

[| 자유 게시판 | >](#)

## [스압] 명령어 구분과 자연어 분석을 동시에 하기 ( 내가 봇을 쓰는 방법 )



윤 챗봇 입문자 1:1 채팅

2020.06.07. 01:13 조회 390

💬 댓글 45 URL 복사 ⋮



안녕하세요. 솔 처묵처묵한 윤입니다.

의외로 제가 사용하는 봇을 궁금하신 분이 계시더라고요.

언젠가 제 프로그래밍 작성 노하우도 기술해야겠다 생각했는데, 좋은 기회인 것 같아 이런 글을 작성하게 되었습니다.

첫째로, 저는 전에 쓴 글을 토대로 봇 코딩을 하는데 바벨을 사용합니다.

둘째로, **response** 함수엔 최저한의 코드를 사용합니다.

셋째로, 명령어를 추가하든 if 가 아니라 **Object** 를 사용합니다.

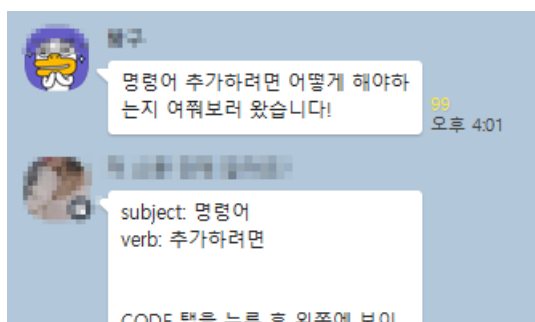
넷째로, 저는 하나의 서비스를 이용하는 커뮤니티 방을 운영하고 있기 때문에, 자주하는 질문에 대한 자연어를 분석해서 자동으로 대답하는 기능이 있습니다.

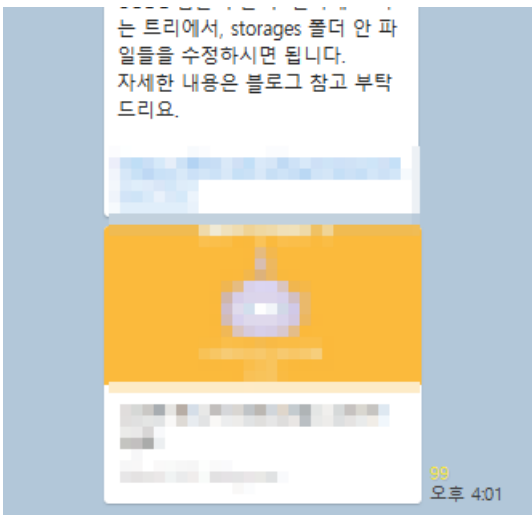
다섯째, 전에 쓴 글을 토대로 수정하면 즉시 적용 가능한 상황입니다.

즉, 보편적인 봇이 아니라 다른 상황을 가지고 있습니다.

제가 항상 작성하는 것은 강좌가 아니라, 경험 기술입니다.

글을 항상 신뢰하지 마시고, 참고만 하시기 바랍니다.





자연어를 분석해서 대답하는 봇

실질적으로 제가 설명할 것은 몇 개 되지 않습니다.

- □ 유지보수가 쉬운 프로그래밍
- □ **WTF** 정규식
- ✂ 정규식을 이용해 **KEY** 추출
- □ 키 조합으로 답변 추출

## □ 유지보수가 쉬운 프로그래밍



현업에 종사하면서, 제가 지금까지 가장 많이 들었던 "좋은 코드" 란 다음과 같았습니다.

“

사람이 해석하기 쉬운 코드

네, 프로그래밍은 컴퓨터를 이해시키기 위한 작업인데, 사람이 이해하기 쉬워야 한다고 합니다.  
이게 무슨 소리냐 하면, 사람이 이해하기 쉬워서 유지보수 하기 용이한 코드가 좋은 코드라고 합니다.

하지만 내가 짠 코드에 대해 다른 사람이 밥그릇 뺏는 것을 보기 싫으면, 제가 쓴 난독화 글을 참고하십시오.

### 가장 완벽한 소스 난독화 기법

어쨌든, 봇은 보통 명령어를 호출하여 특정 상황에 대해 대처를 합니다.  
그러면 명령어 추가는 쉬워야 합니다.

저는 다음과 같은 순서로 명령어를 처리합니다.

