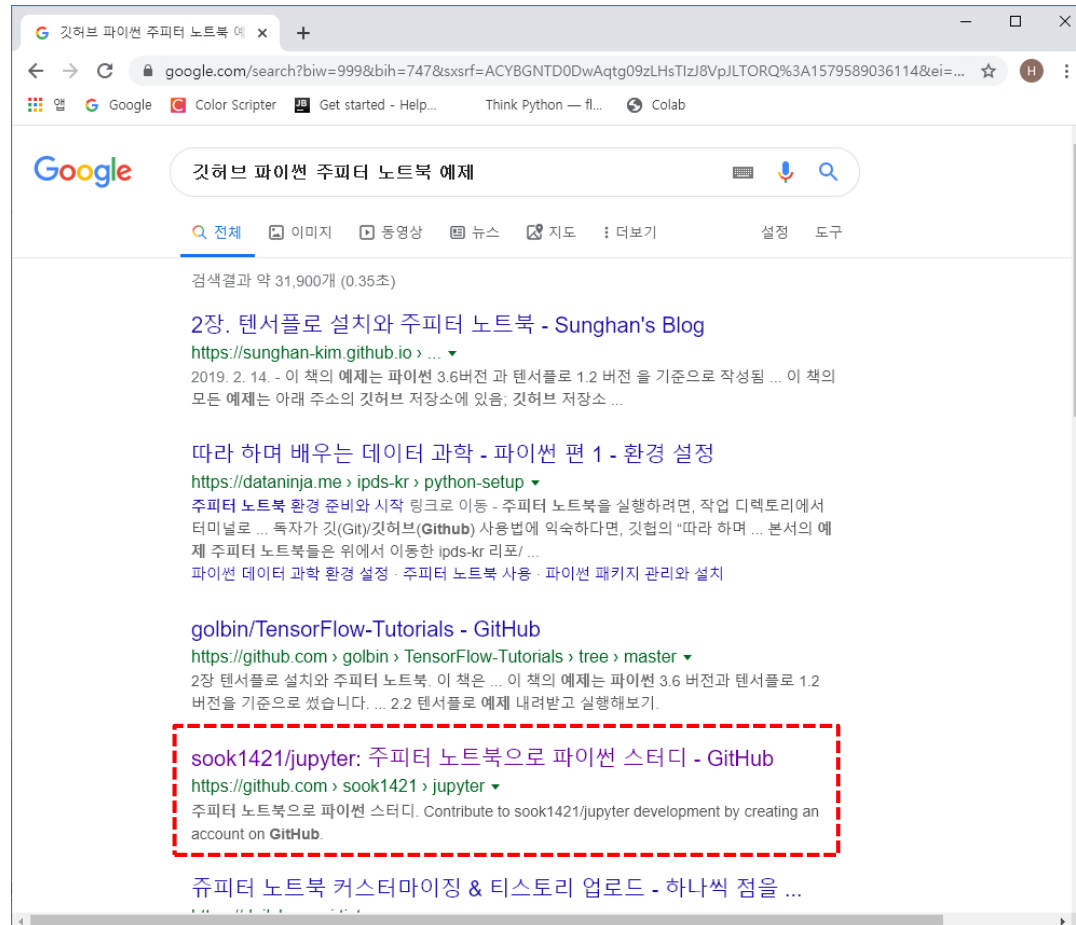


깃허브의 노트 파일을 Colab에서 활용

깃허브 사이트 검색

- <https://github.com/sook1421/jupyter>



노트 열기

- sook1421/jupyter

The screenshot shows the JupyterLab interface. The 'File' menu is open, displaying various options. The 'Open...' option is highlighted. In the background, the 'Open' dialog is visible, showing a search bar with the text 'sook1421/jupyter'. Below the search bar, a list of files is shown, including '기본 문법 알아보기 (파이썬).ipynb', which is highlighted with a red dashed box. The 'File' menu is also highlighted with a red dashed box.

