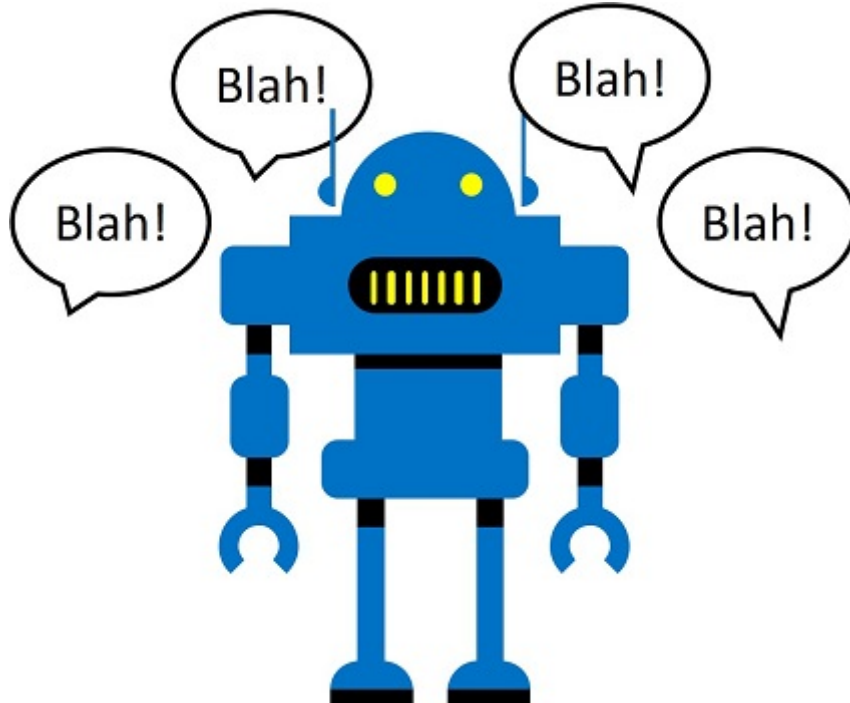


## 언어 이해(Language Understanding)

점점 더, 우리는 컴퓨터가 자연 언어로 말하거나 입력된 명령을 이해하기 위해 AI를 사용할 수 있기를 기대한다. 예를 들어, "전등을 켜라" 또는 "팬을 켜라"와 같은 음성 명령을 사용하여 가정에서 장치를 제어할 수 있도록 하는 홈 자동화 시스템을 구현하고 AI로 구동되는 장치가 명령을 이해하고 적절한 조치를 취하도록 할 수 있다.



## 작성 및 예측 리소스 만들기

Microsoft Cognitive 서비스에는 발언(*Utterance*)에 따라 엔터티에 적용되는 의도(*intent*)를 정의할 수 있는 Language Understanding 서비스가 포함되어 있습니다. **Language Understanding** 또는 **Cognitive 서비스** 리소스를 사용하여 Language Understanding 앱을 게시할 수 있지만 작성을 위해 별도의 **Language Understanding** 리소스를 생성해야 한다.

1. A. 브라우저의 새로운 탭을 열고, Azure portal( <https://portal.azure.com>)을 ( <https://portal.azure.com>)을 입력하고 , Microsoft계정으로 로그인한다..
2. + 리소스 만들기를 클릭하고 *Language Understanding*을 찾는다..
3. 서비스 목록에서 **Language Understanding**을 클릭한다.
4. **Language Understanding** 을 선택하고 **만들기**를 누른다.
5. **만들기** 페이지에서 다음 내용을 입력하고 **만들기**를 클릭한다.
  - **만들기 옵션:** 둘 다
  - **이름:** 유일한 서비스이름
  - **구독:** *Azure* 구독서비스를 선택
  - **리소스 그룹:** 이미 존재하는 리소스그룹을 선택하거나 새로운 것을 만들
  - **작성 위치:** 사용가능한 위치를 선택
  - **작성 가격 책정 계층:** F0
  - **예측 위치:** 작성위치와 같은 위치 선택
  - **예측 가격 책정 계층:** F0 6.리소스가 만들어지까지 기다리면 두개의 Language Understanding 리소스가 만들어지게 되는데 하나는 제작용이고 다른 하나는 예측용이다. 만들어진 리소스 그룹을 클릭하여 각 리소스를 확인할 수 있다.