1. 문자열 입력받음
2. 명령어 시그니처(!, @, 등) 를 검사. ( if 시그니처 부합시 )
3. 명령 모음 오브젝트에서 명령어 검사. ( 명령어는 공백없는 문자 )
4. 명령어 존재시, 함수 실행 또는 문자열 반환 또는 배열에서 랜덤 반환
5. 반환 받은 결과 전송

위와 같은 순서로 이루어집니다.

일단 저는 취한 상태이니, 2번부터 빠르게 진행하고 대충 끝내겠습니다.

```

1  const isCmd = (e) => {
2      const sig = "!";
3      let msg = e.message;
4      if ( msg.indexOf(sig) === 0 ) {
5          msg = msg.replace(sig, "");
6          e.message = msg;
7          e.cmd = msg.split(' ')[0];
8          e.isCmd = true;
9          e.content = msg.replace(e.cmd, "").trim();
10         return true;
11     }
12     return false;
13 };

```

Colored by Color Scripter

받은 문자열에서 시그니처(!) 가 가장 첫번째에 있으면 해당 문자열을 수정합니다.

cmd 변수에는 명령어.

isCmd 에는 해당 문자열이 명령어였다는 것을 알리는 Boolean 변수

content 에는 cmd 를 제외한 매개변수를 담게 됩니다.

객체의 자식요소를 수정하면 함수 호출시 메모리 복사되는 인자와 다르게  
직접 메모리를 수정하는 것이기 때문에 직접적으로 수정이 가능하죠.

사실, 이것은 제가 만든 플랫폼의 봇 코드를 카톡 봇 코드에 이식한 것입니다.

e 는 일종의 정보 객체라 볼 수 있죠.

그러면, e 를 만드는 코드를 보도록 합시다.

```
1 function response(room, message, sender, isGroupChat, replier, imageDB, packageName){
2   // create msg event
3   const msg = {
4     message: message,
5     sender: sender,
6     room: room,
7     isGroupChat: isGroupChat,
8     replier: replier,
9     imageDB: imageDB,
10    packageName: packageName,
11  };
12
13  // do something
14 }
15
```

Colored by Color Scripter

네. 그냥 response 에서 전달받은 것들을 전부 객체로 묶는 것일 뿐입니다.



동작은 하는데, 뭔가 이상해

이제 e.cmd 를 가지고, 명령어가 있는지 검사할 것입니다.  
명령어 관련 객체는 단순히 전역으로 선언하면 됩니다.

```
1 const commandList = {
2   "명령": "일단 어떠한 명령어",
3   "함수": (e) => {
4     return e.sender + "함수";
5   },
6   "랜덤": [
7     "랜덤1",
8     (e) => "랜덤2",
9   ],
10 };

```

명령 객체에 들어갈 수 있는 형 타입은 3개입니다.

- Function
- String
- Array

Function 은 실행시킨 결과값을 반환하고,  
String 은 그 자체를 반환하고,  
Array 를 구성하는 요소는 위 두개로 이루어져 있습니다.  
그것을 랜덤으로 반환합니다.

그럼, 위 명령어를 일괄적으로 하나의 함수에서 실행해, 문자열로 출력하는 코드가 필요합니다.

```
1  const runCmd = (cmd, e) => {
2    let str = "";
3    switch ( typeof cmd ) {
4      case "string": {
5        str = cmd;
6      } break;
7      case "function": {
8        str = cmd(e);
9      } break;
10     case "object": {
11       if ( Array.isArray(cmd) ) {
12         let result = Math.floor(Math.random() * cmd.length);
13         switch ( typeof cmd[result] ) {
14           case "string": {
15             str = cmd[result];
16           } break;
17           case "function": {
18             str = cmd[result](e);
19           }
20         }
21       }
22     } break;
23   }
24   return str;
25 };
26
```

Colored by Color Scriptor

이제 필요한 것은 다 갖춰진 것입니다.  
하나로 조합해서 동작이 되는 것을 확인해 보고 싶습니다.

```
1  const commandList = {
2    "명령": "일단 어떠한 명령어",
3    "함수": (e) => {
4      return e.sender + "함수";
5    },
6    "랜덤": [
7      "랜덤1",
8      (e) => "랜덤2",
9    ],
10 };
11
12 const isCmd = (e) => {
13   const sig = "!";
14   let msg = e.message;
15   if ( msg.indexOf(sig) === 0 ) {
16     msg = msg.replace(sig, "");
17     e.message = msg;
18     e.cmd = msg.split(' ')[0];
19     e.isCmd = true;
20     e.content = msg.replace(e.cmd, "").trim();
21   }
22   return true;
23 }
```

