Welcome

챗봇담당자 김대리님을 위한 단비Ai 튜토리얼 3편

챗봇, 개발자의 영역 이해하기

버전 1.8 / 2020.4.3. / 주식회사 단비아이엔씨

챗봇, 일단 만들어보기

0.챗봇 사례 살펴보기
1. 나의 첫 챗봇 만들기
2. FAQ챗봇
3. 회사소개 챗봇
4. 채널 연결

챗봇, 차근 차근 배우기

5. 회사소개 챗봇 살펴보기
6. 더 똑똑하게!
7. 대화 의도 예문 학습
8. 고유용어? 엔티티?
9. 중의적인 표현
10. 대화흐름 차근 차근

챗봇, 개발자의 영역 이해하기

11. 시스템 연동하기
12. Live Chat스타일 챗봇
13. Event로 시작하는 대화흐름

챗봇, 운영/참고자료

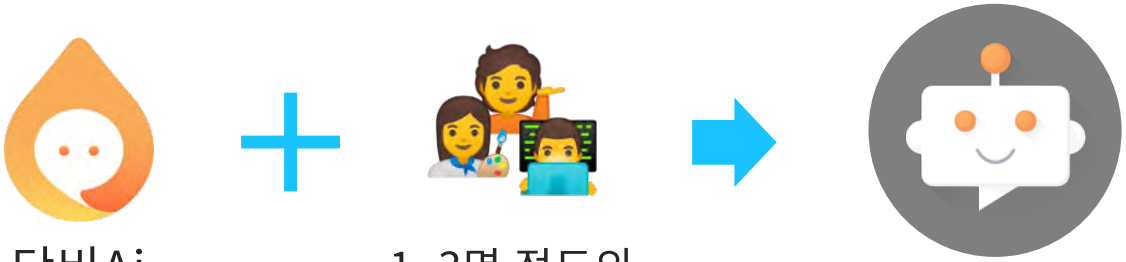
14. 운영 노하우
15. 보너스!
16. 부록

챗봇담당자 여러분, 안녕하세요?
교육을 담당할 '후추'에요~!

이 자료를 열었다면, 챗봇을 만들어 보겠다는 마음을 먹으신 것이겠죠? 멋진 생각입니다.
챗봇은 살아있는 것처럼 말을 알아듣고 대답하고, 요청을 처리해주는 재미있는 디지털 서비스입니다.
"말하는 존재"를 만드는 것은 매우 흥미로운 경험입니다.
더 나아가 사람들이 유용하게 활용할 수 있는 디지털 서비스를 전문적인 프로그래밍 지식 없이
사용자들이 쓸 수 있게 해준다는 점은 단비Ai의 가장 멋진 특징입니다.
이 자료는 단비Ai를 이용해 챗봇을 만드는 챗봇담당자를 위한 교육자료입니다. 최초에는 실습강의에
사용되는 자료였지만, 강의 없이 자료만 보고 독학할 수 있도록 업데이트 되고 있습니다.

1, 2편을 아직 보지 않으신 분은 1, 2편부터 보세요!
문서는 https://bit.ly/345gkGg 튜토리얼에서 모두 받으실 수 있습니다.
※ 영상자료(https://www.youtube.com/channel/UCoRrVH_eUMM4x_zwsOCpSqQ)를 통해 더 많은 정보를 얻으실 수 있어요.

단비Ai는 1~3명 정도의 소규모 팀이 챗봇이나 Siri같은 대화Ai 서비스를 만들 수 있게 해주는 서비스입니다.



가장 쉽고 강력한 챗봇 빌더

#UX #LG CNS #사내벤처 #대화AI #챗봇

LG CNS의 사내벤처로 시작한 단비Ai는 천만명 이상이 사용하는 LG U+ 고객센터 챗봇을 비롯한 다양한 챗봇을 싹틔우고 키워냈어요. 큰 기업이나 기관뿐만 아니라 소규모 팀에서도 대화Ai를 만들고 운영할 수 있는 '챗봇시대'를 열기 위해 오늘도 노력한답니다!

11. 시스템 연동하기
API노드와 Function노드

API, Function. 여기는 개발자의 영역

개발자가 아니더라도, 개발자에게 무엇을 요청해야 하는지 이해를 하는 것이 필요합니다.

오늘 날씨를 물어보면, 적절히 대답해주는 챗봇을 만들어 보도록 하겠습니다.

[그룹 > 그룹챗봇 > API관리] 메뉴로 이동하고 아래 내용을 입력합니다.

1. API 이름 : 날씨API
2. API 메서드 :
   [GET] http://api.openweathermap.org/data/2.5/weather
3. Header :
   > Content-type : JSON(application/json)
3. Query Parameter 5가지
   > appid : 6817a27f6034bec42c2a3b3db6578020 (서비스 제공처에서 발급한 키)
   > lang : kr (언어)
   > mode : json (모드)
   > q : seoul,kr (지역)
   > units : C (단위/섭씨)
4. 응답 > main.temp : 온도

※ 날씨API는 무료가 아닙니다. 예제는 교육을 위해 단비AI에서 발급받은 API키입니다. (사용량이 많으면, 서비스가 중단될 지 몰라요.)
만들고 있는 챗봇이 실습이 아닌 실제 사용자들에게 제공되어야 한다면, Openweathermap에 가입하여 키를 발급받으십시오.
이 내용에 대한 의미를 알 수 없다면, [개발자] 동료나 친구에게 도움을 요청하세요.

차근 차근 입력하세요.

아래와 같이 말이죠.