## Language Understanding 앱 만들기

Language Understanding를 통해 자연어 Language Understanding을 구현하려면 앱을 만든 다음 엔터티, 의도(intents) 및 발언(Utterance)을 추가하여 앱에서 이해할 명령들을 정의한다.

1. 브라우저의 새로운 탭에서 Language Understanding 포털 <https://www.luis.ai> (<https://www.luis.ai>) 을 열고 Azure 구독과 관련된 Microsoft 계정으로 로그인을 한다. 만일 계정이 Language Understanding 포털에 처음으로 로그인을 하였다면 계정 세부 정보에 액세스가 가능하도록 권한을 부여하도록 하는 작업이 필요하다. Azure 구독에서 만들었던 Language Understanding 작성 리소스를 선택함으로써 *Welcome* 과정을 완료한다.
2. **My Apps** 페이지를 클릭하고 구독을 선택하고 Language Understanding 작성 리소스를 선택한다. 그리고 나서 다음과 같은 설정으로 Conversation을 위한 새로운 앱을 생성한다.
  - **Name:** Home Automation
  - **Culture:** English
  - **Description:** 간단한 홈 자동화시스템
  - **Prediction resource:** 앞에서 만든 *Language Understanding* 예측 리소스
3. 효과적인 Language Understanding 앱을 만드는데 필요한 팁 메뉴가 나타나면 달는다.

## 엔터티(Entity) 만들기

*entity*는 언어 모델이 식별하여 수행할 수 있는 작업이다. 이 경우 Language Understanding 앱은 조명이나 팬과 같은 사무실의 다양한 기기를 제어하는 데 사용되므로, 앱에서 사용할 장치 유형 목록이 포함된 *devices* 엔터티를 만든다. 각 장치 유형에 대해 장치의 이름(예: *light*)과 이 장치 유형을 참조하는 데 사용할 수 있는 모든 동의어(예: *lamp*)를 식별하는 하위 목록을 생성한다.

1. 앱의 Language Understanding 페이지에서, 왼쪽에 있는 **Entities**를 클릭한다. 그리고 나서 **Create**를 클릭하고 새로운 엔터티 이름으로 **device**라 넣고, **List** 형식을 선택하고 **Create** 를 클릭한다..
2. **List items** 페이지에서 **Normalized Values** 값 밑에 **light**라 넣고 ENTER를 누른다..
3. **light** 값이 추가된 뒤에 **Synonyms** 밑에다 **lamp**라 입력하고 ENTER를 누른다..
4. 두번째 리스트 항목으로 **fan** 을 넣고 동의어로 **AC**을 입력한다.

## 의도(Intent) 만들기

*Intents*는 하나 이상의 엔터티에 대해 수행할 작업이다. 예를 들어 조명을 켜거나 팬을 끌 수 있다. 이 경우 장치를 켤 때와 장치를 끌 때의 두 가지 의도를 정의한다. 각 의도에 대해 *Intents*를 나타내는 언어 종류를 나타내는 표본 *Utterance*를 지정한다.

1. 페이지 왼편에 있는 **Intents**를 선택하고. **Create** 클릭하고 **switch\_on**이라는 이름으로intent를 추가한다. 그리고 나서**Done**을 클릭한다..
2. **Examples**와 **Example user input**의 밑으로 이동하여 **turn the light on**이라 utterance를 입력하고 **Enter**를 눌러 이것을 리스트에 포함시킨다.
3. **turn the light on** utterance에서 "light" 단어를 클릭하고 **device** 엔터티의 **light** 값을 할당한다.
4. **switch\_on** intent의 두번째 utterance로 **turn the fan on**를 입력한다. 그리고 나서 "fan"단어를 클릭하고 **device** 엔터티의 **fan** 값을 할당한다.
5. 페이지 왼편에 있는 **Intents**를 선택하고. **Create** 클릭하고 **switch\_off**이라는 이름으로intent를 추가한다
6. **switch\_off** intent에 대한 페이지로 이동하여 **turn the light off** utterance를 입력하고 "light" 단어에 **device** 엔터티의 **light** 값을 할당한다.
7. **switch\_off** intent의 두번째 utterance로 **turn the fan off**구를 입력한다. 그리고 나서 "fan"단어에 **device** 엔터티의 **fan** 값을 할당한다.