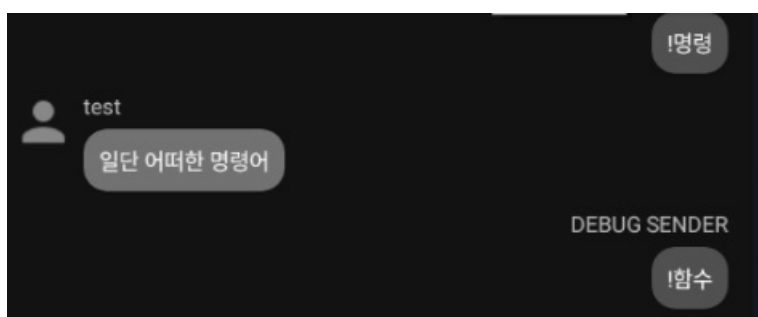
```

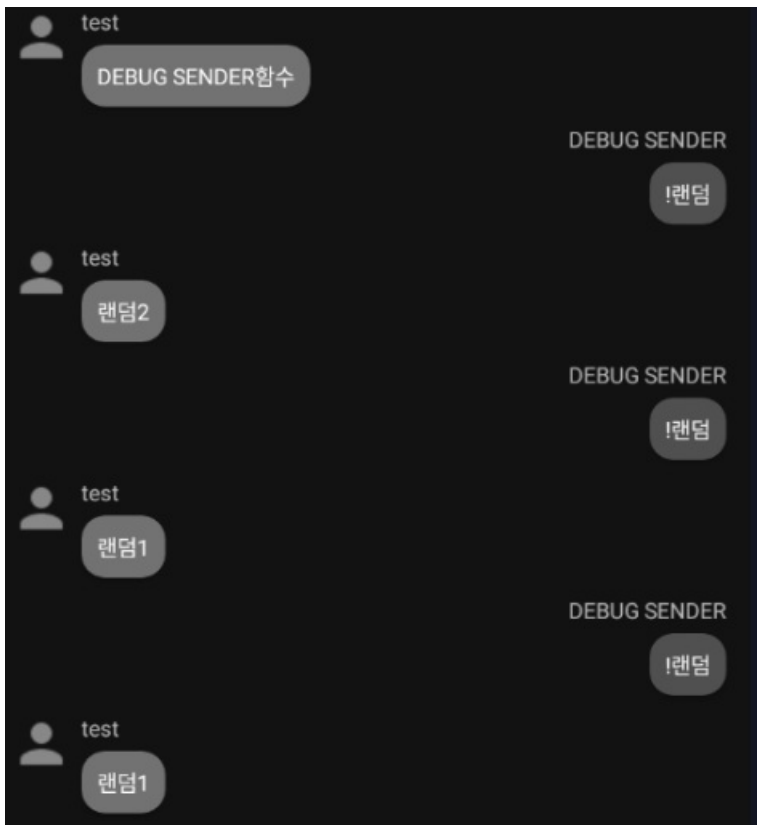
22     }
23     return false;
24 };
25
26
27 const runCmd = (cmd, e) => {
28     let str = "";
29     switch ( typeof cmd ) {
30         case "string": {
31             str = cmd;
32         } break;
33         case "function": {
34             str = cmd(e);
35         } break;
36         case "object": {
37             if ( Array.isArray(cmd) ) {
38                 let result = Math.floor(Math.random() * cmd.length);
39                 switch ( typeof cmd[result] ) {
40                     case "string": {
41                         str = cmd[result];
42                     } break;
43                     case "function": {
44                         str = cmd[result](e);
45                     }
46                 }
47             }
48             } break;
49     }
50     return str;
51 };
52
53
54 function response(room, message, sender, isGroupChat, replier, imageDB, packageName){
55     // create msg event
56     const msg = {
57         message: message,
58         sender: sender,
59         room: room,
60         isGroupChat: isGroupChat,
61         replier: replier,
62         imageDB: imageDB,
63         packageName: packageName,
64     };
65
66     if ( isCmd(msg) ) {
67         cmd = commandList[msg.cmd];
68         if ( cmd ) {
69             str = runCmd(cmd, msg);
70             replier.reply(str);
71         }
72     }
73 }
74

```

Colored by Color Scripter

위 코드를 돌려봅니다.