파일 열기 후

모두 실행

The screenshot shows a JupyterLab interface with a file named '기본 문법 알아보기 (파이썬).ipynb' open. The 'Run' (런타임) menu is open, and the 'Run All' (모두 실행) option is highlighted. The menu also lists other options like 'Run Cell' (이전 셀 실행), 'Run Cell and Next' (초점이 맞춰진 셀 실행), 'Run Cell and Previous' (선택항목 실행), 'Run Cell and Next' (이후 셀 실행), 'Interrupt' (실행 중단), 'Restart' (런타임 다시 시작...), 'Restart and Run All' (다시 시작 및 모두 실행...), 'Reset Interactive Environment' (런타임 초기화), 'Change Interactive Environment' (런타임 유형 변경), 'Session Management' (세션 관리), and 'View Runtime Log' (런타임 로그 보기).

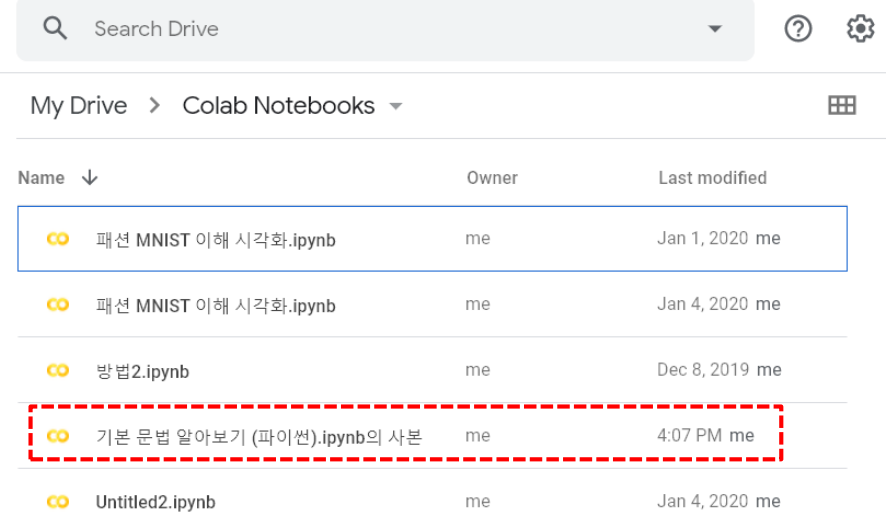
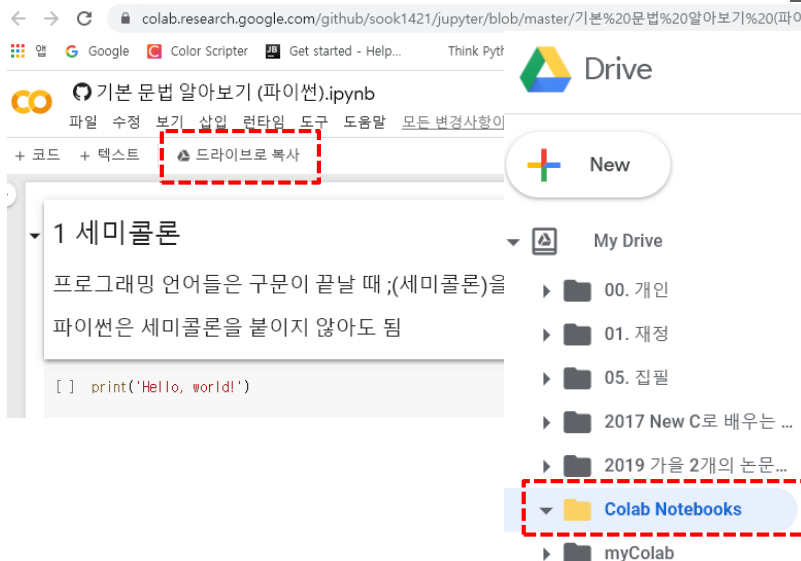
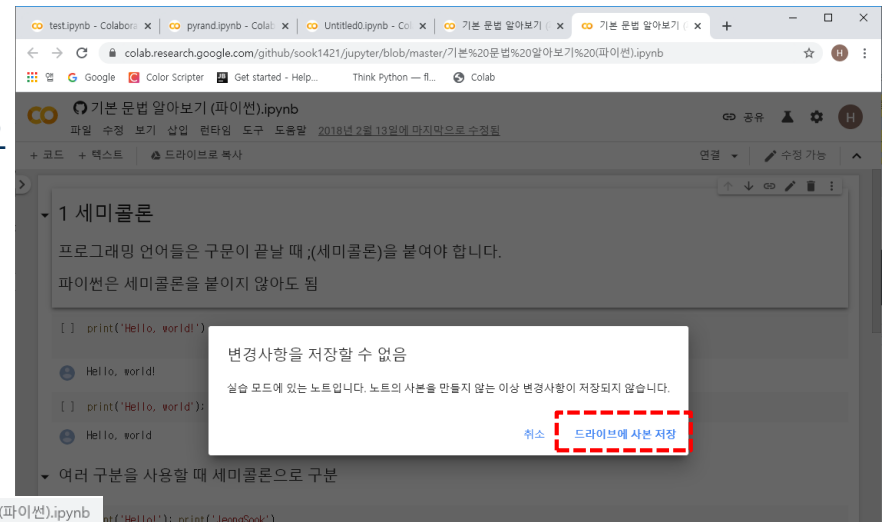
The notebook content includes:

