실행되고 있으면(8시간 이내), 3.3 카카오톡으로 바로 가세요.



비개발자를 위한 챗
봇(0707)

총 캐시 0원 >
메시지 이용권 0개 >
무료 발송 메시지 1,000건 ?

다른 플러스친구 선택하기 ▾

홈

메시지 +

쿠폰 +

1:1채팅 +

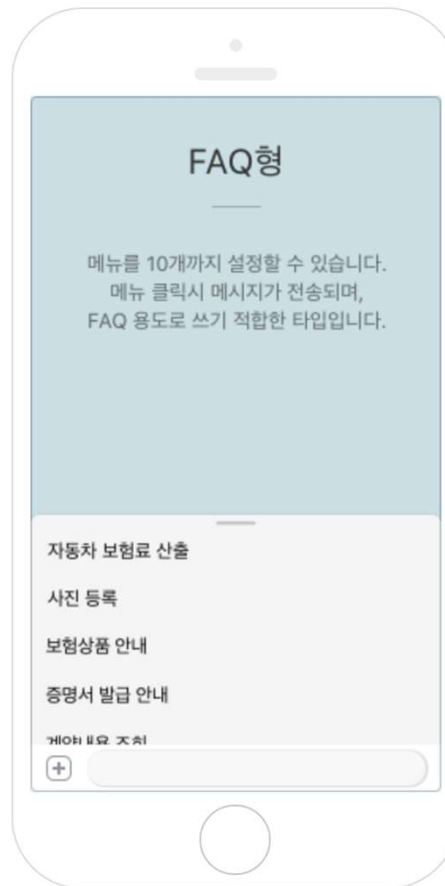
스마트채팅

친구그룹 관리

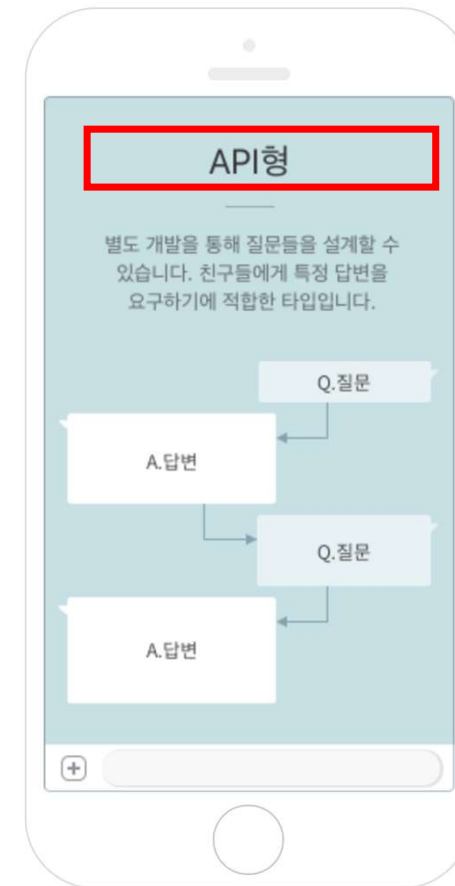
홍보하기

통계 +

관리 +



설정하기



설정하기



비개발자를 위한 챗
봇(0707)

총 캐시 0원 >
메시지 이용권 0개 >
무료 발송 메시지 1,000건 ?

다른 플러스친구 선택하기 ▼

홈

메시지 +

쿠폰 +

1:1채팅 +

스마트채팅

친구그룹 관리

홍보하기

통계 +

관리 +

API형

별도의 개발의 통해 특정 답변을 요구하는 형태의 질문들을 설계하는 타입입니다.

API Document

앱 등록

앱 이름	비개발자를 위한 챗봇	
앱 URL	http://96ad1664.ap.ngrok.io	API 테스트
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



비개발자를 위한 챗
봇(0707)

총 캐시 0원 >
메시지 이용권 0개 >
무료 발송 메시지 1,000건 ?

다른 플러스친구 선택하기 ▾

홈

메시지 +

쿠폰 +

1:1채팅 +

스마트채팅

친구그룹 관리

홍보하기

통계 +

관리 +

API형

별도의 개발의 통해 특정 답변을 요구하는 형태의 질문들을 설계하는 타입입니다.

API Document

앱 등록

앱 이름 비개발자를 위한 챗봇

앱 URL http://96ad1664.ap.ngrok.io

API 테스트

Required*
keyboard OK
{ "type": "text" }

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



비개발자를 위한 챗
봇(0707)

총 캐시 0원 >

메시지 이용권 0개 >

무료 발송 메시지 1,000건 ?