이제 명령어 수정이 용이해졌으며, response 함수는 깨끗합니다.

## □ WTF 정규식





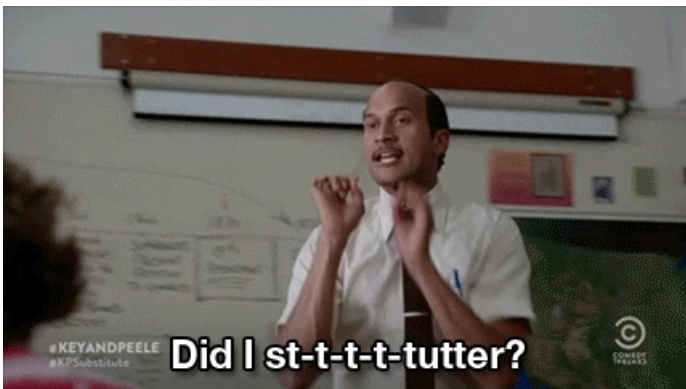
마치 자연어를 다루는 나의 모습

자연어를 처리하는 알고리즘 NLP (Natural Language Processing) 이라합니다.  
하지만 실제로 NLP 를 적용하여 하기엔 아직 너무 이른 감이 없지않아 있습니다.

진짜 NLP 는 나중에 다뤄보도록 하죠.

그래서 전 특정 단어의 패턴을 정규식으로 등록해두는 방식을 사용했습니다.  
정규식은 매우 짜증나지만 실제로 잘 쓰게 되면 이것만큼 편한 문자열 관리 기법이 없죠.

## ☞ 정규식을 이용해 KEY 추출



특정 단어의 패턴, 그리고 해당 단어의 품사(Part Of Speech) 를 KEY로 나열합니다.  
나열한 값은 이후에 사용할 예정입니다.

```
1 //////////////////////////////////////////////////
2 //                                의도 추론 (indent)                                //
3 //////////////////////////////////////////////////
4 /*
5     의도 추론이란, 입력 문장이 어떤 의도인지 분류하기 위한 기준입니다.
6     key: 의도타입 입니다.
7     indent: 의도를 가름하는 정규식입니다.
8     sub-indent: 하위 의도를 가리키는 오브젝트입니다.
9     각 배열의 아이템은 정규식이 될 수 있습니다.
10 */
11
12
13 const indentList = [
14     {
15         "key": "bot",
16         "indent": [ "부\\S*" ]
```



```

16 indent : [ "어떻게\\S", ],
17 "pos": "subject",
18 "sub-indent": [
19     {
20         "key": "how",
21         "pos": "adjective",
22         "indent": [ "어떠\\S", "어떡\\S", "어떻\\S" ],
23         "sub-indent": [
24             {
25                 "key": "make",
26                 "pos": "a-verb",
27                 "indent": [ "만드\\S+", "만들\\S+", "제작\\S+", ],
28                 "finish": true,
29             },
30         ],
31     },
32 ],
33 },
34 ];
35
36 const searchIndent = (msg, indent = indentList, deep = []) => {
37     for ( let id of indent ) {
38         if ( deep[deep.length - 1] && deep[deep.length - 1].key === "." ) {
39             break;
40         }
41
42         for ( let regxStr of id.indent ) {
43             if ( deep[deep.length - 1] && deep[deep.length - 1].key === "." ) {
44                 break;
45             }
46
47             const regx = new RegExp(regxStr);
48             const result = msg.match(regx);
49             if ( result ) {
50                 if ( typeof id['sub-indent'] === "object" ) {
51                     const len = deep.length;
52                     const t_res = searchIndent(msg.substr(result.index), id['sub-indent'], deep);
53                 }
54
55                 if ( id.finish ) {
56                     deep.push({ key: ".", val: "끝", pos: 'finish' });
57                 }
58
59                 if ( deep[deep.length - 1] && deep[deep.length - 1].key === "." ) {
60                     deep.unshift({ key: id.key, val: result[0], pos: id.pos });
61                 }
62             }
63         }
64     }
65     return deep;
66 };
67

```

Colored by Color Scriptor

indent가 정규식 배열 형태로, sub-indent의 깊이를 레벨이라고 하겠습니다.