API 메서드

GET http://api.openweathermap.org/data/2.5/weather

Header + Header 추가

필수	Name	설명	Test Value
	Content-Type		JSON(application/json)

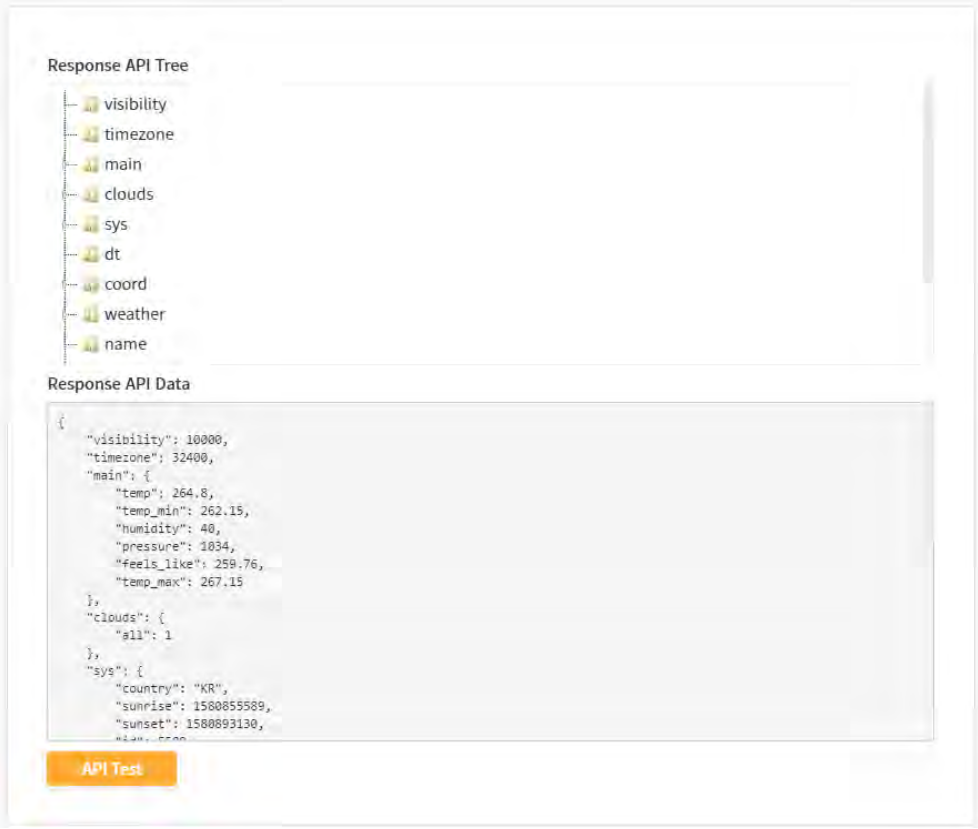
Query Parameter + Query Parameter 추가

필수	Name	설명	Test Value
✓	appid	서비스 제공처에서 발	6817a27f6034bec42c2a3b3db6578020
✓	lang	언어	kr
✓	mode	모드	json
✓	q	지역	seoul,kr
✓	units	단위/섭씨	C

API 테스트를 해보겠습니다.

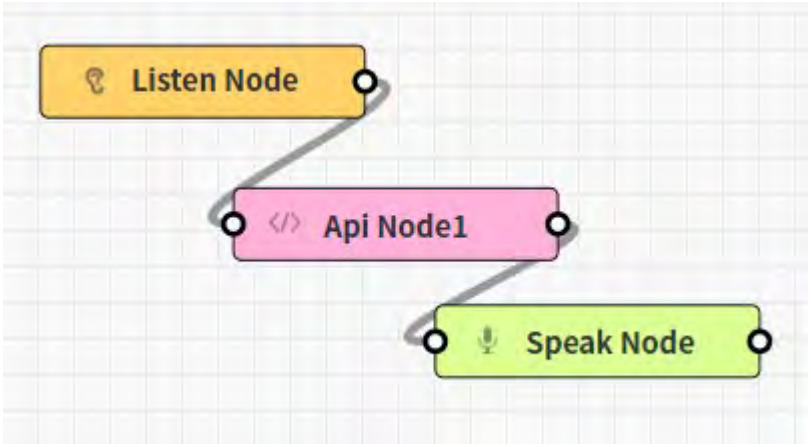
[API Test]를 클릭하면, 아래와 같이 API호출 결과가 나타납니다.

temp는 날씨를 의미하고 humidity는 습도를 의미합니다. 살펴보시죠.

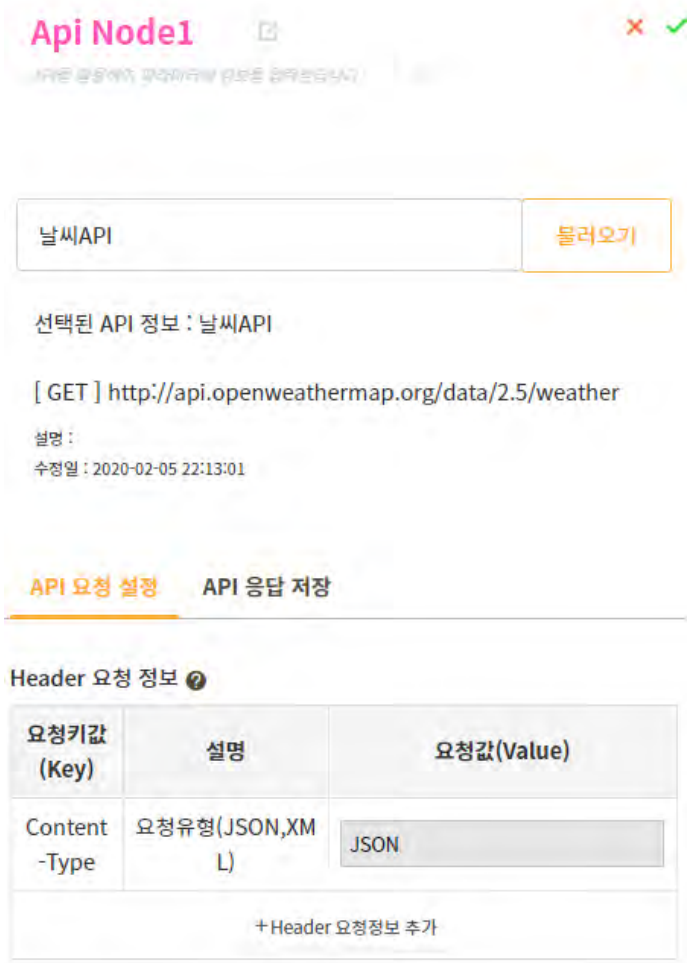


대화의도에 '날씨'를 만들고 대화흐름을 하나 연결합니다.

'날씨'라고 입력하면 '날씨문의'로 이해하는 대화의도를 하나 생성하고, 대화흐름을 하나 만듭니다. 그리고 나서 '날씨'라는 파라미터를 하나 추가합니다. 엔티티는 sys.any로 설정하겠습니다. 그리고 대화흐름은 Listen-API-Speak구조로 만듭니다.



이제 API노드를 설정합니다.



날씨API를 선택하고 [불러오기]를 클릭합니다.

