

# 구글 머신러닝 스터디잼

## 가이드라인

2019.02.12

박찬성

# 퀵랩(Qwik Lab)

스터디잼을 수행하기 위한  
기본 기반 플랫폼

# 퀵랩 (qwik lab)

스텝-by-스텝의 인스트럭션을 따라서, 요구하는 내용을 완료하는 하나의 과정

크레딧이 필요함 (쿠폰 등록 시 1달 프리패스)

제한된 시간 내에 완료가 되어야 함

**Start Lab** 버튼을 누르면 시간이 깎임

도중에 멈춤이 불가능

인스트럭션을 충분히 숙지한 후, **Start Lab**하는 것이 권장사항

# 퀵랩 (qwik lab) - 2

퀵랩은 기본적으로 **GCP** (구글 클라우드 플랫폼)에 기반함.

**Start Lab**을 하면, 임시적인 **GCP** 계정이 생성되고,  
제한된 시간안에 해당 계정을 이용해서 랩 문제를 해결하면 됨

시간이 종료되면, 임시 **GCP** 계정이 저장하고 있던 모든 데이터가 증발함

- 남기고 싶다면, 백업을 해 둬야함

# 첫 번째 쿼랩

1 크레딧

15분 소요

[Google Cloud Speech API](#)  
[: Qwik Start](#)

# 오버뷰 & 수행 과제

Google Cloud (이하 GC) Speech API는 개발 될 프로그램 (앱)에 구글이 제공하는 음성인식 기술을 탑재할 수 있게 해준다

GC Speech API를 사용해서 오디오 파일을 클라우드로 전송하면, 그 오디오에 포함된 말 소리를 글로 바꾸어 되돌려 받을 수 있다

수행 과제는 크게 다음과 같다

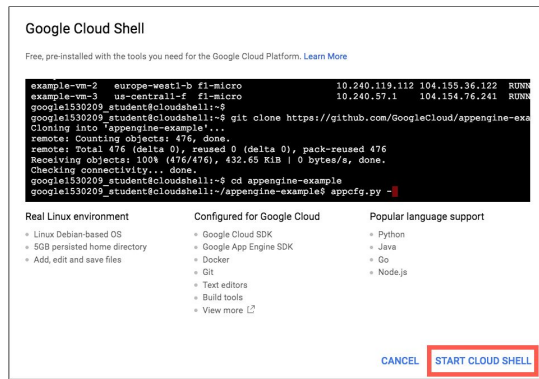
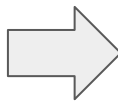
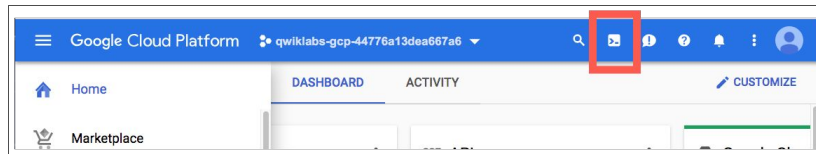
- API 키 생성 해 보기
- Speech API 요청 메시지 **작성**해 보기
- Speech API 요청 메시지 **전송**해 보기

# 진행 방법 (공통-1)

1. Start Lab 버튼 클릭
2. 왼쪽 패널에서 Connection Details 부분을 찾아서,  
⇒ 사용자 이름 (username)을 클립보드로 복사
3. “Open Google Console” 버튼을 클릭 ⇒ 구글 로그인 화면이 나타날 것임
4. 사용자 계정을 선택(Choose an account)해야 한다면,  
⇒ 다른 계정으로 로그인 버튼(Use another account)을 클릭
5. 복사해 둔 사용자 이름을 넣고, 다음(Next) 버튼을 클릭
6. 2번 화면으로 돌아가서, 비밀번호 (password)를 클립보드로 복사

## 진행 방법 (공통-2)

7. 비밀번호 입력 후, Next 버튼 클릭
8. Welcome to your new account 단계에서  
⇒ “Accept the terms and conditions” 체크 후, Next 버튼 클릭
9. Protect your account 단계는 별다른 조작 없이 건너뛸
10. Updates to Terms of Service 단계에서 ⇒ 모두 Yes 선택 후 진행
11. 아래 그림의 빨간 박스를 클릭하여 ⇒ Google Cloud Shell 프로그램을 실행.

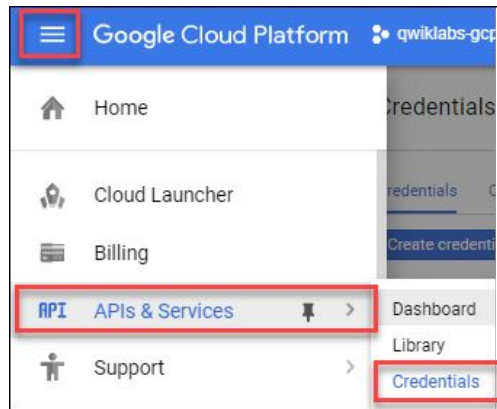
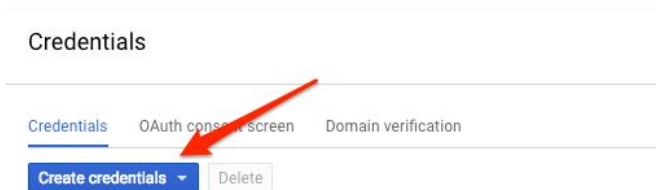




# 진행 방법 (API 키 생성)

Speech API를 사용하려면, 접근시 허가를 위한 키가 있어야함

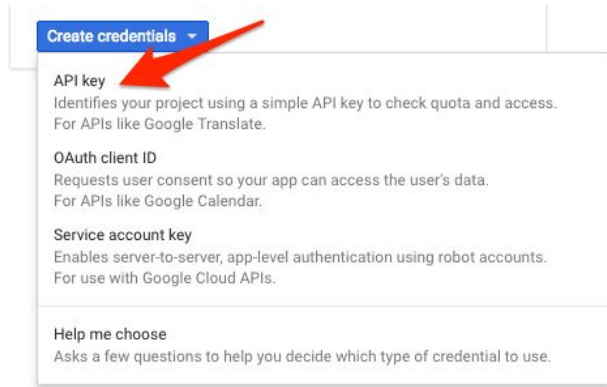
1. 좌측 상단의 햄버거 버튼 클릭한다  
⇒ APIs & Services ⇒ Credentials
2. Create credentials 버튼 클릭한다



## 진행 방법 (API 키 생성-2)

3. 드롭다운 메뉴에서, API key를 선택한다  
⇒ API key가 생성됨

4. Google Cloud Shell 에서,  
아래와 같이 환경 변수를 설정한다  
`export API_KEY=<YOUR_API_KEY>`



\* <YOUR\_API\_KEY>는 3번에서 생성된 API를 넣어줘야함

# 진행 방법 (Speech API 요청 메시지)

1. 텍스트로 변환할 예제 음성 파일 주소의 URL을 확인

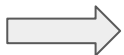
: `gs://cloud-samples-tests/speech/brooklyn.flac`

⇒ GS는 Google cloud Storage의 약자로, GCP에서 제공하는 저장소 (Storage)  
GS에 접근하기 위한, 구글이 정의한 별도의 프로토콜이 있음.