- 1 세미콜론**
프로그래밍 언어들은 파이썬은 세미콜론을 사용하지 않습니다.
- `[] print('Hello, world!')`
Hello, world!
- `[] print('Hello, world')`
Hello, world
- 여러 구분을 사용할 때 세미콜론으로 구분**
`[] print('Hello!'); print('JeongSook')`
Hello!
JeongSook
- 2 주석**
주석은 프로그램의 실행에 영향을 주지 않습니다.
한 줄 주석과 범위 주석 두 가지가 있는데,,

내 파일로 저장

• 저장

- '드라이브에 사본 저장...'이 먼저 필요
- 드라이브의 다음 폴더에 저장
 - Colab Notebooks
- 버튼 메뉴
 - 드라이버로 복사 사용 권장



실습

- '케라스 창시자에게 배우는 딥러닝 깃허브' 검색
 - <https://github.com/gilbutITbook/006975>

첫 파일 열기

• 2.1-a-first-look-at-a-neural-network.ipynb

에

최근 사용

Google 드라이브

GitHub


업로드

GitHub URL을 입력하거나 조직 또는 사용자로 검색하세요.


☒ 비공개 저장소 포함

gilbutITbook/006975

🔍


저장소: 

gilbutITbook/006975 ▼



브랜치: 


master ▼

경로






2.1-a-first-look-at-a-neural-network.ipynb






3.4-classifying-movie-reviews.ipynb





3.5-classifying-newswires.ipynb



3.6-predicting-house-prices.ipynb

취소

자신 드라이브에 저장

2.1-a-first-look-at-a-neural-network.ipynb의 사본 ☆

파일 수정 보기 삽입 런타임 도구 도움말 오후 4:43에 마지막으로 저장됨

+ 코드 + 텍스트

```
import keras
keras.__version__
```

Using TensorFlow backend.
'2.2.2'

신경망과의 첫 만남

이 노트북은 [케라스 창시자에게 배우는 딥러닝](#) 책의 2장 1절의 코드 예제입니다. 책에는 더 많은 내용과 그림이 있습니다. 이 노트북에는 소스 코드에 관련된 설명만 포함합니다.

케라스 파이썬 라이브러리를 사용하여 손글씨 숫자 분류를 학습하는 구체적인 신경망 예제를 살펴보겠습니다. 케라스나 비슷한 라이브러리를 사용한 경험이 없다면 당장은 이 첫 번째 예제를 모두 이해하지 못할 것입니다. 아직 케라스를 설치하지 않았을지도 모릅니다. 괜찮습니다. 다음 장에서 이 예제를 하나하나 자세히 설명합니다. 코드가 좀 이상하거나 요술처럼 보더라도 너무 걱정하지 마세요. 일단 시작해 보겠습니다.