앞서 API관리 메뉴에서 등록한 API가 불러와집니다.

Query Parameter가 비어있군요!

Query Parameter 요청 정보

요청키값 (Key)	설명	요청값(Value)
appid *	서비스 제공처에서 발급한 키	필수값 입니다.
lang *	언어	필수값 입니다.
mode *	모드	필수값 입니다.
q *	지역	필수값 입니다.
units *	단위/섭씨	필수값 입니다.

+Query Parameter 요청정보 추가

API관리에서 Query Parameter를 등록했는데, 왜 또 입력해야 하나요?

API가 정상적으로 동작하는지 테스트 하기 위한 것이었고, 대화흐름상에서는 '지역'이 대화 내용에 따라 바뀔 수 있습니다. 예를 들어 Slot노드를 추가하고, [지역]을 물어보고 'seoul,kr' 부분에 #{지역}을 설정하면, 지역을 다르게 안내할 수 있습니다.

Query Parameter 요청 정보

요청키값 (Key)	설명	요청값(Value)
appid *	서비스 제공처에서 발급한 키	6817a27f6034bec42c2a3b3db6578
lang *	언어	kr
mode *	모드	json
q *	지역	seoul,kr
units *	단위/섭씨	C

+Query Parameter 요청정보 추가

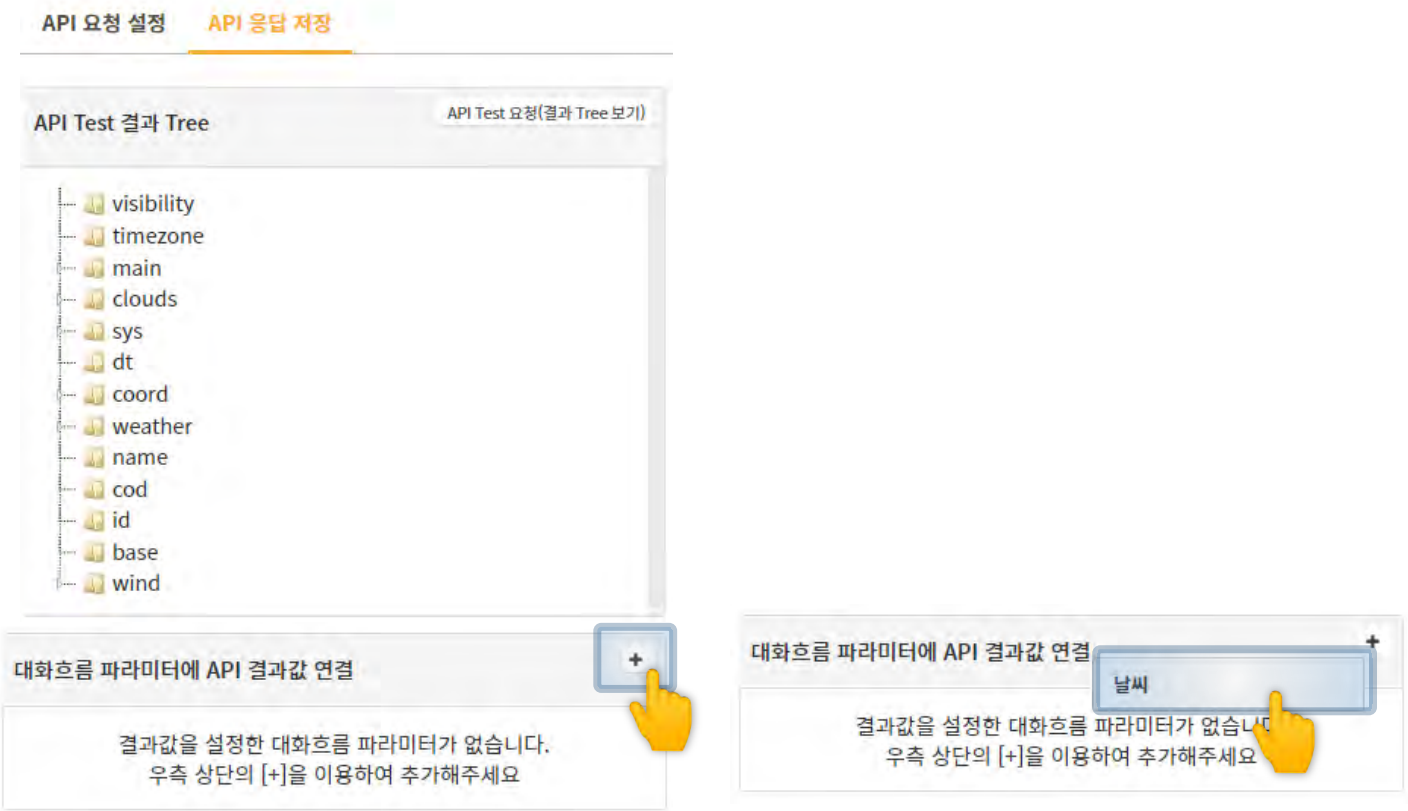
Query Parameter 요청 정보

요청키값 (Key)	설명	요청값(Value)
appid *	서비스 제공처에서 발급한 키	6817a27f6034bec42c2a3b3db6578
lang *	언어	kr
mode *	모드	json
q *	지역	#{지역}
units *	단위/섭씨	C