누가 제 코딩 스타일을 보고 싶다 하셨는데, 이걸 보시면 될 것 같네요.

레벨진입은 재귀적으로 호출됩니다.

입력받은 메시지의 앞부터 봅니다.

1레벨: 안녕

2레벨: 하세요

이 때 "안녕 하세요"는 부합하지만, "하세요 안녕"은 부합하지 않습니다.

함수는 다음과 같이 불러집니다.

```
1 searchIndent("봇을 어떻게 만드나요?");
```

그러면 오브젝트 레벨에 따라 다음과 같은 결과가 나옵니다.

```
1  [  
2    {  
3      "key": "bot",  
4      "val": "봇을",  
5      "pos": "subject"  
6    },  
7    {  
8      "key": "how",  
9      "val": "어떻게",  
10     "pos": "adjective"  
11   },  
12   {  
13     "key": "how",  
14     "val": "제작하나요?",  
15     "pos": "a-verb"  
16   },  
17   {  
18     "key": ".",  
19     "val": "끝",  
20     "pos": "finish"  
21   }  
22 ]
```

## □ 키 조합으로 답변 추출



이제 위에서 나온 결과를 가지고 다시 답변을 추론해야 합니다.  
다행히 레벨 순차별로 오브젝트가 배열인 상태입니다.

단순히 반복만 돌리면 되지 않을까요?

```
1 //////////////////////////////////////  
2 //                                답변 추리 (answer)                                //  
3 //////////////////////////////////////  
4 /*
```

```

5  답변 처리란, 현재까지 주어진 데이터와 입력받은 문장 속의 의도를 취합하여 최종적인 대답을 도출해낼
6  의도 추론의 결과인 deep 배열에 있는 정보들을 기준으로 각 상황에 맞을법한 문장을 선택하여 조합합니
7  */
8
9
10
11 const answerList = [
12   {
13     "key": "bot",
14     "sub-answer": [
15       {
16         "key": "how",
17         "sub-answer": [
18           {
19             "key": "make",
20             "answer": "알아서 잘 만듭니다.",
21           },
22         ],
23       },
24     ],
25   },
26 ];
27
28 const searchAnswer = (deep, list = answerList, level = 0) => {
29   let answer = "";
30   const fidx = deep.findIndex(d => d.pos === "finish");
31   if ( fidx !== -1 ) {
32     deep.splice(fidx, 1);
33   }
34
35   for ( const item of deep ) {
36     const idx = list.findIndex((an) => an.key === item.key);
37     if ( idx === -1 ) continue;
38
39     const obj = list[idx];
40
41     if ( obj['answer'] ) {
42       answer += `\\n\\n${obj['answer']}`;
43     } else if ( typeof obj['sub-answer'] === "object" ) {
44       const deepDump = [];
45       for ( const d of deep ) {
46         if ( d !== item ) {
47           deepDump.push(d);
48         }
49       }
50
51       const tempAnswer = searchAnswer(deepDump, obj['sub-answer'], level+1);
52       if ( tempAnswer.trim() !== "" ) {
53         answer += tempAnswer;
54       }
55     }
56   }
57
58   if ( deep.length === 0 && level > 0 ) {
59     list.forEach(l => {
60       if ( l['answer'] ) {
61         answer += `\\n\\n${l['answer']}`;
62       } else if ( l['sub-answer'] ) {
63         answer += searchAnswer(deep, l['sub-answer'], level+1);
64       }
65     });
66   }
67
68   return answer;
69 };
70