⇒ [샘플 오디오 파일 다운로드 링크](#)

2. Google Cloud Shell에서, 원하는 디렉토리 밑에서  
“request.json” 이라는 파일을 생성한다

```
touch request.json
```



request.json 파일을 생성, 내용은 완전히 비어 있음

## 진행 방법 (Speech API 요청 메시지-2)

3. 사용에 익숙한 편집기를 사용해서, request.json 파일을 연다  
(그룹장의 경우, vim 편집기를 사용)

4. 파일에 아래의 내용을 채워 넣는다

```
{
  "config": {
    "encoding": "FLAC",
    "sample_rate": 16000,
    "language_code": "en-US"
  },
  "audio": {
    "uri": "gs://cloud-samples-tests/speech/brooklyn.flac"
  }
}
```

} 요청을 어떻게 처리  
되어야 하는지를 기술함

encoding: 오디오 파일 형식  
sample\_rate: 오디오 파일의 Hertz 정보  
language\_code: 변환 대상의 언어

[모든 지원되는 언어 리스트 링크](#)  
한국어: "ko-KR"

변환 대상의 오디오파일

5. 저장 후, 편집기에서 빠져나온다

# 진행 방법 (Speech API 요청 전송)

## 1. Google Cloud Shell에서

아래와 같이 **curl** 명령어로 직전에 만든 요청 메시지를 전송한다.

```
curl -s -X POST -H "Content-Type: application/json" --data-binary @request.json \  
"https://speech.googleapis.com/v1beta1/speech:syncrecognize?key=${API_KEY}"
```

동기식으로 요청을 처리함

비동기식을 원하면, **asyncrecognize**를 사용

API Key 환경변수

요청 메시지 파일

## 2. 요청 직후, 응답 메시지를 화면에서 아래와 같이 보여줌

```
{  
  "results": [  
    {  
      "alternatives": [  
        {  
          "transcript": "how old is the Brooklyn Bridge",  
          "confidence": 0.98267895  
        }  
      ]  
    }  
  ]  
}
```

오디오 파일에서 추출된 문자열 텍스트

추출된 결과가 얼마나 신뢰할 수  
있는지에 대한 스코어 (1 == 100%)

# 진행 방법 (종료)

1. End Lab 버튼을 클릭하여, 퀵랩을 종료한다.
2. 퀵랩이 종료되면, 퀵랩을 평가하는 설문지 창이 나타난다.  
(그냥 건너뛰어도 무방함)
3. 정상 종료 후, 새로고침 하면 다음과 같은 문구를 확인할 수 있음

Lab completed on Mon, 11 Feb 2019 00:45:34 -0500. Elapsed time: 4 minutes



# 두 번째 퀵랩

1 크레딧

40분 소요

**Cloud Natural Language API  
: Qwik Start**

# 오버뷰 & 수행 과제

GC 자연어(NL, Natural Language) API는 텍스트에서 사람/장소/이벤트 등 의미있는 정보를 추출해 주는 기능이다. 아래는 주요 기능을 나열한다.

- 문법 분석 (문장의 부분 요소들을 인식 후, 각 요소들간의 관계를 트리화)
- 개체 인식 (사람/장소등 추출) / 감정 분석(문장이 긍정? 부정?)
- 콘텐츠 분류 (미리 정해진 700+ 카테고리)
- 다중 언어 지원 / REST API 통합 지원

수행 과제는 크게 다음과 같다

- API 키 생성 해 보기
- 개체 분석 요청 메시지 전송 해 보기

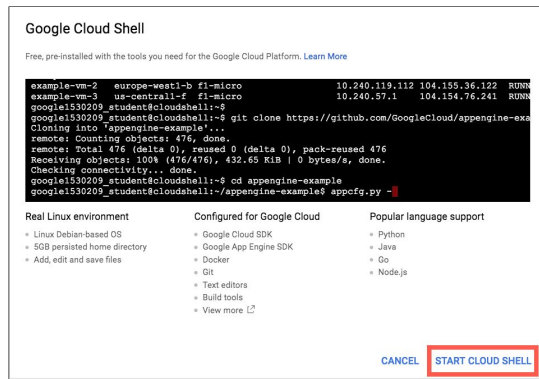
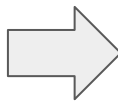
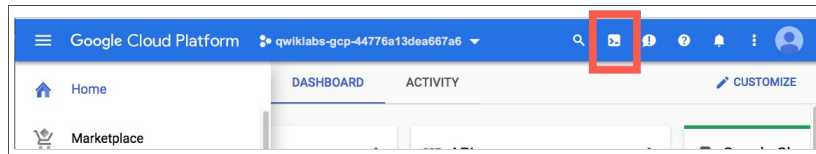


# 진행 방법 (공통-1)

1. Start Lab 버튼 클릭
2. 왼쪽 패널에서 Connection Details 부분을 찾아서,  
⇒ 사용자 이름 (username)을 클립보드로 복사
3. “Open Google Console” 버튼을 클릭 ⇒ 구글 로그인 화면이 나타날 것임
4. 사용자 계정을 선택(Choose an account)해야 한다면,  
⇒ 다른 계정으로 로그인 버튼(Use another account)을 클릭
5. 복사해 둔 사용자 이름을 넣고, 다음(Next) 버튼을 클릭
6. 2번 화면으로 돌아가서, 비밀번호 (password)를 클립보드로 복사

## 진행 방법 (공통-2)

7. 비밀번호 입력 후, Next 버튼 클릭
8. Welcome to your new account 단계에서  
⇒ “Accept the terms and conditions” 체크 후, Next 버튼 클릭
9. Protect your account 단계는 별다른 조작 없이 건너뛸
10. Updates to Terms of Service 단계에서 ⇒ 모두 Yes 선택 후 진행
11. 아래 그림의 빨간 박스를 클릭하여 ⇒ Google Cloud Shell 프로그램을 실행.



# 진행 방법 (API 키 생성)

NL API를 사용하려면, 접근시 허가를 위한 키가 있어야함. Speech API와는 다르게, JSON 파일로 접근 Key가 저장됨.