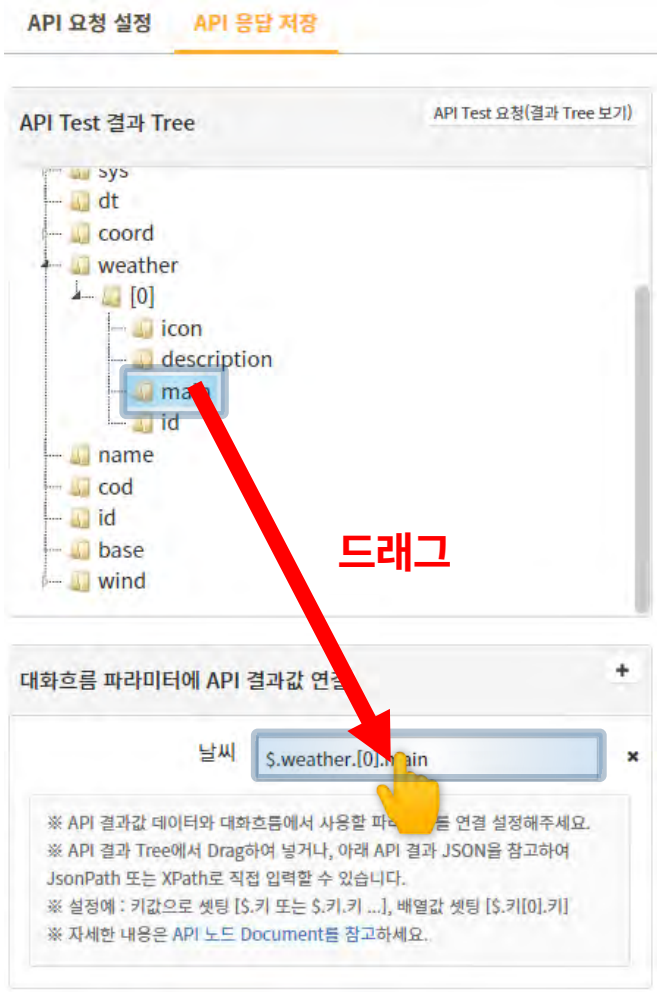
+Query Parameter 요청정보 추가

API결과를 파라메터에 담아서 답변에 활용하기

[API응답저장] 탭을 클릭하고 [API Test 요청]을 클릭합니다. API호출 결과가 Tree형태로 표시됩니다. Tree 아래에 [+]버튼을 클릭합니다. 그리고 [날씨]를 클릭합니다.



API결과를 파라메터에 담아서 답변에 활용하기

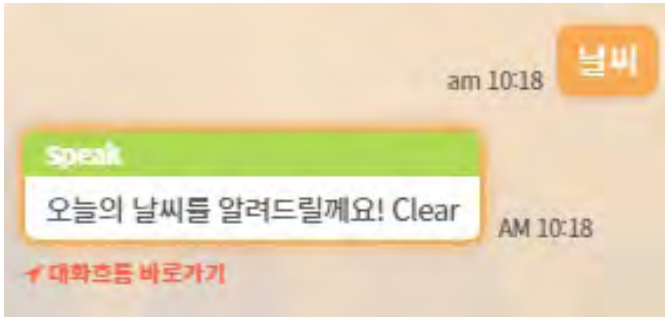


Tree에서 weather > [0] > main 을 드래그하여 날씨 옆 공간에 놓습니다.

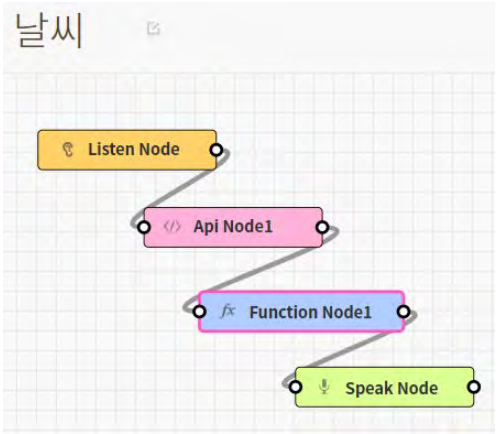
$.weather.[0].main 이라고 표시되는 것을 알 수 있습니다. JSON데이터에 대해 익숙하신 분들을 위해 직접 타이핑 또는 Copy&Paste할 수 있게도 해놓았습니다.

API결과를 파라메터에 담아서 답변에 활용하기

Speak노드에 #{날씨}를 포함해서 날씨를 알려주는 멘트를 작성합니다.



‘Clear’라는 정보를 ‘맑음’으로, ‘Clouds’라는 정보를 ‘구름많음’으로 나오게 Function노드를 이용해 데이터를 변경합니다.

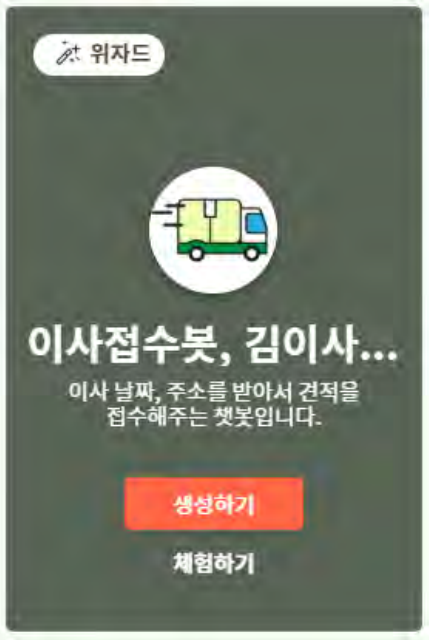


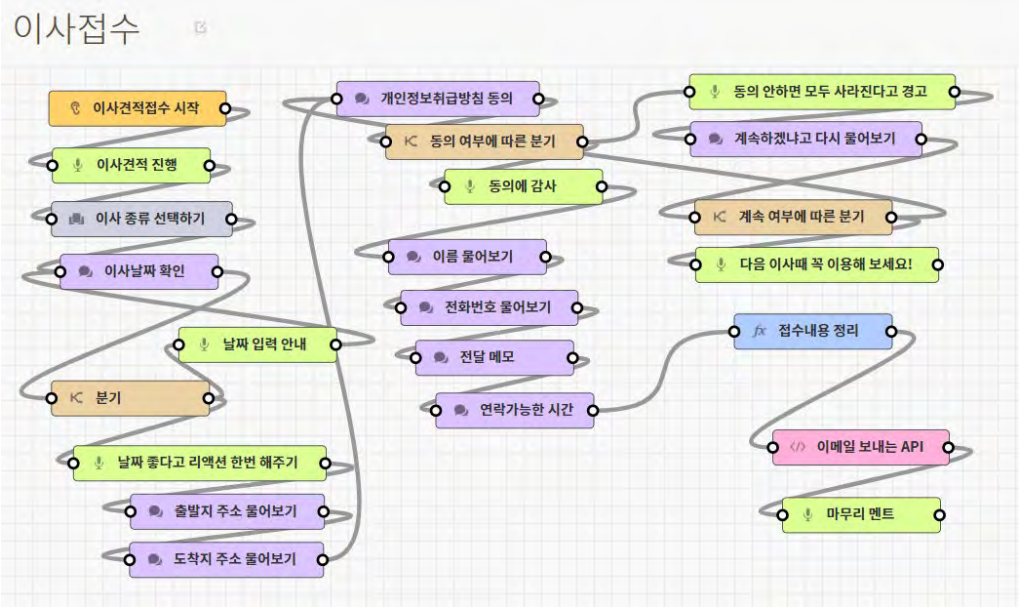
API노드와 Speak노드 사이에 Function노드를 연결하고 Function노드 안에 아래와 같이 코드를 작성합니다.

```
if (날씨=='Clear'){
   날씨='맑음';
}else if (날씨=='Clouds'){
   날씨='구름많음';
}
```

단비Ai에서 제공하는 API를 활용하면, 개발자가 없더라도 간단한 접수업무를 챗봇에게 맡길 수 있어요.