```

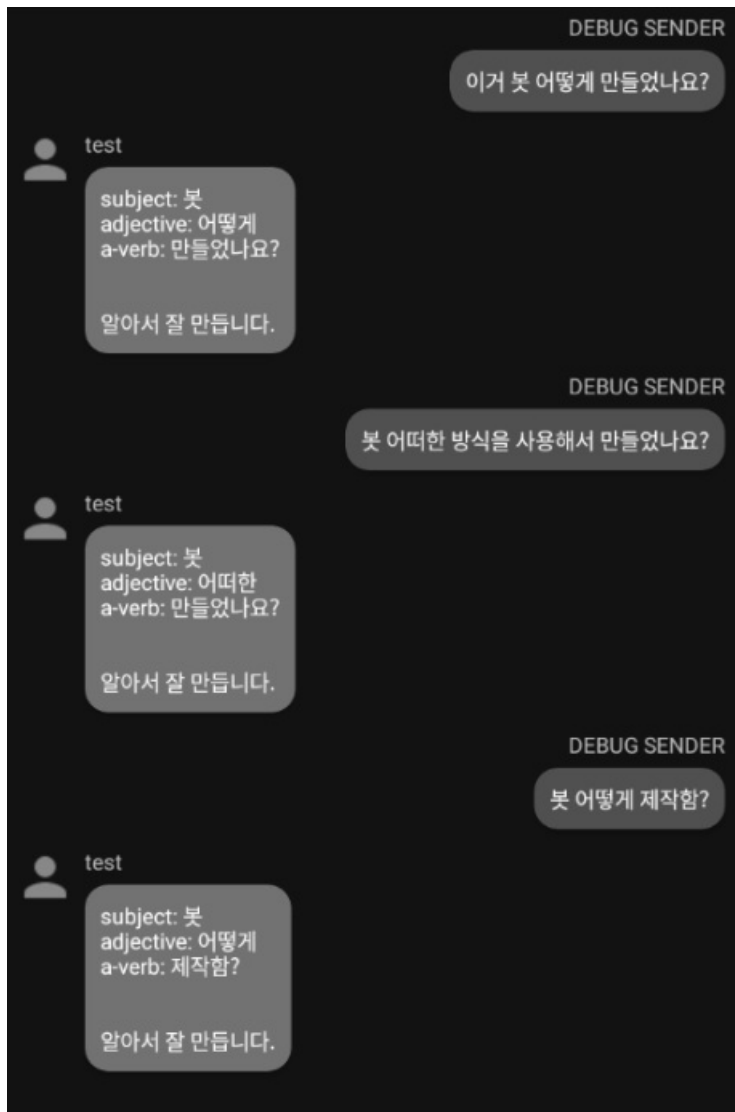
키 배열에 따른 레벨을 자식 객체로 만들어 총계형태를 띄웁니다.

위에서 나온 key 는 bot, how, make 였으니... "알아서 잘 만듭니다."가 답변이 됩니다.  
만약 2 레벨 이상에서 포함되는 하위 답변들은 한 번에 더해져 답변이 만들어집니다.

key 가 bot, how 여도 "알아서 잘 만듭니다." 라는 답변이 나온다는 얘기죠.

그럼, 이 코드가 잘 되는지 증명해 보아야 합니다.

for ... of 문법은 코볼소에서 문법체커가 에러를 뱉어서 바벨을 돌려 만들었습니다.



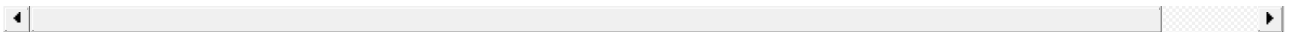
잘 됩니다.


이 방식을 잘 사용한다면, 학습도 더 멋지게 할 수 있을 것 같습니다.








오늘도 읽어주셔서 감사하니까 하트 주세요.




 [윤님의 게시글 더보기 >](#)

 좋아요 21  댓글 45


 공유 | 신고


댓글 등록순 최신순 


댓글알림 


 **MatSoGeum**  
ㄷ ㄷ


2020.06.07. 01:13 답글쓰기


 **윤 작성자**  
벌써 읽으시다니 ㄷ ㄷ  
2020.06.07. 01:14 답글쓰기


 **사과님**  
간결하게 잘 쓰시네요.  
2020.06.07. 01:50 답글쓰기


 **윤 작성자**  
칭찬 감사합니다  
2020.06.07. 01:59 답글쓰기


 **줄러**  
읽는 재미가 쏠쏠한 글들을 작성하시네요  
앞으로도 재밌는 글들 기대하고 있겠습니다  
2020.06.07. 01:53 답글쓰기


 **윤 작성자**  
앞으로도 흥미로운 짤들 기대해 주세요  
2020.06.07. 01:59 답글쓰기


 **줄러**


 윤 사실 글의 메인은 '짤'이었던 건가요  
2020.06.07. 02:00 답글쓰기



 윤 작성자  
줄려 짤 구하는 시간이 제일 많습니다  
2020.06.07. 02:00 답글쓰기



 AlphaDo  
보기만 할게요.  
2020.06.07. 02:04 답글쓰기


 윤 작성자  
봐주셔서 감사합니다.  
2020.06.07. 09:26 답글쓰기


 지민  
저도 저런 비슷한 원리로 구현해본적이 있는데, 훨씬 깔끔하네요.  
역시 현직은 못따라갑니다.  
2020.06.07. 02:14 답글쓰기