1. Google Cloud Shell에서 다음 명령을 수행하여, 현재 프로젝트 ID를 추출한 값을 GOOGLE\_CLOUD\_PROJECT 환경변수로 등록한다.

```
export GOOGLE_CLOUD_PROJECT=$(gcloud config get-value core/project)
```

현재 프로젝트 ID 추출 명령

2. NL API의 접근을 위한 IAM Service Account 라는 것을 아래의 명령으로 생성한다.

```
gcloud iam service-accounts create my-natlang-sa \n--display-name "my natural language service account"
```

Service Account 의 내부 이름

IAM: Identity and Access Management

⇒ 특정 사용자에게 합당한 이유, 합당한 시간에 사용 가능한 클라우드 자원을 사용할 수 있는 권한을 컨트롤함

Service Account 의 외부 노출 이름  
(옵서널 정보로, 반드시 필요한 것은 아님)

## 진행 방법 (API 키 생성-2)

3. 생성된 IAM Service Account를 통해서, 아래 명령어로 **API 키**를 생성한다.  
(생성될 API 키는 JSON 파일로 저장됨)

```
gcloud iam service-accounts keys create ~/key.json \
```

```
--iam-account my-natlang-sa@${GOOGLE_CLOUD_PROJECT}.iam.gserviceaccount.com
```

API 키가 저장될 파일 이름 지정

IAM Service Account  
로그인 플래그

2번에서 만든  
IAM Service Account 이름

1번에서 만든  
프로젝트 ID 환경변수

4. 생성된 **API 키**의 위치를 손쉽게 활용하기 위해서,  
아래 명령으로 환경변수로 등록

```
export GOOGLE_APPLICATION_CREDENTIALS="/home/USER/key.json"
```

JSON 파일이 있는 디렉토리 위치를  
“pwd” 명령으로 확인 후, 입력

# 진행 방법 (개체 분석 요청 전송)

1. 단순히 아래의 gcloud 명령을 수행한다.

NL API 요청을  
수행하겠다는 의미

NL API의 기능 중  
개체 분석을 의뢰한다는 의미



다른 분석 의뢰 명령 종류

- analyze-entity-sentiment
- analyze-sentiment
- analyze-syntax
- classify-text

gcloud ml language analyze-entities \

--content="Michelangelo Caravaggio, Italian painter, is known for 'The Calling of Saint Matthew'."

분석 대상이 될 텍스트

--content-file=파일이름

⇒ 파일 자체를 분석 의뢰할 수도 있음

--language=언어

⇒ 분석 대상 언어를 선택할 수 있음

⇒ 명시하지 않으면, 자동으로 감지함

# 진행 방법 (개체 분석 요청 전송-2)

## 2. 결과 정보가 수신 후, 출력된다. (지면상 약간 잘라냈음)

```
{
  "entities": [...]
  {
    "name": "Michelangelo Caravaggio",
    "type": "PERSON",
    "metadata": {
      "wikipedia_url": "http://en.wikipedia.org/wiki/Caravaggio",
      "mid": "/m/020bg"
    },
    "salience": 0.83047235,
    "mentions": [
      {
        "text": {
          "content": "Michelangelo Caravaggio",
          "beginOffset": 0
        },
        "type": "PROPER"
      },
      .....
    ]
  },
  .....
],
"language": "en"
}
```

개체 하나에 대한  
분석 내용

**name:** 개체 이름 (전체 텍스트 중 분석 대상이 된 부분 문자열)

**type:** PERSON? LOCATION? EVENT?

**metadata:** 연관된 위키피디아 정보가 있을시, 포함됨

**salience:** 전체 텍스트에서, 이 개체가 중요한 정도 (최대 1.0)

**mentions:** 전체 텍스트에서, 이 개체와 동일한 개체가 발견된 다른  
지점(index)에 대한 정보 목록

# 진행 방법 (종료)

1. End Lab 버튼을 클릭하여, 퀵랩을 종료한다.
2. 퀵랩이 종료되면, 퀵랩을 평가하는 설문지 창이 나타난다.  
(그냥 건너뛰어도 무방함)
3. 정상 종료 후, 새로고침 하면 다음과 같은 문구를 확인할 수 있음

Lab completed on Mon, 11 Feb 2019 00:45:34 -0500. Elapsed time: 4 minutes



# 세 번째 퀵랩

5 크레딧  
30분 소요

**Speech to Text Transcription  
with the Cloud Speech API**



# 오버뷰 & 수행 과제

왜? 쿼랩1번은 v1beta1 를 사용하는지 모르겠음?

Cloud Speech API를 실습하는 내용으로

⇒ 1번 쿼랩(베타버전)에서 사용된 **API**와 사실상 “거의” 동일한 **API**를 사용한다.

⇒ 결국 과정도 1번 쿼랩과 “거의” 동일하다.

다른점 (수행 과제 측면, **API** 사용법이 아님)

- 영어 외의 다른 언어에 대한 실습 포함
- **API**를 호출하기 위한 **URL**이 다르다.

⇒ 1번 쿼랩의 경우 > [https://speech.googleapis.com/v1beta1/speech:syncrecognize?key=\\${API\\_KEY}](https://speech.googleapis.com/v1beta1/speech:syncrecognize?key=${API_KEY})

⇒ 3번 쿼랩의 경우 > [https://speech.googleapis.com/v1/speech:syncrecognize?key=\\${API\\_KEY}](https://speech.googleapis.com/v1/speech:syncrecognize?key=${API_KEY})

April 18, 2017

**FEATURE** Release of Speech-to-Text v1.

**CHANGED** The **v1beta1** release of Speech-to-Text has been deprecated. **v1beta1** continues to be available for a period of time as defined in the [terms of service](#). To avoid being impacted when the **v1beta1** is discontinued, replace references to **v1beta1** in your code with **v1** and update your code with valid **v1** API names and values.



공식문서에 적힌 내용

⇒ **v1beta1**는 사라질 예정. 언제 사라질지는 모름

⇒ 현재는 **v1**이 배포되어, **v1beta1**을 대체하고 있음.