## 언어 모델을 학습시키고 테스트하기

이제 엔터티들, intents와 utterance의 형식으로 제공한 데이터를 이용하여 앱의 언어모델을 학습시킨다.

1. 앱의 Language Understanding 페이지에서 **Train**을 클릭하여 언어 모델을 학습시킨다.
2. 모델이 하였다면 **Test**를 클릭하고 아래의 구들을 입력하여 예측된 intent를 Test페이지에서 확인한다.
  - *switch the light on*
  - *turn off the fan*
  - *turn the lamp off*
  - *switch on the AC*
3. 테스트 창을 닫는다.

## 모델을 게시하고 엔드포인트를 설정하기

학습시킨 모델을 클라이언트 응용프로그램에서 사용하기 위해서는 클라이언트 응용프로그램이 새로운 utterance이 보낼 수 있는 엔드포인트로 게시해야 한다. 이렇게 하면 intent와 엔터티들을 예측할 수 있다.

1. 앱에서 Language Understanding 페이지에 있는 **Publish**를 클릭한다. 그다음 **Production slot** 클릭하고 **Done**을 선택한다..
2. 모델이 게시된 후 앱의 Language Understanding 페이지의 위에 있는 **Manage**를 클릭한다. 그리고 나서 **Application Information** 탭에서, 앱의 **App ID**를 확인한다. 이것을 복사해서 아래의 셀에 있는 **YOUR\_LU\_APP\_ID**에 붙여넣기를 한다..
3. **Azure Resources** 탭에서 예측 리소스의 **Primary key**와 **Endpoint URL** 를 확인한다. 복사하여 아래 코드의 **YOUR\_LU\_KEY**와 **YOUR\_LU\_ENDPOINT**에 대신하여 붙여넣기를 한다..
4. 아래 셀의 **Run cell** (▶) 버튼(왼쪽에 있음)을 클릭하여 실행한다. 입력창이 나타나t *turn the light on*라고 입력한다. 해당 테스트는 Language Understanding 모델이 학습을 하여 알맞은 이미지를 보여줄게 된다

In [ ]:

```
from python_code import luis
import matplotlib.pyplot as plt
from PIL import Image
import os
%matplotlib inline

try:
    # API 구성 설정하기
    luis_app_id = 'YOUR_LU_APP_ID'
    luis_key = 'YOUR_LU_KEY'
    luis_endpoint = 'YOUR_LU_ENDPOINT'

    # 명령창 띄우기
    command = input('Please enter a command: \n')

    # 예측된 intent와 엔터티 가져오기(python_code.home_auto.py 코드에 있음)
    action = luis.get_intent(luis_app_id, luis_key, luis_endpoint, command)

    # 알맞은 이미지 보여주기
    img_name = action + '.jpg'
    img = Image.open(os.path.join("data", "luis", img_name))
    plt.axis('off')
    plt.imshow(img)
except Exception as ex:
    print(ex)
```

아래와 같은 구로 위를 다시 실행해 보자:

- *turn on the light*
- *put the lamp off*
- *switch the fan on*
- *switch the light on*
- *switch off the light*
- *turn off the fan*
- *switch the AC on*

**알림:** 만일 Language Understanding앱이 intents와 엔터티를 가져오는 방법에 대해서 궁금하다면 `python_code` 폴더의 `luis.py` 파일을 참고하라.

## 음성 제어(Voice Control)추가하기

지금까지 우리는 텍스트를 분석하는 방법을 살펴보았다. 하지만 점점 더 AI 시스템은 음성 인식을 통해 인간이 소프트웨어 서비스와 의사소통할 수 있게 한다. 이를 지원하기 위해 **Speech Cognitive** 서비스는 음성을 텍스트로 간단하게 기록할 수 있는 방법을 제공한다.

## Cognitive 서비스 리소스 만들기

이미 만들어진 Cognitive 서비스가없다면 다음과 같은 순서로 Azure 구독에서 **Cognitive Services** 리소스를 생성할 수 있다.