여기에서 돌리고 하는 문제는 흑백 손글씨 숫자 이미지(28x28 픽셀)를 10개의 범주(0에서 9까지)로 분류하는 것입니다. 머신 러닝 커뮤니티에서 고전으로 취급받는 데이터셋인 MNIST를 사용하겠습니다. 이 데이터셋은 머신 러닝의 역사만큼 오래되었고 많은 연구에 사용되었습니다. 이 데이터셋은 1980년대에 미국 국립표준기술연구소에서 수집한 6만 개의 훈련 이미지와 1만 개의 테스트 이미지로 구성되어 있습니다. MNIST 문제를 알고리즘이 제대로 작동하는지 확인하기 위한 딥러닝계의 'Hello World'라고 생각해도 됩니다. 머신 러닝 기술자가 되기까지 연구 논문이나 블로그 포스트 등에서 MNIST를 보고 또 보게 될 것입니다.

MNIST 데이터셋은 넘파이 배열 형태로 케라스에 이미 포함되어 있습니다:

```
[ ] from keras.datasets import mnist

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

train_images와 train_labels가 모델이 학습해야 할 훈련 세트를 구성합니다. 모델은 test_images와 test_labels를 사용하여 테스트합니다.

My Drive > Colab Notebooks

Name	Owner	Last modified	
2.1-a-first-look-at-a-neural-network.ipynb의 사본	me	4:43 PM	
기본 문법 알아보기 (파이썬).ipynb의 사본	me	4:07 PM	

노트 파일의 깃허브 URL을
Colab으로 열기

노트 파일을 읽는 방법

- 단일 .ipynb 파일을 clone 하는 방법
 - <https://github.com/~ ~ ~> 에서 ~ ~ ~ 부분을
 - <https://colab.research.google.com/github/~ ~ ~> 로 교체

깃허브 노트 바로 코랩에서 열기

- 내가 알고 있는 노트 파일 url
 - <https://github.com/zzsza/TIL/blob/master/python/tensorflow-1.ipynb>
 - 노트저장소
 - 깃허브 서버 정보 빼고 다음 정보 확인
 - </zzsza/TIL/blob/master/python/tensorflow-1.ipynb>
- 코랩 url 이후 /github 를 붙이고 계속 붙이기
 - <https://colab.research.google.com>
 - /github 추가
 - <https://colab.research.google.com/github>
- 다음 주소로 바로 접근
 - <https://Colab주소/github/노트저장소>
 - <https://colab.research.google.com/github/zzsza/TIL/blob/master/python/tensorflow-1.ipynb>

코랩 접속 화면

The screenshot shows the Google Colaboratory (Colab) interface in a web browser. The browser tab is titled "tensorflow-1.ipynb - Colaboratory". The address bar shows the URL "colab.research.google.com/github/zzsza/TIL/blob/master/python/tensorflow-1.ipynb". The Colab logo and "tensorflow-1.ipynb" are visible in the top left. The top right shows a share icon, a user profile icon, and a settings icon. Below the top bar, there are tabs for "파일" (File), "수정" (Edit), "보기" (View), "삽입" (Insert), "런타임" (Runtime), "도구" (Tools), and "도움말" (Help). A status bar indicates "오후 4:53 이후 저장되지 않은 변경사항" (Changes not saved after 4:53 PM). The main area shows a Jupyter Notebook with the following code cells:

```

import tensorflow as tf

[ ] hello = tf.constant("Hi")

[ ] print(hello)

Tensor("Const:0", shape=(), dtype=string)

[ ] # Tensor : 자료형

[ ] a = tf.constant(10)
    b = tf.constant(2)
    c = a + b

[ ] print(c)

Tensor("add:0", shape=(), dtype=int32)