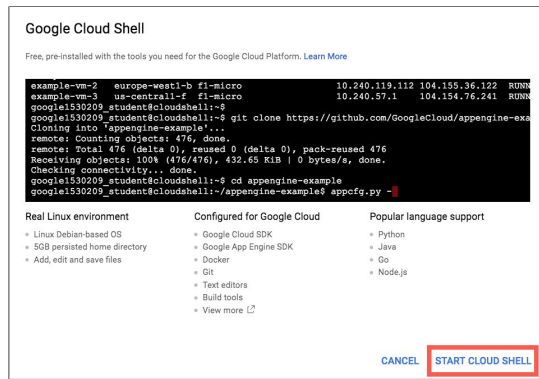
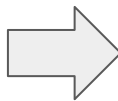
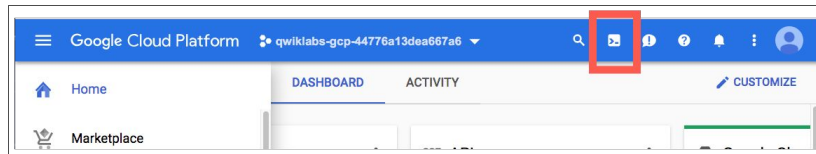
⇒ 프로덕션 제품 개발시엔 **v1beta1**을 사용하지 말것

# 진행 방법 (공통-1)

1. Start Lab 버튼 클릭
2. 왼쪽 패널에서 Connection Details 부분을 찾아서,  
⇒ 사용자 이름 (username)을 클립보드로 복사
3. “Open Google Console” 버튼을 클릭 ⇒ 구글 로그인 화면이 나타날 것임
4. 사용자 계정을 선택(Choose an account)해야 한다면,  
⇒ 다른 계정으로 로그인 버튼(Use another account)을 클릭
5. 복사해 둔 사용자 이름을 넣고, 다음(Next) 버튼을 클릭
6. 2번 화면으로 돌아가서, 비밀번호 (password)를 클립보드로 복사

## 진행 방법 (공통-2)

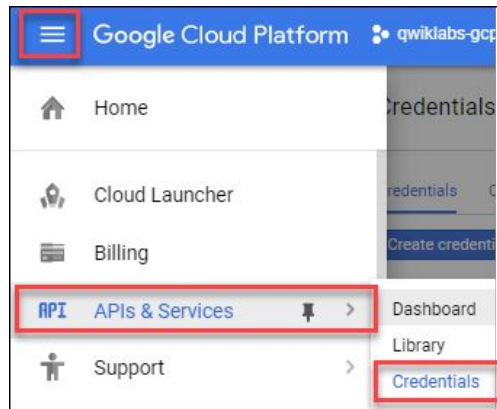
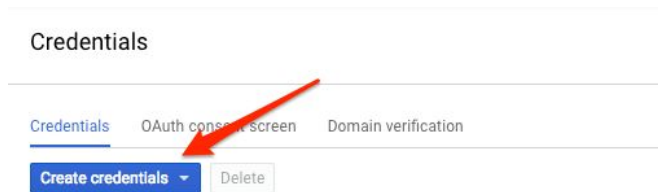
7. 비밀번호 입력 후, Next 버튼 클릭
8. Welcome to your new account 단계에서  
⇒ “Accept the terms and conditions” 체크 후, Next 버튼 클릭
9. Protect your account 단계는 별다른 조작 없이 건너뛸
10. Updates to Terms of Service 단계에서 ⇒ 모두 Yes 선택 후 진행
11. 아래 그림의 빨간 박스를 클릭하여 ⇒ Google Cloud Shell 프로그램을 실행.



# 진행 방법 (API 키 생성 / 쿼랩#1과 동일)

Speech API를 사용하려면, 접근시 허가를 위한 키가 있어야함

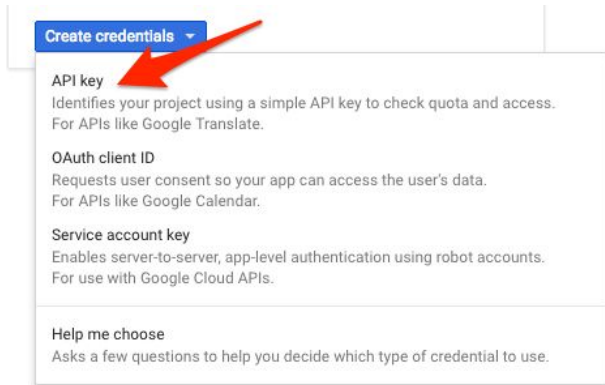
1. 좌측 상단의 햄버거 버튼 클릭한다  
⇒ APIs & Services ⇒ Credentials
2. Create credentials 버튼 클릭한다



# 진행 방법 (API 키 생성-2 / 쿼랩#1과 동일)

3. 드롭다운 메뉴에서, API key를 선택한다  
⇒ API key가 생성됨

4. Google Cloud Shell 에서,  
아래와 같이 환경 변수를 설정한다  
`export API_KEY=<YOUR_API_KEY>`



\* <YOUR\_API\_KEY>는 3번에서 생성된 API를 넣어줘야함

# 진행 방법 (Speech API 요청 메시지 / 쿼랩#1과 동일)

1. 텍스트로 변환할 예제 음성 파일 주소의 URL을 확인

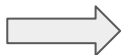
: `gs://cloud-samples-tests/speech/brooklyn.flac`

⇒ GS는 Google cloud Storage의 약자로, GCP에서 제공하는 저장소 (Storage)  
GS에 접근하기 위한, 구글이 정의한 별도의 프로토콜이 있음.

⇒ [샘플 오디오 파일 다운로드 링크](#)

2. Google Cloud Shell에서, 원하는 디렉토리 밑에서  
“request.json” 이라는 파일을 생성한다

```
touch request.json
```



request.json 파일을 생성, 내용은 완전히 비어 있음

# 진행 방법 (Speech API 요청 메시지-2)

/ 퀵랩#1과 동일하지만, `sample_rate`이 없고 `language_code`가 `languageCode`로 바뀜)

3. 사용에 익숙한 편집기를 사용해서, `request.json` 파일을 연다  
(그룹장의 경우, `vim` 편집기를 사용)

4. 파일에 아래의 내용을 채워 넣는다

```
{
  "config": {
    "encoding": "FLAC",
    "languageCode": "en-US"
  },
  "audio": {
    "uri": "gs://cloud-samples-tests/speech/brooklyn.flac"
  }
}
```

} 요청을 어떻게 처리  
되어야 하는지를 기술함

`encoding`: 오디오 파일 형식  
`language_code`: 변환 대상의 언어

[모든 지원되는 언어 리스트 링크](#)  
한국어: "ko-KR"

변환 대상의 오디오파일

5. 저장 후, 편집기에서 빠져나온다

# 진행 방법 (Speech API 요청 전송 / 쿼랩 #1과 동일하지만, URL만 다르다)

## 1. Google Cloud Shell에서

아래와 같이 curl 명령어로 직전에 만든 요청 메시지를 전송한다.

```
curl -s -X POST -H "Content-Type: application/json" --data-binary @request.json  
"https://speech.googleapis.com/v1/speech:recognize?key=${API_KEY}"
```

recognize 서비스 이용

API Key 환경변수

요청 메시지 파일

## 2. 요청 직후, 응답 메시지를 화면에서 아래와 같이 보여줌

```
{  
  "results": [  
    {  
      "alternatives": [  
        {  
          "transcript": "how old is the Brooklyn Bridge",  
          "confidence": 0.98267895  
        }  
      ]  
    }  
  ]  
}
```

오디오 파일에서 추출된 문자열 텍스트

추출된 결과가 얼마나 신뢰할 수  
있는지에 대한 스코어 (1 == 100%)



# 진행 방법 (프랑스어, Speech API 요청 메시지)

1. 텍스트로 변환할 예제 음성 파일 주소의 URL을 확인

: `gs://speech-language-samples/fr-sample.flac`

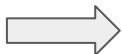
⇒ GS는 Google cloud Storage의 약자로, GCP에서 제공하는 저장소 (Storage)  
GS에 접근하기 위한, 구글이 정의한 별도의 프로토콜이 있음.