위자드 챗봇 중에 [이사접수봇, 김이사]를 가져오기 해보세요. 단비Ai에서 제공하는 이메일 보내는 API를 통해서 챗봇이 접수받은 내용을 이메일로 전달해줍니다. 응용해서 구글 SpreadSheet에 한 줄씩 추가해주거나 회사시스템과 연계할 수도 있습니다.





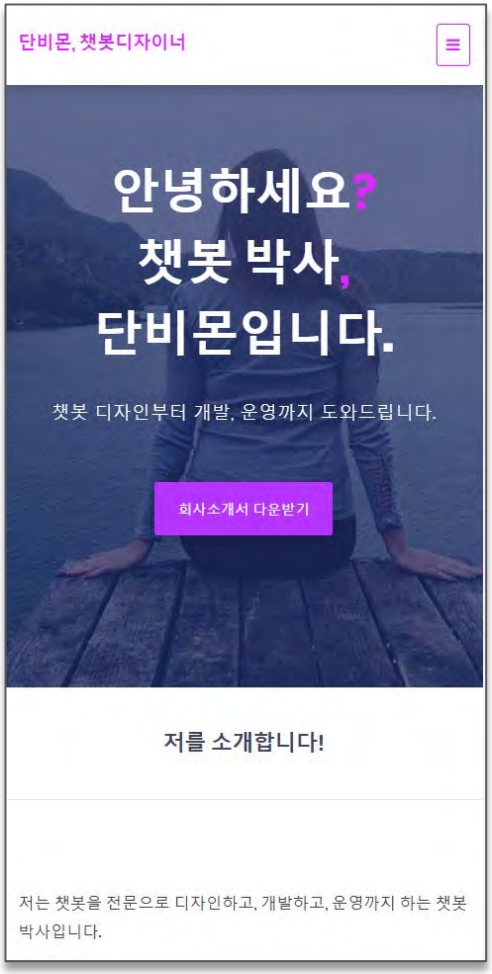
챗봇을 만드는데 필요한 역량과 구현 가능한 챗봇 수준

챗봇 수준	난이도	개발역량 수준
단순 문의 응답을 진행하는 FAQ챗봇	브론즈	불필요
멀티턴 대화를 통한 세밀한 답변을 해주는 문의 응답 챗봇	실버	불필요
단비API를 이용한 간단한 접수 챗봇	골드	API원리 이해필요
시스템 연동을 통한 실 정보 표시	플래티넘	개발자 필요
사용자 맞춤 정보 제공	그랜드마스터	개발자 필요

12. Live Chat스타일 챗봇 (Frogue Embedding)

단비홈페이지의 단비봇이나 파브르에게 '실습자료'라고 입력하면, 실습자료 압축파일을 다운받으실 수 있습니다.

1. 실습자료에 [챗봇박사홈페이지] 폴더를 열어보세요.
2. index.html을 더블 클릭해보세요.
3. 오른쪽과 같은 홈페이지를 보실 수 있습니다.



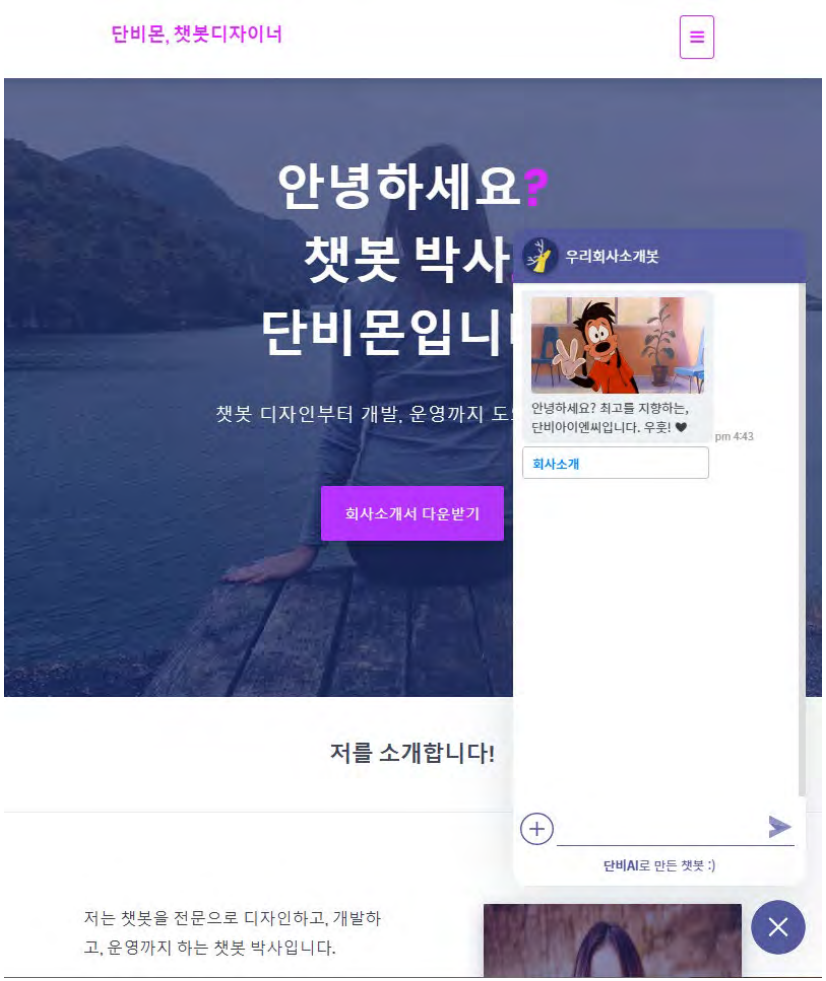
챗봇 > 챗봇제작 > 채널연결 > Frogue연결 설정에 Embed 코드를 복사합니다. (Ctrl + C)