[ ] # 텐서플로우는 그래프 생성 후, 그래프 실행하는 과정을 거쳐야 함. 지연 실행 방법을 사용

[ ] sess = tf.Session()
    print(sess.run(c))

12

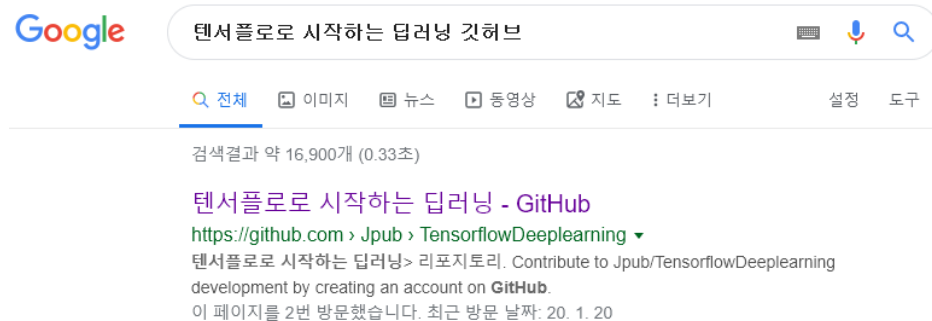
[ ] print(sess.run(hello))

b'Hi'

```

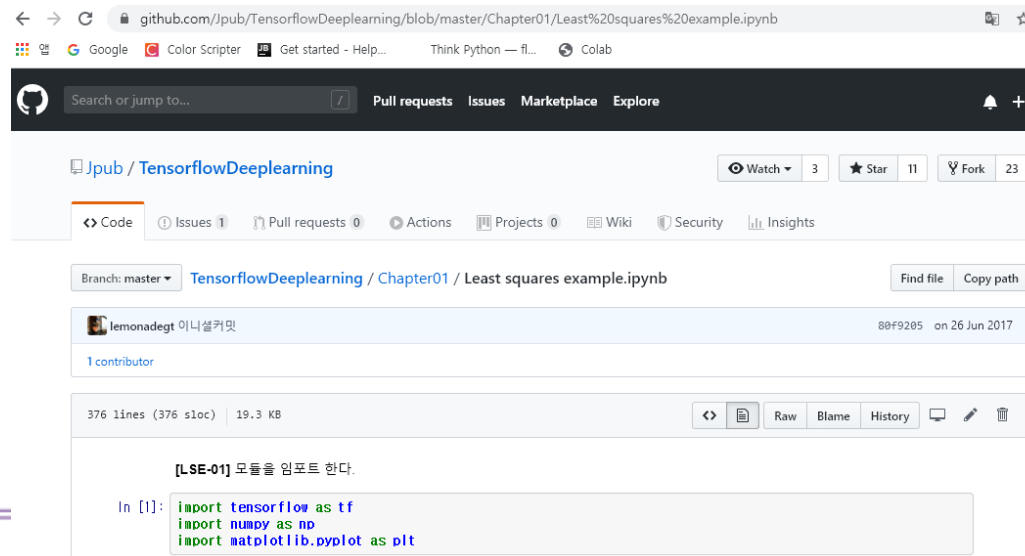
실습

• 텐서플로로 시작하는 딥러닝 깃허브



• 깃허브 페이지

- <https://github.com/Jpub/TensorflowDeeplearning/blob/master/Chapter01/Least%20squares%20example.ipynb>



접속 url

- 코랩 url

- <https://colab.research.google.com>

- 노트 url

- [/Jpub/TensorflowDeeplearning/blob/master/Chapter01/Least%20squares%20example.ipynb](https://colab.research.google.com/Jpub/TensorflowDeeplearning/blob/master/Chapter01/Least%20squares%20example.ipynb)

- 결과

- [/github 추가](#)
- <https://colab.research.google.com/github/Jpub/TensorflowDeeplearning/blob/master/Chapter01/Least%20squares%20example.ipynb#scrollTo=2ULb6gM5JWDO>

열기 성공

- 주소

- <https://colab.research.google.com/github/Jpub/TensorflowDeeplearning/blob/master/Chapter01/Least%20squares%20example.ipynb#scrollTo=2ULb6gM5JWDO>

```
[LSE-01] 모듈을 임포트 한다.  
  
import tensorflow as tf  
import numpy as np  
import matplotlib.pyplot as plt  
  
[LSE-02] Placeholder x를 정의한다.  
  