⇒ [샘플 오디오 파일 다운로드 링크](#)

2. Google Cloud Shell에서, 원하는 디렉토리 밑에서

“request.json” 이라는 파일을 생성한다 (직전에 만들어진 파일은 삭제한다. “rm request.json”)

```
touch request.json
```



request.json 파일을 생성, 내용은 완전히 비어 있음

## 진행 방법 (프랑스어, Speech API 요청 메시지-2)

3. 사용에 익숙한 편집기를 사용해서, request.json 파일을 연다  
(그룹장의 경우, vim 편집기를 사용)

4. 파일에 아래의 내용을 채워 넣는다

```
{  
  "config": {  
    "encoding": "FLAC",  
    "languageCode": "fr"  
  },  
  "audio": {  
    "uri": "gs://speech-language-samples/fr-sample.flac"  
  }  
}
```

} 요청을 어떻게 처리  
되어야 하는지를 기술함

encoding: 오디오 파일 형식  
language\_code: 변환 대상의 언어

[모든 지원되는 언어 리스트 링크](#)

한국어: "ko-KR"

프랑스어: "fr"

변환 대상의 오디오파일

5. 저장 후, 편집기에서 빠져나온다

# 진행 방법 (프랑스어 Speech API 요청 전송)

## 1. Google Cloud Shell에서

아래와 같이 `curl` 명령어로 직전에 만든 요청 메시지를 전송한다.

```
curl -s -X POST -H "Content-Type: application/json" --data-binary @request.json  
"https://speech.googleapis.com/v1/speech:recognize?key=${API_KEY}"
```

recognize 서비스 이용

API Key 환경변수

요청 메시지 파일

## 2. 요청 직후, 응답 메시지를 화면에서 아래와 같이 보여줌

```
{  
  "results": [  
    {  
      "alternatives": [  
        {  
          "transcript": "maître corbeau sur un arbre perché tenait en son bec un fromage",  
          "confidence": 0.9710122  
        }  
      ]  
    }  
  ]  
}
```

오디오 파일에서 추출된 문자열 텍스트

추출된 결과가 얼마나 신뢰할 수  
있는지에 대한 스코어 (1 == 100%)

# 진행 방법 (종료)

1. End Lab 버튼을 클릭하여, 퀵랩을 종료한다.
2. 퀵랩이 종료되면, 퀵랩을 평가하는 설문지 창이 나타난다.  
(그냥 건너뛰어도 무방함)
3. 정상 종료 후, 새로고침 하면 다음과 같은 문구를 확인할 수 있음

Lab completed on Mon, 11 Feb 2019 00:45:34 -0500. Elapsed time: 4 minutes



# 네 번째 퀵랩

5 크레딧  
40분 소요

**Entity and Sentiment Analysis  
with the Natural Language API**

# 오버뷰 & 수행 과제

#2 쿼랩과 비슷하다.

단지 수행 내용이 더 많고, gcloud CLI 대신 curl을 이용한 API 호출을 한다.

⇒ 이 과정에서 service account 대신, #1 및 #3에서 생성한 것과 동일한 방식으로 credentials를 통한 API 키를 생성해야 한다.

수행 과제는 크게 다음과 같다

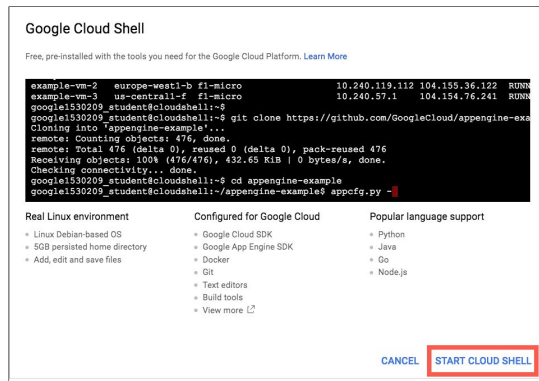
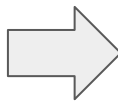
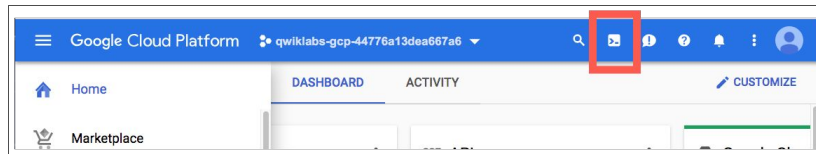
- #2 쿼랩과 동일한, NL API 요청 전송 및 응답 확인
- 감정 분석 (분석을 의뢰한 텍스트가 긍정? 또는 부정? 임을 분석)
- 개체 감정 분석 (분석을 의뢰한 텍스트에 포함된, 개체별 감정을 분석해줌)
- 문법, 품사 분석 (분석을 의뢰한 텍스트의 문법/품사 분석을 해줌)
- 영어 외의 다른 언어에 대해서, NL API 요청 전송 및 응답 확인

# 진행 방법 (공통-1)

1. Start Lab 버튼 클릭
2. 왼쪽 패널에서 Connection Details 부분을 찾아서,  
⇒ 사용자 이름 (username)을 클립보드로 복사
3. “Open Google Console” 버튼을 클릭 ⇒ 구글 로그인 화면이 나타날 것임
4. 사용자 계정을 선택(Choose an account)해야 한다면,  
⇒ 다른 계정으로 로그인 버튼(Use another account)을 클릭
5. 복사해 둔 사용자 이름을 넣고, 다음(Next) 버튼을 클릭
6. 2번 화면으로 돌아가서, 비밀번호 (password)를 클립보드로 복사

## 진행 방법 (공통-2)

7. 비밀번호 입력 후, Next 버튼 클릭
8. Welcome to your new account 단계에서  
⇒ “Accept the terms and conditions” 체크 후, Next 버튼 클릭
9. Protect your account 단계는 별다른 조작 없이 건너뛸
10. Updates to Terms of Service 단계에서 ⇒ 모두 Yes 선택 후 진행
11. 아래 그림의 빨간 박스를 클릭하여 ⇒ Google Cloud Shell 프로그램을 실행.