다른 플러스친구 선택하기 ▼

홈

메시지 +

쿠폰 +

1:1채팅 +

스마트채팅

친구그룹 관리

홍보하기

통계 +

관리 +

앱 이름 비개발자를 위한 챗봇

앱 URL http://96ad1664.ap.ngrok.io

API 테스트

Required*
keyboard OK
{"type": "text"}

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



비개발자를 위한 챗
봇(0707)

총 캐시 0원 >

메시지 이용권 0개 >

무료 발송 메시지 1,000건 ?

다른 플러스친구 선택하기 ▾

홈

메시지 +

쿠폰 +

1:1채팅 +

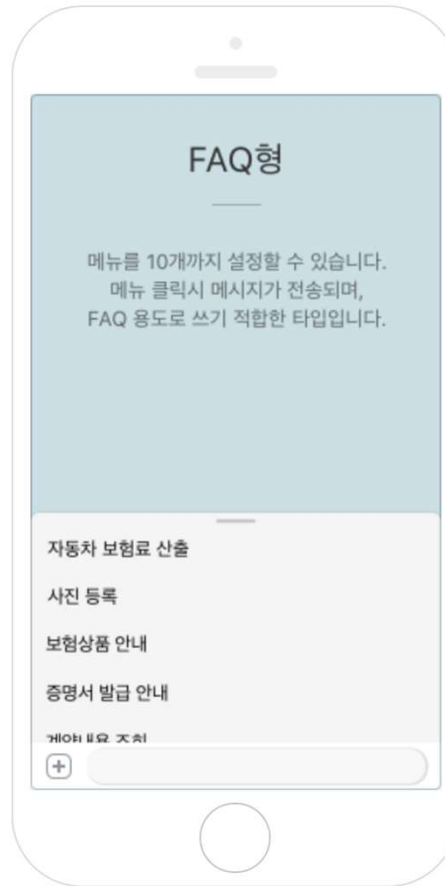
스마트채팅

친구그룹 관리

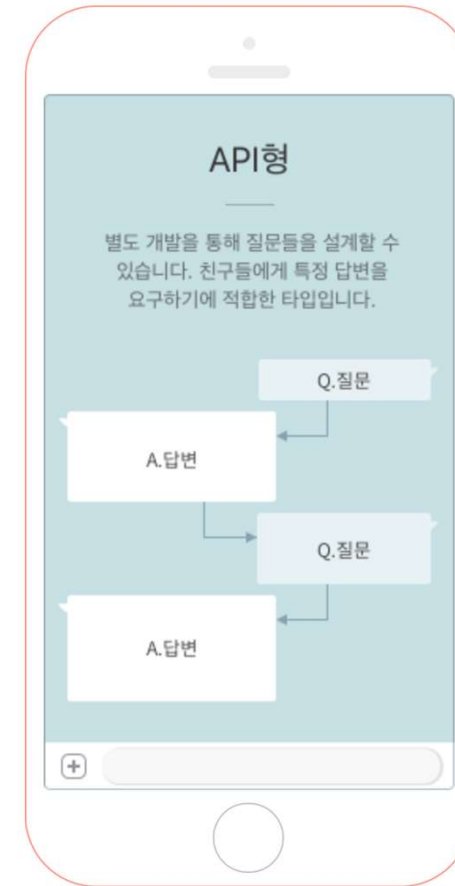
홍보하기

통계 +

관리 +

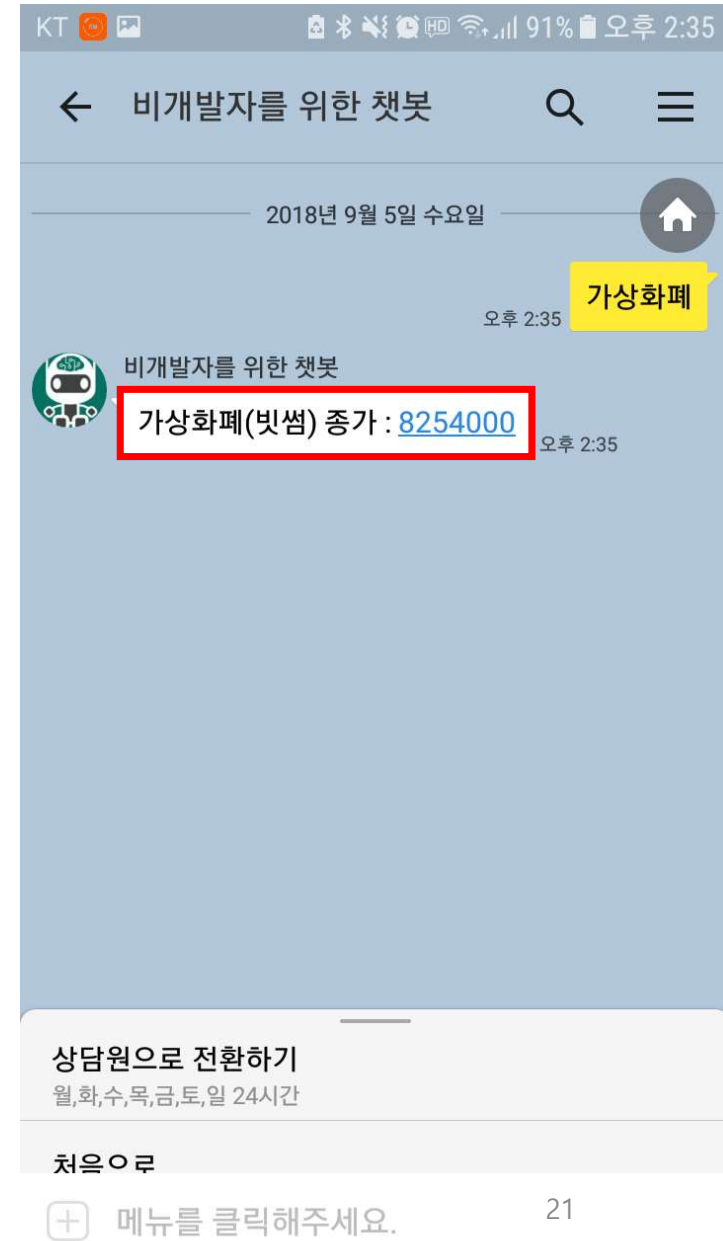
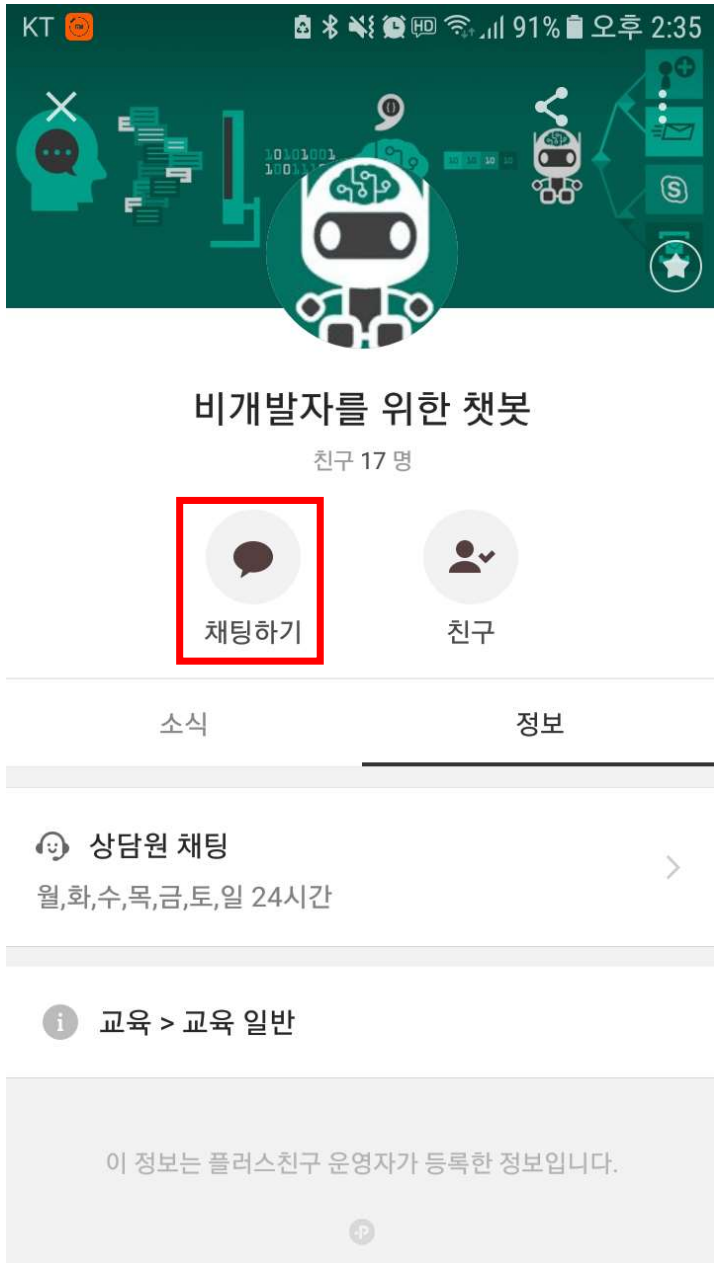