[ ] x = tf.placeholder(tf.float32, [None, 5])  
  
[LSE-03] Variable w를 정의한다.  
  
[ ] w = tf.Variable(tf.zeros([5, 1]))  
  
[LSE-04] 계산식 y를 정의한다.  
  
[ ] y = tf.matmul(x, w)  
  
[LSE-05] Placeholder t를 정의한다.  
  
[ ] t = tf.placeholder(tf.float32, [None, 1])  
  
[LSE-06] 오차 함수 loss를 정의한다.
```

실습

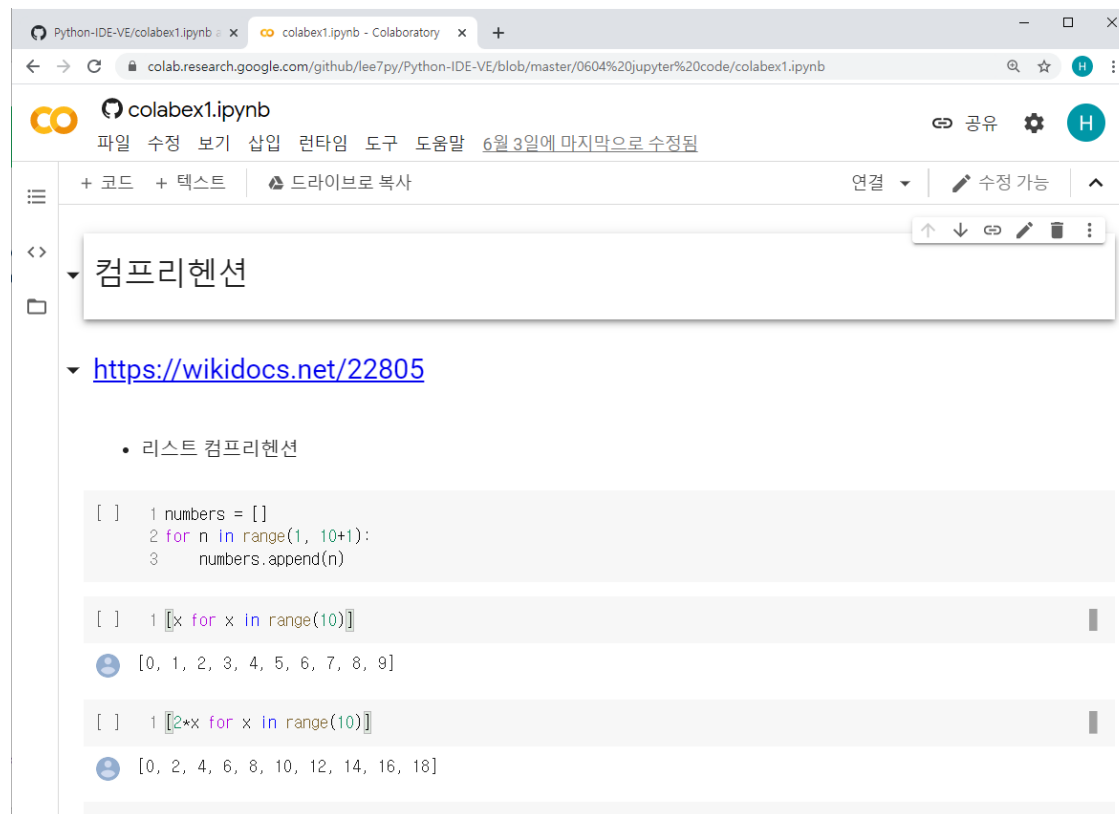
- 우리 수업 사이트

- <https://github.com/lee7py/Python-IDE-VE/>
- 파일
 - <https://github.com/lee7py/Python-IDE-VE/blob/master/0604%20jupyter%20code/colabex1.ipynb>

우리 자료 코랩으로 열기

- URL

- <https://colab.research.google.com/github/lee7py/Python-IDE-VE/blob/master/0604%20jupyter%20code/colabex1.ipynb>



구글 Colab 노트
Github에 저장

자신의 깃허브에 저장소를 하나 생성

- Git7py에 하나 생성
 - Python ide ve

Create a New Repository

github.com/new

Search or jump to...

Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: git7py / Repository name: python ide ve ✓

Great repository name: Your new repository will be created as python-ide-ve. about super-duper-potato?

Description (optional)

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None Add a license: None ⓘ

Create repository

Welcome에서 Pandas 소개로 이동

- 다양한 자습 리소스 지원

머신러닝 단기집중과정

다음은 Google 온라인 머신러닝 과정에서 가져온 일부 노트입니다. [전체 과정 웹사이트](#)에서 자세한 내용을 확인하세요.

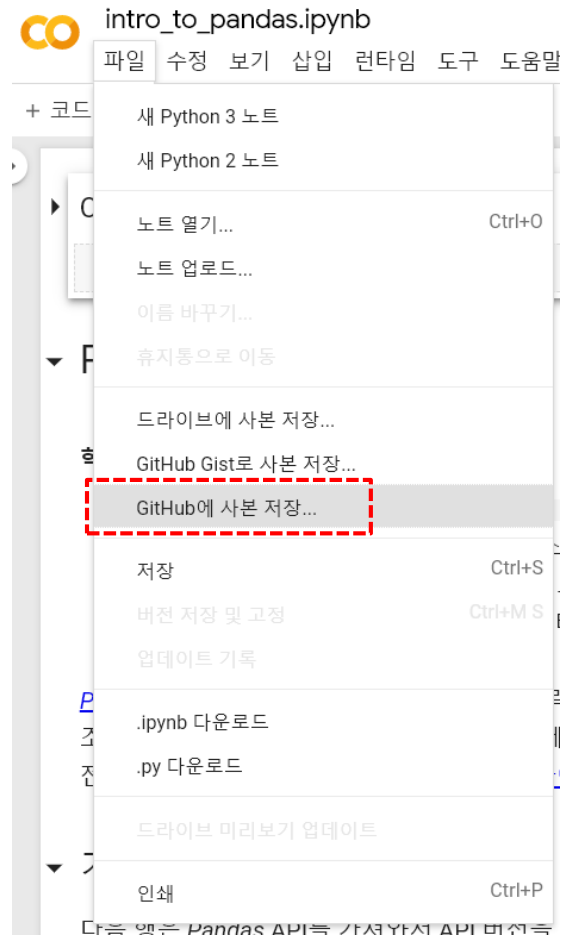
- [Pandas 소개](#)
- [TensorFlow 개념](#)
- [TensorFlow 첫걸음](#)
- [신경망 소개](#)
- [희소 데이터 및 임베딩 소개](#)

가속 하드웨어 사용하기

- [GPU를 사용한 TensorFlow](#)
- [TPU를 사용한 TensorFlow](#)

노트 파일에서 메뉴 선택

- 파일 하부
 - Github에 사본 저장




깃허브의 저장소로 저장

- 깃허브 저장소 입력
 - 인증(로그인) 필요

GitHub으로 복사

저장소:  hskang7/test ▼

브랜치:  master ▼

파일 경로
intro_to_pandas.ipynb

변경사항 설명 메시지
Colaboratory를 통해 생성됨