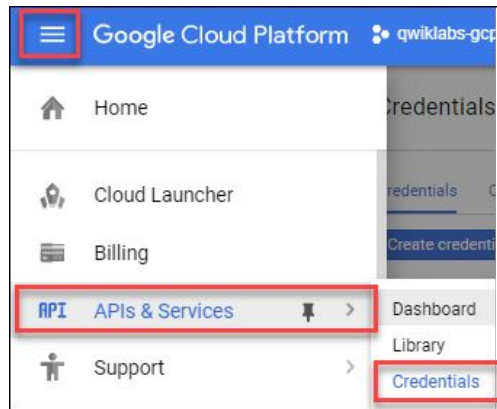
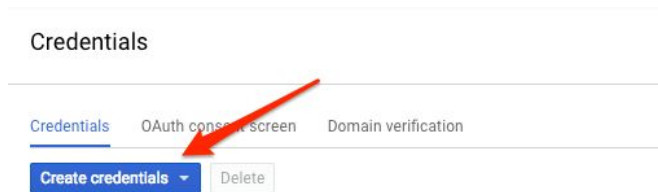




# 진행 방법 (API 키 생성 / 쿼랩#1과 동일)

Speech API를 사용하려면, 접근시 허가를 위한 키가 있어야함

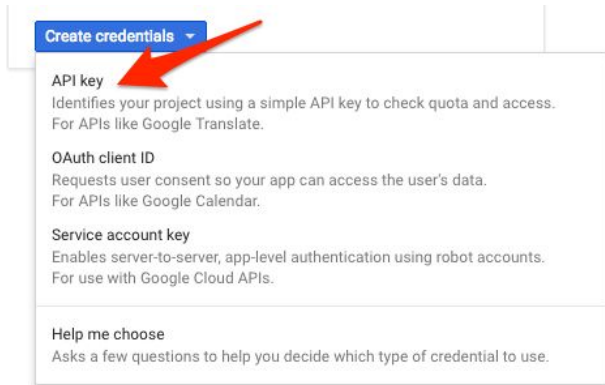
1. 좌측 상단의 햄버거 버튼 클릭한다  
⇒ APIs & Services ⇒ Credentials
2. Create credentials 버튼 클릭한다



# 진행 방법 (API 키 생성-2 / 쿼랩#1과 동일)

3. 드롭다운 메뉴에서, API key를 선택한다  
⇒ API key가 생성됨

4. Google Cloud Shell 에서,  
아래와 같이 환경 변수를 설정한다  
`export API_KEY=<YOUR_API_KEY>`



\* <YOUR\_API\_KEY>는 3번에서 생성된 API를 넣어줘야함

# 진행 방법 (Entity Analysis Request / 개체 분석 요청 메시지 작성)

1. 아래 내용을 가지는 request.json 파일을 원하는 디렉토리에 작성/저장한다.

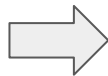
```
{  
  "document": {  
    "type": "PLAIN_TEXT",  
    "content": "Joanne Rowling, who writes under the pen names J. K. Rowling and Robert  
Galbraith, is a British novelist and screenwriter who wrote the Harry Potter fantasy  
series."  
  },  
  "encodingType": "UTF8"  
}
```

type은 “PLAIN\_TEXT”와 “HTML” 두 가지가 지원된다.

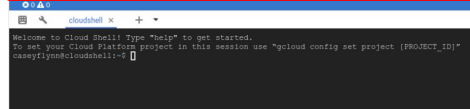
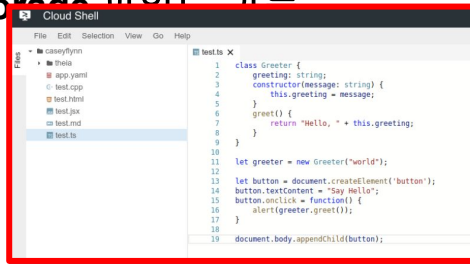
content는 분석 대상이 되는 텍스트를  
기입

⇒ gcsContentUri 를 사용하면, Cloud  
Storage 파일도 가능

- \* Google Cloud Shell 에서  
자체 에디터 기능을 제공.  
(vim/emacs등 사용이  
어려운 경우에 유용함)



편집  
기



# 진행 방법 (Entity Analysis Request / 개체 분석 요청 메시지 전송)

## 1. 다음의 명령어를 수행

```
curl "https://language.googleapis.com/v1/documents:analyzeEntities?key=${API_KEY}" \
-s -X POST -H "Content-Type: application/json" --data-binary @request.json
```

개체 분석 서비스 의뢰를 명시

API 키 환경변수

요청 메시지

## 2. 응답 결과는 쿼랩 #2와 동일한 구조를 가진다.

“entities”: [...] 이므로, 개체 여러개가 저장되는 배열

```
{
  "entities": [
    {
      "name": "Robert Galbraith",
      "type": "PERSON",
      "metadata": {
        "mid": "/m/042xh",
        "wikipedia_url": "https://en.wikipedia.org/wiki/J._K._Rowling"
      },
      "salience": 0.7980405,
      "mentions": [
        {
          "text": {
            "content": "Joanne Rowling",
            "beginOffset": 0
          },
          "type": "PROPER"
        },
        ...
      ],
      ...
    },
    ...
  ],
  ...
}
```

개체 하나에 대한  
분석 내용

**name:** 개체 이름 (전체 텍스트 중 분석 대상이 된 부분 문자열)

**type:** PERSON? LOCATION? EVENT?

**metadata:** 연관된 위키피디아 정보가 있을시, 포함됨

**salience:** 전체 텍스트에서, 이 개체가 중요한 정도 (최대 1.0)

**mentions:** 전체 텍스트에서, 이 개체와 동일한 개체가 발견된 다른  
지점(index)에 대한 정보 목록

# 진행 방법 (Sentiment analysis / 감정 분석 요청 메시지 전송)

1. 앞서 작성한 `request.json` 파일의 내용을 다음과 같이 교체한다. ( 지운 후, 다시 만들어도 무방)

```
{
  "document":{
    "type":"PLAIN_TEXT",
    "content":"Harry Potter is the best book. I think everyone should read it."
  },
  "encodingType": "UTF8"
}
```

2. 다음의 명령어를 수행

```
curl "https://language.googleapis.com/v1/documents:analyzeSentiment?key=${API_KEY}" \
  -s -X POST -H "Content-Type: application/json" --data-binary @request.json
```