설정하기



중지하기

6.3 카카오톡 테스트



6.4 에코봇에서 달라진 점

- 소스 코드 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
from urllib.request import urlopen
```

```
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"]
```

```
    if content == "코인":
        response = urlopen("https://api.bithumb.com/~~").read().decode('utf-8')
        responsejson = json.loads(response)
        coin = responsejson['data']['opening_price']
        content = coin
```

```
    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)
```



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



```
from urllib.request import urlopen
```

```
if content == "코인" :
```

```
    response = urlopen("https://api.bithumb.com/~~").read().decode('utf-8')
```

```
    responsejson = json.loads(response)
```

```
    content = responsejson['data']['opening_price']
```

에코 봇

코인 봇

6.5 웹 크롤링

- 원하는 데이터를 가져오는 방법

웹 스크레이퍼(크롤링)

- 웹 서버에 특정 페이지를 요청하는 GET 요청을 보내고,
- 그 페이지의 HTML 결과(우리는 JSON)를 얻고,
- 우리가 원하는 콘텐츠를 뽑아내는 과정을 말한다.

API 사용

- 최근 API 라고 하면, 보통 웹 애플리케이션 API를 말합니다.
- API가 요청을 보낼 때는 HTTP를 통해 데이터를 요청하면
- API는 이 데이터를 XML 이나 JSON 형태로 반환합니다.
- API 크롤링이 일반 크롤링보다 사용하기 더 쉽습니다.
- 완전히 정형화된 문법을 사용하고,
- HTML 이 아닌 JSON 형태로 데이터를 보내기 때문입니다.

```
from urllib.request import urlopen
```

```
response = urlopen("https://api.bithumb.com/~~").read().decode('utf-8')
```

```
responsejson = json.loads(response)
```

```
coin = responsejson['data']['opening_price']
```

from urllib.request import urlopen

- urllib 라이브러리에서
 - 파이썬 모듈 request 를 읽고
 - urlopen 함수 하나만 import 합니다.
-
- urllib 는 표준 라이브러리로서, 따로 설치할 필요가 없습니다.
 - urllib 는 웹을 통해 데이터를 요청합니다.
 - urlopen 은 HTML, 이미지 등을 쉽게 열 수 있는 함수입니다.
 - <https://docs.python.org/3/library/urllib.html>

외부 모듈 사용

- 함수보다 더 큰 개념으로 class 가 있고,
- 클래스보다 더 큰 개념으로 module 이 있습니다.
- 모듈은 파이썬 프로그램 단위(파일 단위 .py) 로 생성됩니다.
- library 란 여러 개의 module 로 구성되어 있습니다.
- 함수 < class < module < library

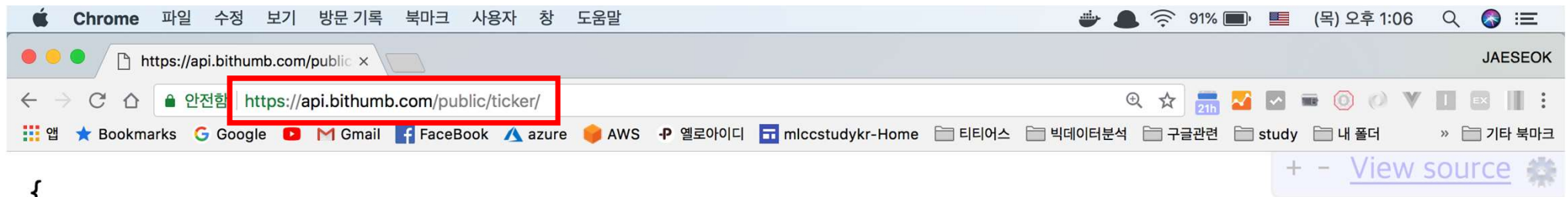
외부 라이브러리

- 외부 라이브러리를 사용하려면,
- 라이브러리를 설치한 후,
- 원하는 함수가 들어 있는 모듈을 호출(import) 합니다.

urlopen("https://api.bithumb.com/~ ~")

- urlopen() : HTML, 이미지 등을 쉽게 열 수 있는 함수, 즉 인터넷에서 소스 코드를 긁어오는 기능을 합니다.
- response = **urlopen**("https://api.bithumb.com/~ ~").read().decode('utf-8')
- read()
- decode('utf-8')

<https://api.bithumb.com/public/ticker/>