Index.html을 [메모장]등의 Text편집기를 이용해 열고,
</body>바로 앞에 붙여넣기 합니다. (Ctrl+V)그리고 저장합니다.

```
        <!--Custom JS-->
        <script src="assets/js/custom.js"></script>

<div id="frogue-container" class="position-right-bottom"
        data-chatbot="803ebc0a-20d8-436a-be40-b7d15c965e99"
        data-user="사용자ID"
        data-init-key="value"
        ></div>
<!-- data-init-식별키=값 으로 셋팅하면 챗플로우에 파라미터와 연동가능. 식별키는 소문자만 가능 -->
<script>
(function(d, s, id){
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) {return;}
    js = d.createElement(s); js.id = id;
    js.src = "https:₩/₩/danbee.ai/js/plugins/frogue-embed/frogue-embed.min.js";
    fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'frogue-embed'));
</script>

    </body>

</html>
```

Ctrl+V

처음에 열었던 웹페이지를 새로고침합니다. (F5 또는 Ctrl+R)
짠~! 홈페이지에 챗봇 등장!



STEP #13

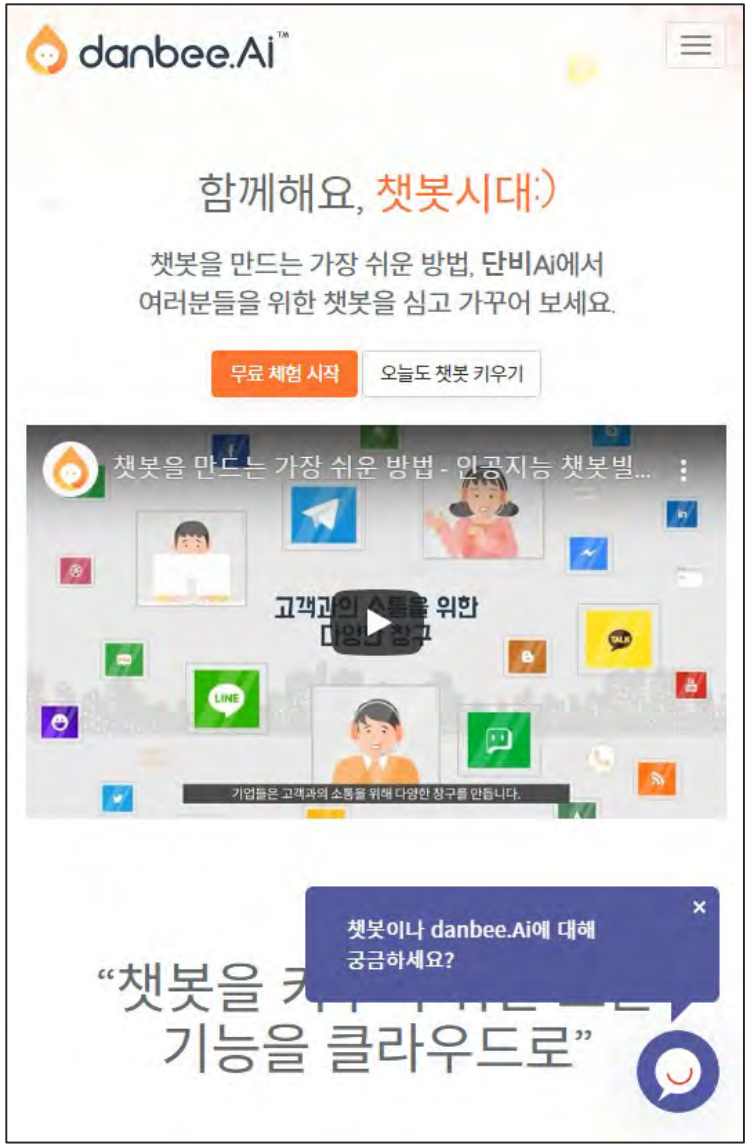
13. Event로 시작하는 대화흐름

대화를 Event로 시작한다는 것의 의미.

챗봇이 사용자가 “발화”를 했을 때만 말을 하는 것은 아닙니다.

- 홈페이지에 있는 챗봇은 첫 페이지에서 먼저 반갑게 방문자에게 인사할 수 있습니다.
- 업무시스템에 있는 챗봇은 사내 공지사항이 있을 때 먼저 알려 줄 수 있습니다.
- 특정 화면에 갔을 때, 참고할 수 있는 정보를 제공할 수 도 있습니다.
  예를 들어 쇼핑몰의 신상 원피스 상품의 상세화면으로 들어가면,
  “이 상품은 이번 여름 신상이에요!” 라고 말할 수 있는 것이죠.
- 또는 어떤 버튼을 클릭했을 때 챗봇이 반응 할 수도 있습니다.

먼저 웹사이트에 도착했을 때, Live Chat 아이콘에서 먼저 인사를 건네는 대화흐름을 적용해 볼께요.



오른쪽 그림과 같이 우측하단에 채팅을 할 수 있을 것으로 보이는 아이콘을 Live Chat 아이콘이라고 합니다.