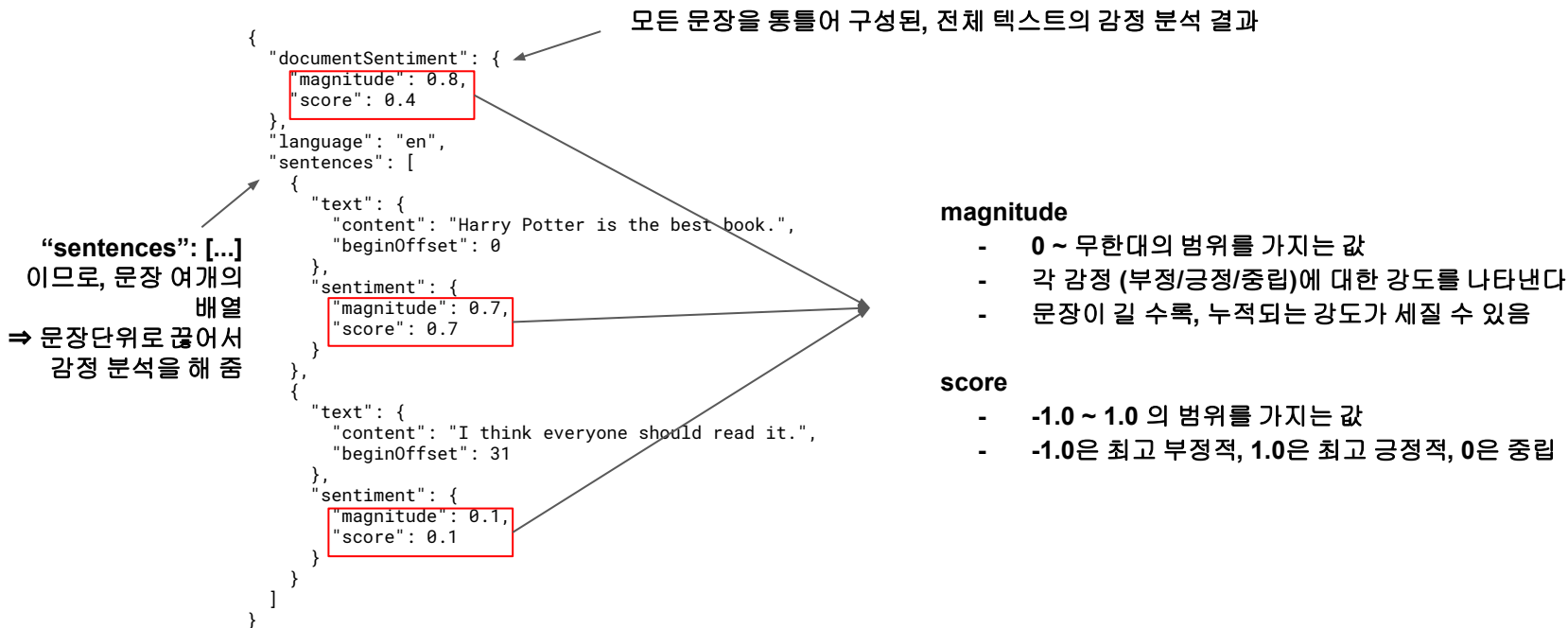
감정 분석 서비스 의뢰를 명시

API 키 환경변수

요청 메시지

# 진행 방법 (Sentiment analysis / 감정 분석 요청 응답)

## 1. 다음과 같은 응답 메시지를 수신한다



# 진행 방법 (Analyzing entity sentiment / 개체별 감정 분석 요청 메시지 전송)

\* “스시가 맛있긴 했지만, 서비스는 최악이었다.” 같은 레스토랑 리뷰에서, 어떤 부분이 긍정이고, 어떤 부분이 부정인지를 구분해서 분류하고 싶을 때가 있다.

1. 앞서 작성한 **request.json** 파일의 내용을 다음과 같이 교체한다. ( 지운 후, 다시 만들어도 무방)

```
{  
  "document":{  
    "type":"PLAIN_TEXT",  
    "content":"I liked the sushi but the service was terrible."  
  },  
  "encodingType": "UTF8"  
}
```

2. 다음의 명령어를 수행

개체별 감정 분석 서비스  
의뢰를 명시

```
curl "https://language.googleapis.com/v1/documents:analyzeEntitySentiment?key=${API_KEY}" \  
-s -X POST -H "Content-Type: application/json" --data-binary @request.json
```

API 키 환경변수

요청 메시지

# 진행 방법 (Analyzing entity sentiment / 개체별 감정 분석 응답)

- 오른쪽과 같은 응답 메시지를 수신한다.
  - 각 **key** 값 (**magnitude**, **salience**, **score**,...)의 의미하는 것은 동일하다.
  - 다만, 각 개체 (**entity**)에 대한 분석 결과를 배열로서 (**entities**) 나열하고 있다.
- 샘플 결과에서 보다시피, **sushi** (스시)에 대한 감정은 스코어가 **0.9**로서, “긍정”
- service** (서비스)에 대한 감정은 스코어가 **-0.9**로서, “부정” 임을 알 수 있다.

```
{
  "entities": [
    {
      "name": "sushi",
      "type": "CONSUMER_GOOD",
      "metadata": {},
      "salience": 0.52716845,
      "mentions": [
        {
          "text": {
            "content": "sushi",
            "beginOffset": 12
          },
          "type": "COMMON",
          "sentiment": {
            "magnitude": 0.9,
            "score": 0.9
          }
        }
      ]
    },
    {
      "name": "service",
      "type": "OTHER",
      "metadata": {},
      "salience": 0.47283158,
      "mentions": [
        {
          "text": {
            "content": "service",
            "beginOffset": 26
          },
          "type": "COMMON",
          "sentiment": {
            "magnitude": 0.9,
            "score": -0.9
          }
        }
      ]
    }
  ],
  "language": "en"
}
```



# 진행 방법 (Analyzing syntax and parts of speech / 품사와 문법 분석 요청 메시지 전송)

1. 앞서 작성한 `request.json` 파일의 내용을 다음과 같이 교체한다. ( 지운 후, 다시 만들어도 무방)

```
{
  "document":{
    "type":"PLAIN_TEXT",
    "content": "Joanne Rowling is a British novelist, screenwriter and film producer."
  },
  "encodingType": "UTF8"
}
```

2. 다음의 명령어를 수행

```
curl "https://language.googleapis.com/v1/documents:analyzeSyntax?key=${API_KEY}" \
  -s -X POST -H "Content-Type: application/json" --data-binary @request.json
```