```
{
  status: "0000",
  - data: {
    opening_price: "7193000",
    closing_price: "7195000",
    min_price: "7136000",
    max_price: "7263000",
    average_price: "7208682.4278",
    units_traded: "2429.06983868",
    volume_1day: "2429.06983868",
    volume_7day: "28175.95518144",
    buy_price: "7190000",
    sell_price: "7196000",
    24H_fluctate: 2000,
    24H_fluctate_rate: "0.02",
    date: "1531368324230"
  }
}
```



빗썸 API 명세 - RESPONSE

Returned value description

```
{
  "status": "0000",
  "data": {
    "opening_price" : "504000",
    "closing_price" : "505000",
    "min_price" : "504000",
    "max_price" : "516000",
    "average_price" : "509533.3333",
    "units_traded" : "14.71960286",
    "volume_1day" : "14.71960286",
    "volume_7day" : "15.81960286",
    "buy_price" : "505000",
    "sell_price" : "504000",
    "24H_fluctate" : "1000",
    "24H_fluctate_rate" : "0.19",
    "date" : 1417141032622
  }
}
```

// 상태코드
// 24시간 데이터
// 거래시작금액
// 마지막 거래금액
// 최저 거래가
// 최고 거래가
// 평균 거래가
// 거래량
// 거래량
// 7일간 거래량
// 구매가
// 판매가
// 변동 금액
// 변동률
// 타임스탬프

공공데이터포털 : api 제공

<https://www.data.go.kr/>

데이터셋

제공신청

활용사례

정보공유

이용안내

데이터를 검색해보세요!



2.미세먼지

여러분이 원하는 데이터, **데이터 1번가** 로 신청해주세요!

국가중점데이터

데이터 카테고리



교육



국토관리



공공행정



재정금융



산업고용



사회복지



식품건강



문화관광



보건의료



재난안전



교통물류



환경기상



과학기술



농축수산



통일외교안보



법률

「데이터1번가」

여러분이 필요로 하는 공공데이터를 자유롭게
신청할 수 있는 데이터 소통창구

[신청하기](#) [자세히보기](#)

오픈데이터 포럼

민·관 협력을 통한 공공데이터 개방 정책 수립 및
공공데이터 활용성 증진을 위해 운영되는 단체

**공공데이터가
일 자리를
만듭니다!**

공공데이터 활용사례



인기 데이터

파일데이터

오픈 API

교통사고 통계 현황

api 가 아닌 직접 크롤링 영역

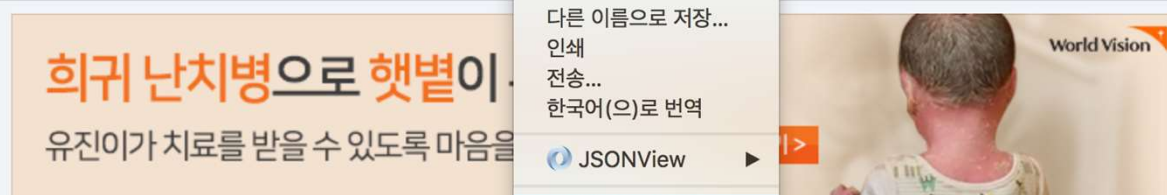
네이버를 시작페이지로 > | 주니어네이버 | 해피빈



메일 카페 블로그 지식IN 쇼핑 Pay TV

지도 영화 뮤직 책 웹툰 더보기

3 한서희



아이디 로그인 IP보안 ON 일회용 로그인

☐ 로그인 상태 유지 ☐ 회원가입 아이디/비밀번호 찾기

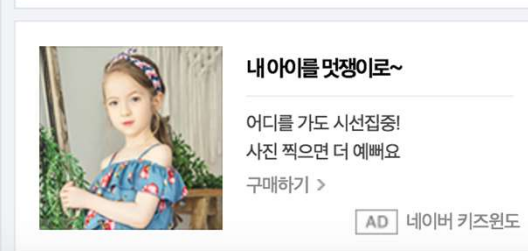
연합뉴스 > 한은, 올해 성장률 전망 2%대로 하향... "잠재성장률" 검사 음성

07.12 (목) 영어회화 > 5/6

뉴스스탠드 > 전체 언론사 MY 뉴스

당신 탓이 아니에요.
It's not your fault.
잇츠 낫 유어 폴트

JTBC	데일리안	한국경제TV	NewDaily	zdNet Korea	시사IN
KBSWORLD	ChosunBiz	연합뉴스TV	에럴드경제	동아일보	석간 문화일보
연합뉴스	CNN	MK 스포츠	MTN 머니투데이방송	뉴스컬처	매일노동뉴스







Thank you

JAESEOK LEE
www.ttearth.com / www.facebook.com/EonNow