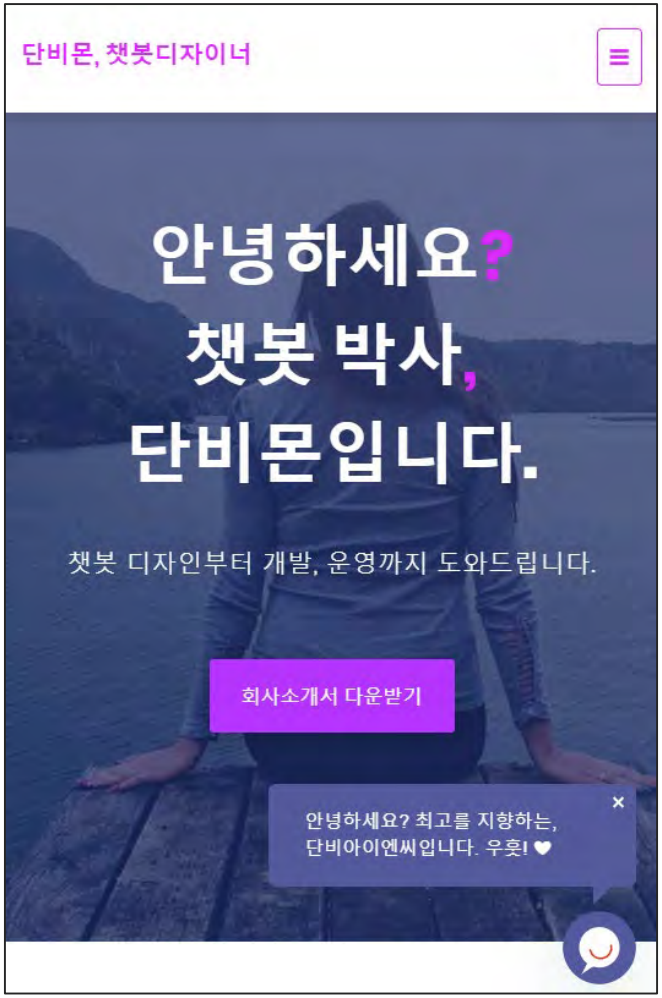
Index.html을 [메모장]등의 Text편집기를 이용해 열고,
</body>바로 앞에 아래 내용을 붙여넣기 합니다. (Ctrl+V) 그리고 저장.

```
<script>
if (document.addEventListener) {
   // Mozilla, Opera, Webkit
    document.addEventListener("DOMContentLoaded", function () {
        document.removeEventListener("DOMContentLoaded", arguments.callee, false);
        htmlReady();
    }, false);
} else if (document.attachEvent) {
   // Internet Explorer
   document.attachEvent("onreadystatechange", function () {
        if (document.readyState === "complete") {
            document.detachEvent("onreadystatechange", arguments.callee);
            htmlReady();
        }
    });
}
var htmlReadyTimer = null;
   //DOM이 모두 로드 되었을 때
function htmlReady () {
   htmlReadyTimer = setTimeout(function() {
        if (frogueReadyFlag) {
            froguePushEvent('landingEvent', {event_name: '', event_value1: '', event_value2: ''});
            htmlReadyTimer = null;
        } else {
            htmlReady();
        }
    }, 500);
}
</script>
```

Event로 시작하는 대화흐름을 만들어 보아요.

1. [챗봇 > 챗봇제작 > 대화흐름]에서 [+대화흐름 생성]을 클릭합니다.
2. “랜딩페이지도착”이라는 이름으로 만들게요.
3. 대화흐름 캔버스에서 [이벤트플로우로 전환]을 활성화 합니다. 이벤트 명에는 “landingEvent”라고 입력해요.
4. 이렇게 설정하면, Listen노드는 사용자의 자연어 입력이 아닌 웹사이트의 “화면로딩끝”이라는 이벤트로 시작하게 됩니다.
5. 이미 만들어져있는 Welcome 메시지에 JUMP노드로 연결합니다.

적용이 되고 나면, 그림과 같이 먼저 말을 걸어요!



Event를 발생시키는 버튼을 만들어 보아요.

event플로우적용.txt 파일에 있는 내용 중 아래 내용을 복사(Ctrl+C)하여,

index.html 내에 <!--이벤트 버튼 시작 --> 부분에 붙여넣습니다.(Ctrl+V)

```
<!--이벤트 버튼 시작 -->
<a href="javascript:click('info')">소개</a>
<a href="javascript:click('map')">위치</a>
<a href="javascript:click('contact')">연락처</a>
<!--/이벤트 버튼 끝 -->
```

버튼을 클릭하면 챗봇에게 Event를 전달하는 스크립트도 추가합니다.

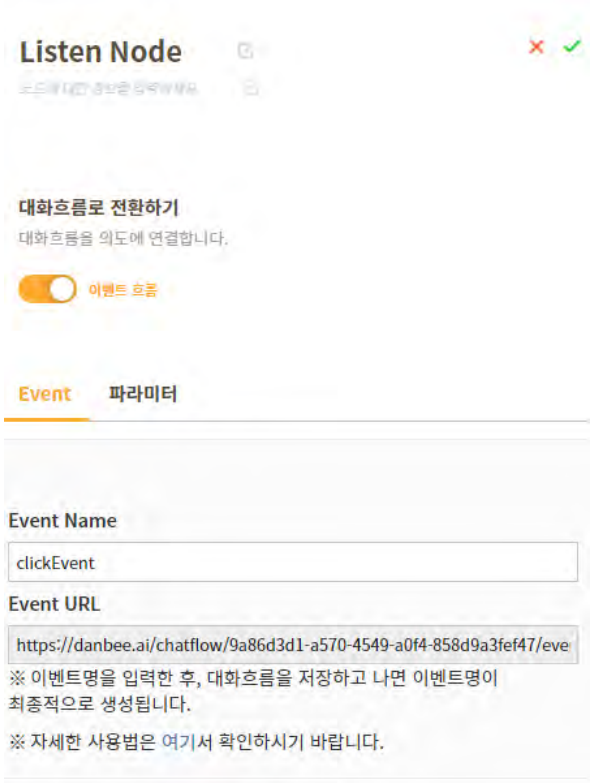
event플로우적용.txt 파일에 있는 내용 중 아래 내용을 복사(Ctrl+C)하여,

index.html 내에 (이벤트 시작버튼 바로 아래 또는 </body> 바로 윗줄에) 붙여넣습니다.(Ctrl+V) 그리고 저장합니다.

```
<!-- Event 적용 시작 -->
<script>
function click(v) {
   if (frogueReadyFlag) {
      if (v === 'info') {
         froguePushEvent('clickEvent', {'name':'info'});
       } else if (v === 'map') {
          froguePushEvent('clickEvent', {'name':'map'});
       } else if (v === 'contact') {
          froguePushEvent('clickEvent', {'name':'contact'});
       }
    }
    return false;
   }
</script>
<!-- /Event 적용 끝 -->
```

대화흐름에서 Event플로우를 만들어요.

이제 단비Ai에 대화흐름을 하나 생성합니다. 대화흐름 이름을 “이벤트플로우”라고 짓고, 아래처럼 대화흐름을 구성합니다. 그리고 Listen노드에서 [이벤트 흐름으로 전환] 하고 나서 “clickEvent”라는 Event Name을 부여하고 저장합니다.