 윤 작성자  
원래 소스는 한 4년 전에 만들었다가 최근 리팩토링 한 거예요! 지민 님은 1년만 있어도 하실 듯  
2020.06.07. 09:27 답글쓰기


 클라이드  
  
2020.06.07. 02:46 답글쓰기


 윤 작성자  
  
2020.06.07. 09:27 답글쓰기

 도미 doami2005  
잘 보다가 의도추론 어쩌구에서 댓글창 내려왔어요 ㅎㅎ  
2020.06.07. 04:26 답글쓰기

 윤 작성자  
그냥 있어보이려고..... ㅎㅎ  
2020.06.07. 09:27 답글쓰기

 도미 doami2005  
윤 어려워서ㅜ  
2020.06.07. 09:35 답글쓰기

 a8M9uQ2WRV  
새글알림 일하네요  
2020.06.07. 08:48 답글쓰기

 윤 작성자  
부지런하네요. 저도 일해야 하는데  
2020.06.07. 09:27 답글쓰기



**jomin398**

저랑 같은길을 걸으시군요 ㅎㅎ  
깃헙으로 소스공유는 하실건가요?

자동학습을 저기있는 분리소스랑 합쳐서 NLP로....

2020.06.07. 09:52 답글쓰기



**윤 작성자**

에이, 제 코드 공유한다고 누가 봅니까. 부끄러워서 못 합니다.

2020.06.07. 09:54 답글쓰기



**jomin398**

윤 제가 보조 ㅎㅎ

날씨 데이터를 자연스럽게... ㅎㅎ

2020.06.07. 10:03 답글쓰기



**도미 doami2005**

**jomin398** 정말이지 좋은 온도예여! ㅋㅋㅋㅋㅋㅋㅋㅋ

2020.06.07. 10:03 답글쓰기



**jomin398**

**도미 doami2005** ㅋㅋ (사심이 가득)

2020.06.07. 10:04 답글쓰기



**nolbo**

유익한 글이 많네요 시간날때 전체 글 읽어보겠습니다

2020.06.07. 10:51 답글쓰기



**윤 작성자**

감사합니다

2020.06.07. 10:54 답글쓰기



**정연**

핵심만 뽑아주시네요  
잘봤습니다

2020.06.07. 14:45 답글쓰기



**윤 작성자**

읽어주셔서 감사합니다

2020.06.07. 14:47 답글쓰기



**재승**

ㅌ나 ㄷㄷ

2020.06.07. 14:52 답글쓰기



**qbvkde**

?? 이분이 왜 아직 등급이 챗봇신입생이지  
완전 강의들 내용 다 좋은데

2020.06.07. 15:00 답글쓰기



**윤 작성자**

제가 닉네임 뒤에 붙는 새싹을 포기 못해서 등급 신청을 안 합니다

2020.06.07. 15:02 답글쓰기



**qbvkde**

윤 알ㅋㅋㅋㅋㅋㅋㅋㅋ

2020.06.07. 15:03 답글쓰기



**AlphaDo**

윤 새싹 없앨게요

2020.06.07. 19:07 답글쓰기



윤 작성자

AlphaDo 아이고 아이고 선생님 전 가진게 없어서 이것마저 뺏어가 버리시면...

2020.06.07. 19:08 답글쓰기



AlphaDo

윤 찼찼

2020.06.07. 19:09 답글쓰기



윤소희

호롤로

2020.06.08. 01:22 답글쓰기



야옹

잘쓰신다!

2020.06.10. 23:33 답글쓰기



윤 작성자

감사합니다

2020.06.10. 23:34 답글쓰기



허니

와드

2020.06.12. 10:01 답글쓰기



윤 작성자

바..... 박혀버렸

2020.06.12. 10:01 답글쓰기



허니

윤 게임 길드특방을 위해 봇을 만들고 있습니다. 마침 자연어 처리에 대해 고민 중이었는데, 도움이 될 거 같아서 와드 박았습니다. 좋은 글 감사합니다!