문법 분석  
의뢰를 명시

API 키 환경변수

요청 메시지

# 진행 방법 (Analyzing syntax and parts of speech / 품사와 문법 분석 응답)

## 1. 다음과 같은 응답 메시지를 수신한다.

(여러개의 **text**가 배열로 존재할 것임, 여기서 하나만 예로 보여줌)

{

```
"text": {
  "content": "is",
  "beginOffset": 15
},
"partOfSpeech": {
  "tag": "VERB",
  "aspect": "ASPECT_UNKNOWN",
  "case": "CASE_UNKNOWN",
  "form": "FORM_UNKNOWN",
  "gender": "GENDER_UNKNOWN",
  "mood": "INDICATIVE",
  "number": "SINGULAR",
  "person": "THIRD",
  "proper": "PROPER_UNKNOWN",
  "reciprocity": "RECIPROCITY_UNKNOWN",
  "tense": "PRESENT",
  "voice": "VOICE_UNKNOWN"
},
"dependencyEdge": {
  "headTokenIndex": 2,
  "label": "ROOT"
},
"lemma": "be"
}, ...
```

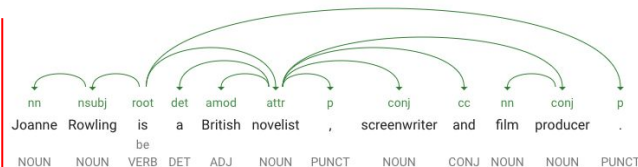
분석 대상 토큰  
- **content**에 명시됨  
- 이 경우에는 “is”

품사 정보  
- “is” 가 그 대상으로,  
동사 (VERB)로 분석됨

의존성 파스 트리 (dependency parse tree)를  
생성하기 위해 사용될 수 있는 정보

<https://cloud.google.com/natural-language/> 서비스로 만들어 볼 수 있음  
<https://cloud.google.com/natural-language/docs/reference/rest/v1/Token#Label>  
⇒ 레이블 종류 나열 문서

원형  
- is, was, were, am, are  
의 원형은 be



[의존성 파스 트리] 구성의 예

### headTokenIndex

- 이 토큰이 의존성 파스 트리에서  
**Joanne**을 가리키는 인덱스 정보  
**label**  
- 문장에서 해당 토큰의 역할  
**ROOT**는 어근을 의미함

# 진행 방법 (Multilingual natural language processing / 영어 외의 다른 자연어 처리 요청 메시지 전송)

1. 앞서 작성한 `request.json` 파일의 내용을  
다음과 같이 교체한다. ( 지운 후, 다시 만들어도 무방)

```
{  
  "document": {  
    "type": "PLAIN_TEXT",  
    "content": "日本のグーグルのオフィスは、東京の六本木ヒルズにあります"  
  }  
}
```

2. 다음의 명령어를 수행

```
curl "https://language.googleapis.com/v1/documents:analyzeEntities?key=${API_KEY}" \  
-s -X POST -H "Content-Type: application/json" --data-binary @request.json
```

개체 분석  
의뢰를 명시

API 키 환경변수

요청 메시지

# 진행 방법 (Multilingual natural language processing / 영어 외의 다른 자연어 처리 응답, 언어만 다르고, 응답 구조는 동일함)

“entities”: [...]   
이므로, 개체  
여러개가  
저장되는 배열

```
{  
  "entities": [  
    {  
      "name": "日本",  
      "type": "LOCATION",  
      "metadata": {  
        "mid": "/m/03_3d",  
        "wikipedia_url": "https://en.wikipedia.org/wiki/Japan"  
      },  
      "salience": 0.23854347,  
      "mentions": [  
        {  
          "text": {  
            "content": "日本",  
            "beginOffset": 0  
          },  
          "type": "PROPER"  
        }  
      ],  
      ...  
    },  
    ...  
  ]  
}
```

개체 하나에 대한  
분석 내용

**name:** 개체 이름 (전체 텍스트 중 분석 대상이 된 부분 문자열)

**type:** PERSON? LOCATION? EVENT?

**metadata:** 연관된 위키피디아 정보가 있을시, 포함됨

**salience:** 전체 텍스트에서, 이 개체가 중요한 정도 (최대 1.0)

**mentions:** 전체 텍스트에서, 이 개체와 동일한 개체가 발견된 다른  
지점(index)에 대한 정보 목록

# 진행 방법 (종료)

1. End Lab 버튼을 클릭하여, 퀵랩을 종료한다.
2. 퀵랩이 종료되면, 퀵랩을 평가하는 설문지 창이 나타난다.  
(그냥 건너뛰어도 무방함)
3. 정상 종료 후, 새로고침 하면 다음과 같은 문구를 확인할 수 있음

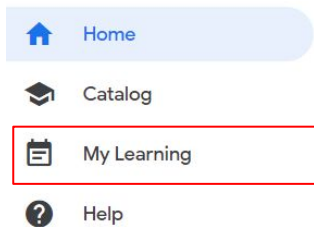
Lab completed on Mon, 11 Feb 2019 00:45:34 -0500. Elapsed time: 4 minutes



# 완료 확인 방법

# 네 개의 퀵랩을 모두 정상 종료 했는지 확인

1. <https://www.qwiklabs.com/home> 으로 접속
2. 로그인 안되어 있으면, 로그인 한다.
3. **My Learning**을 클릭



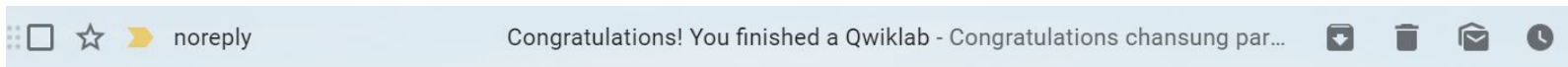
4. 다음과 같이 **Completed Labs**에 네 개의 리스트를 확인

## Completed Labs

✓	Entity and Sentiment Analysis with the Natural Language API	Completed 7 minutes ago
✓	Speech to Text Transcription with the Cloud Speech API	Completed about 10 hours ago
✓	Cloud Natural Language API: Qwik Start	Completed about 10 hours ago
✓	Google Cloud Speech API: Qwik Start	Completed about 19 hours ago

# 네 개의 퀵랩을 모두 정상 종료 했는지 확인

5. 일정 시간이 지난 후, 계정 이메일에 다음과 같은 메일이 수신됨을 확인한다.



Congratulations chansung park,

You completed the lab *Entity and Sentiment Analysis with the Natural Language API*. Your completion percentage was 85.0%. Get more information by visiting your [dashboard](#).

We want to know what you thought about this lab. Please take our short survey [here](#).

Thank you,

Steve

Qwiklabs

[support@qwiklabs.com](mailto:support@qwiklabs.com)



모두 수고 하셨습니다.

부족한 자료를 많이 활용해 주셔서 감사 드립니다