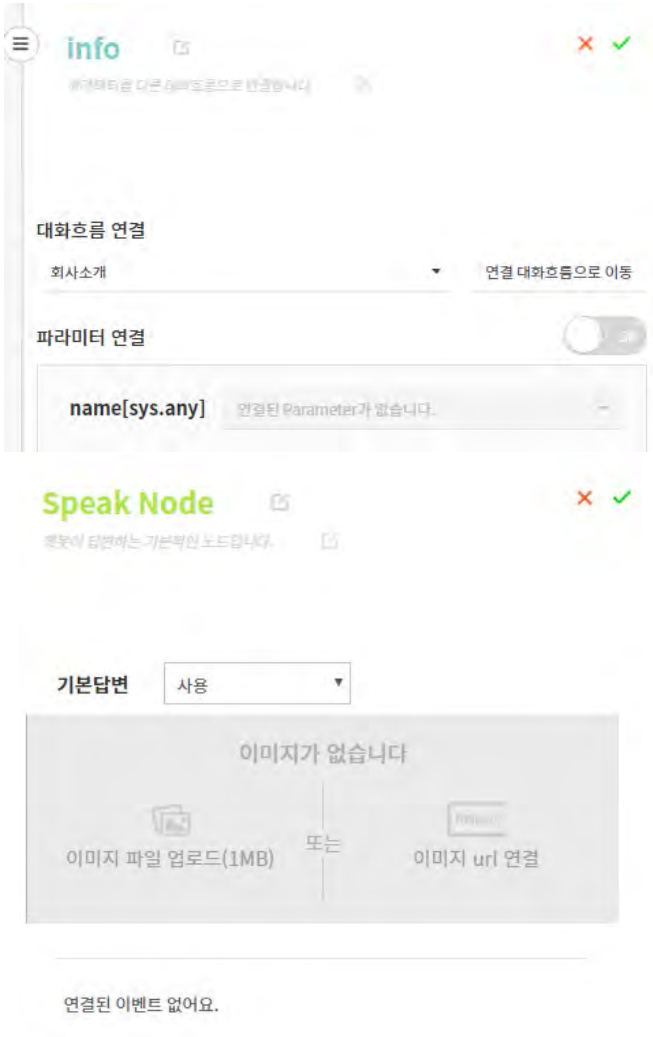
홈페이지가 전달하는 정보를 받는 대화흐름내 파라미터를 설정합니다.

파라미터에 #{name}을 추가합니다. 엔티티는 sys.any로 설정해요.

Name 에 전달받은 정보에 따라 다른 흐름을 타도록 분기합니다.

대화 흐름이 흘러가는 분기 조건을 그림과 같이 설정합니다. JUMP노드에 이름을 붙여 놓으면, 작업이 편합니다.

Jump노드에서는 이미 만들져 있는 노드를 연결하겠습니다.



Jump노드에는

'info'는 [회사소개]로

'map'은 [오시는 길]로

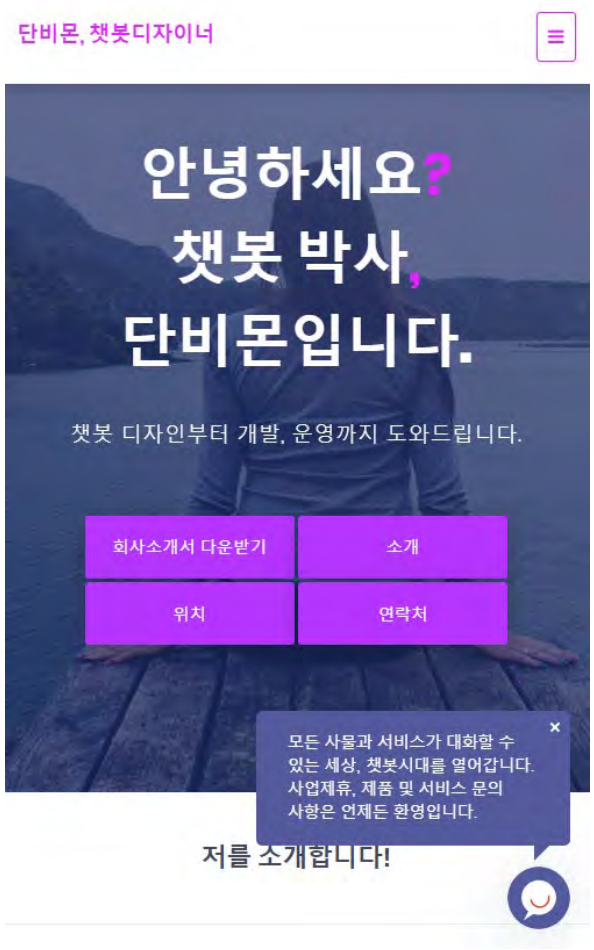
'contact'은 [연락처]에 각각 연결합니다.

Speak노드에는

name으로 전달되는 파라미터가 없을 경우

표시할 메시지를 설정합니다.

[버튼을 클릭]하면, 챗봇이 반응합니다.
[버튼을 클릭] 대신 다른 Event도 가능합니다. 생각해보아요.



소개, 위치, 연락처 버튼을 클릭하면 챗봇이 먼저 말을 시작하는 것을 볼 수 있습니다.

교육과정에서는 Click이라는 이벤트를 이용했지만, 아래와 같은 이벤트에도 적용할 수 있습니다.

어떤 이미지에 마우스 오버,
어떤 게시물에 마우스 오버,
스크롤을 맨 밑으로 내렸을 때,
어떤 입력필드를 클릭해서 타이핑 하려는 순간 등

다음편에서 만나요!

끝!

챗봇, 일단 만들어보기

0.챗봇 사례 살펴보기
1. 나의 첫 챗봇 만들기
2. FAQ챗봇
3. 회사소개 챗봇
4. 채널 연결

끝!

챗봇, 차근 차근 배우기

5. 회사소개 챗봇 살펴보기
6. 더 똑똑하게!
7. 대화 의도 예문 학습
8. 고유용어? 엔티티?
9. 중의적인 표현
10. 대화흐름 차근 차근

끝!

챗봇, 개발자의 영역 이해하기

11. 시스템 연동하기
12. Live Chat스타일 챗봇
13. Event로 시작하는 대화흐름

챗봇, 운영/참고자료

14. 운영 노하우
15. 보너스!
16. 부록

감사합니다!

이 매뉴얼에서 개선해야 할 점이 있다면?
contact@danbee.ai에 교육자료 버전과 함께 보내주시면 적극적으로 반영하겠습니다.

만들다가 막히면?
파브르는 단비Ai 서비스 우측하단에서 언제나 일하고 있습니다.
언제든 물어봐주세요:)