1. 브라우저의 새로운 탭을 열고, Azure portal( <https://portal.azure.com>)을 ( <https://portal.azure.com>)을 입력하고 , Microsoft계정으로 로그인한다..
2. + 리소스 만들기 버튼을 클릭하고, *Cognitive Services* 서비스를 찾은 다음, **Cognitive Services** 리소스를 다음과 같은 내용으로 생성한다.
  - 이름: 유일한 이름을 입력한다(가능하면 영문과 숫자사용).
  - 구독: *Azure* 구독선택.
  - 위치: 가능한 위치(한국 중부 추천):
  - 가격책정계층: 표준 S0
  - 리소스 그룹: 원하는 유리한 이름(가능하면 영문과 숫자사용).
3. 배포가 완료될 때까지 기다린다. 그런 다음 Cognitive Services 리소스로 이동하여 \*개요 페이지에서 링크를 클릭하여 서비스 키를 관리한다. 클라이언트 응용 프로그램에서 Cognitive Services 리소스에 연결하려면 엔드포인트와 키가 필요하다.

## Cognitive Services 리소스에 있는 키와 엔드포인트 가져오기

Cognitive Services 리소스를 사용하기 위해서는, 클라이언트 응용프로그램에서는 엔드포인트와 인증 키가 필요하다.

1. Azure portal에서, Cognitive Services 리소스를 선택하고 키 및 엔드포인트 페이지를 선택한 다음 키1 을 복사하여 아래의 **YOUR\_COG\_KEY**.를 붙여 넣는다.
2. 리소스에 있는 엔드포인트 를 복사해서 아래의 **YOUR\_COG\_ENDPOINT**.에 붙여 넣는다.
3. 리소스에 있는 위치 를 복사해서 아래의 **YOUR\_COG\_REGION**.에 붙여 넣는다.
4. 아래 셀을 선택한 다음 셀 왼쪽에있는 셀 실행(▶) 버튼을 클릭하여 아래 코드를 실행한다.

In [ ]:

```
cog_key = 'YOUR_COG_KEY'
cog_endpoint = 'YOUR_COG_ENDPOINT'
cog_region = 'YOUR_COG_REGION'

print('Ready to use cognitive services in {} using key {}'.format(cog_region, cog_key))
```

Cognitive 서비스 리소스에 있는 Speech 서비스를 이용하기 위해서는 Azure Cognitive Services Speech SDK를 설치해야 한다.

In [ ]:

```
!pip install azure.cognitiveservices.speech
```

아래의 셀을 실행하여 오디오 파일로부터 음성을 텍스트로 바꾸고 이것을 Language Understanding 앱의 명령으로 사용해보자.

In [ ]:

```
from python_code import luis
import os
import IPython
import os
from azure.cognitiveservices.speech import SpeechConfig, SpeechRecognizer, AudioConfig

try:

    # 오디오 파일로부터 음성 명령을 가져온다.
    file_name = 'light-on.wav'
    audio_file = os.path.join('data', 'luis', file_name)

    # 음성 인식기를 구성한다
    speech_config = SpeechConfig(cog_key, cog_region)
    audio_config = AudioConfig(filename=audio_file) # Use file instead of default (microphone)
    speech_recognizer = SpeechRecognizer(speech_config, audio_config)

    # 음성을 텍스트로 바꾸기 위해 일회성 및 동기화된 호출을 사용한다.
    speech = speech_recognizer.recognize_once()

    # 예측된 intent와 엔티티를 가져온다(python_code.home_auto.py에 있는 코드를 가져온다)
    action = luis.get_intent(luis_app_id, luis_key, luis_endpoint, speech.text)

    # 알맞은 이미지를 가져온다
    img_name = action + '.jpg'

    # 오디오를 실행하고 이미지를 보여준다
    IPython.display.display(IPython.display.Audio(audio_file, autoplay=True),
                           IPython.display.Image(data=os.path.join("data", "luis" ,img_name)))
except Exception as ex:
    print(ex)
```

위의 코드에서 음성 파일을 **light-off.wav**로 바꾸어 실행해보자.

## 심화학습

Language Understanding 에 대한 추가 문서는 다음을 참조하라 [service documentation](https://docs.microsoft.com/azure/cognitive-services/luis/)  
(<https://docs.microsoft.com/azure/cognitive-services/luis/>).