2020.06.12. 10:13 답글쓰기



NenkaLab

잘 읽었습니다?

2020.06.19. 12:01 답글쓰기



윤 작성자

잘 읽어주셔서 감사합니다

2020.06.19. 12:15 답글쓰기



jomin398

본게시글의 내용을 모듈화를 부탁....

2020.07.15. 23:15 답글쓰기



윤 작성자

그냥 복사하셔서 쓰기만 해도 돼요

2020.07.15. 23:16 답글쓰기

Hibot

댓글을 남겨보세요



등록

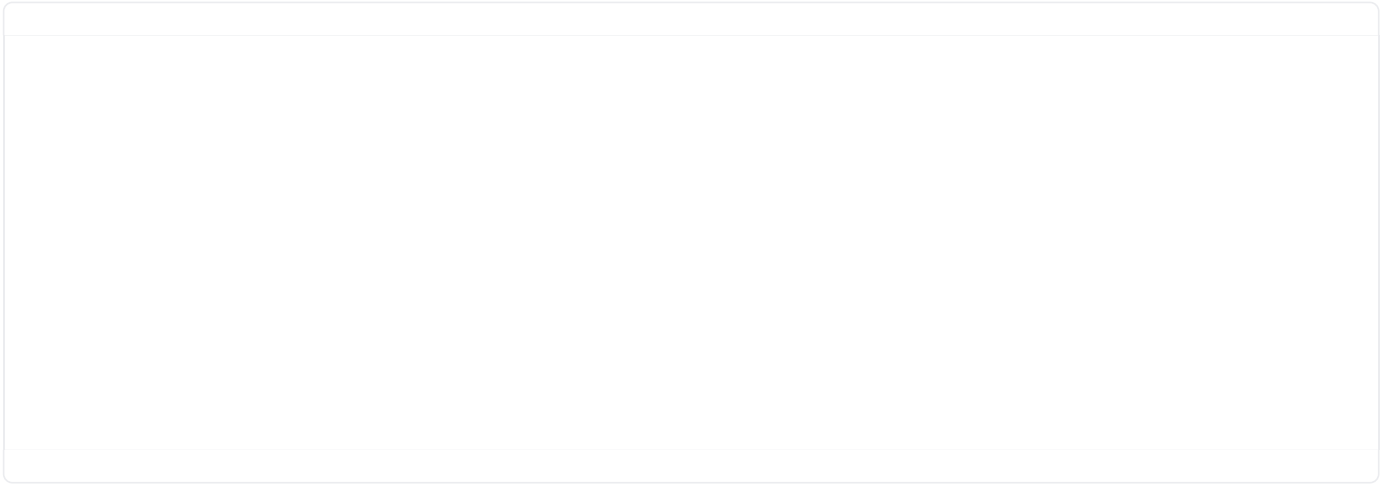
글쓰기

답글

목록

▲ TOP





'자유 게시판' 게시판 글

이 게시판 새글 구독하기 ☐

[카링] 입퇴장 🙋 [12]	Project	2020.06.07.
카톡 랜덤퀴즈 봇 같은거 어떻게 만들어요? 문제 답 맞춘사람 틀린사람 [5]	묵	2020.06.07.
[스압] 명령어 구분과 자연어 분석을 동시에 하기 ( 내가 봇을 쓰는 방법 ) 🙋 [45]	윤	2020.06.07.
Api 키 [5]	imroy09	2020.06.07.
이발?... 🙋 [59]	hoy	2020.06.06.

1 2 3 전체보기

이 카페 인기글

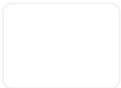
안녕하십니까.

JSR  
♡0 💬8



학교 급식 정보 Remake

도만 jomin2005  
자동화음소스  
♡0 💬7  
tomohong  
♡0 💬9



메신저R카톡방 들어와 채팅치면 환영해주는 소스



카톡봇 사람들은 꼭봐야됨

자만 OtakoidTony  
기본 개념에 대한 강좌 | 조건문과 반복문  
♡4 💬8  
♡0 💬6

성빈님

린  
♡0 💬1

버튼 누르면 intent가 함수를 실행하게하려면

jomin398  
♡0 💬13

여러분은 어케 카톡봇 배우게 되었나요?

사로로  
♡0 💬25

1 2 3 4 5