☒ Colaboratory 링크 추가

취소

확인

GitHub으로 복사

저장소:  git7py/python-ide-ve ▼

브랜치:  master ▼

파일 경로
intro_to_pandas.ipynb

변경사항 설명 메시지
Colaboratory를 통해 생성됨

☒ Colaboratory 링크 추가

취소

확인

자신의 저장소 확인, 예 1

- hskang7/test

hskang7 / test

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Branch: master test / intro_to_pandas.ipynb Find file Copy path

hskang7 Colaboratory를 통해 생성됨 f57e526 5 minutes ago

1 contributor

656 lines (656 sloc) 19.6 KB

Open in Colab

Copyright 2017 Google LLC.

In [0]:

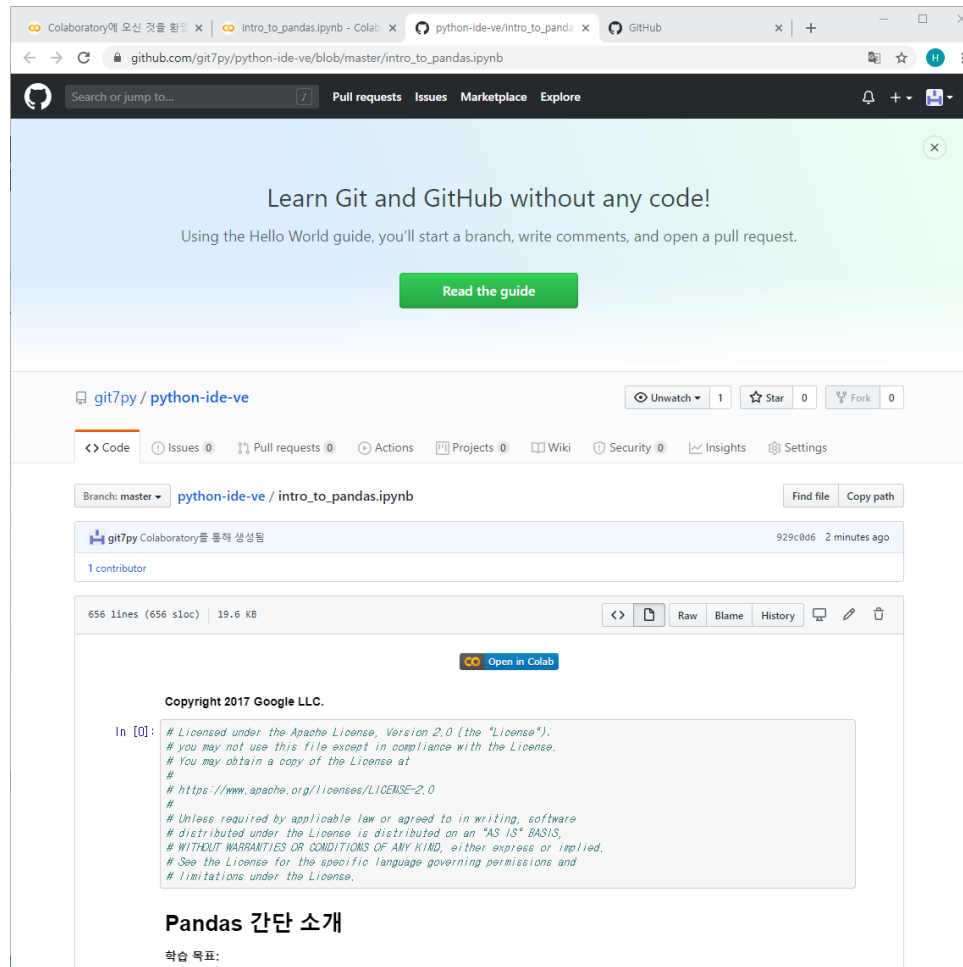
```
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

Pandas 간단 소개

학습 목표:

자신의 저장소 확인, 예 2

- 복사가 성공하면 자동으로 브라우저 활성화



구글 Colab 노트
깃허브 Gist에 저장

Welcome에서 Tensorflow 개념으로 이동

- 다양한 자습 리소스 지원

머신러닝 단기집중과정

다음은 Google 온라인 머신러닝 과정에서 가져온 일부 노트입니다. [전체 과정 웹사이트](#)에서 자세한 내용을 확인하세요.

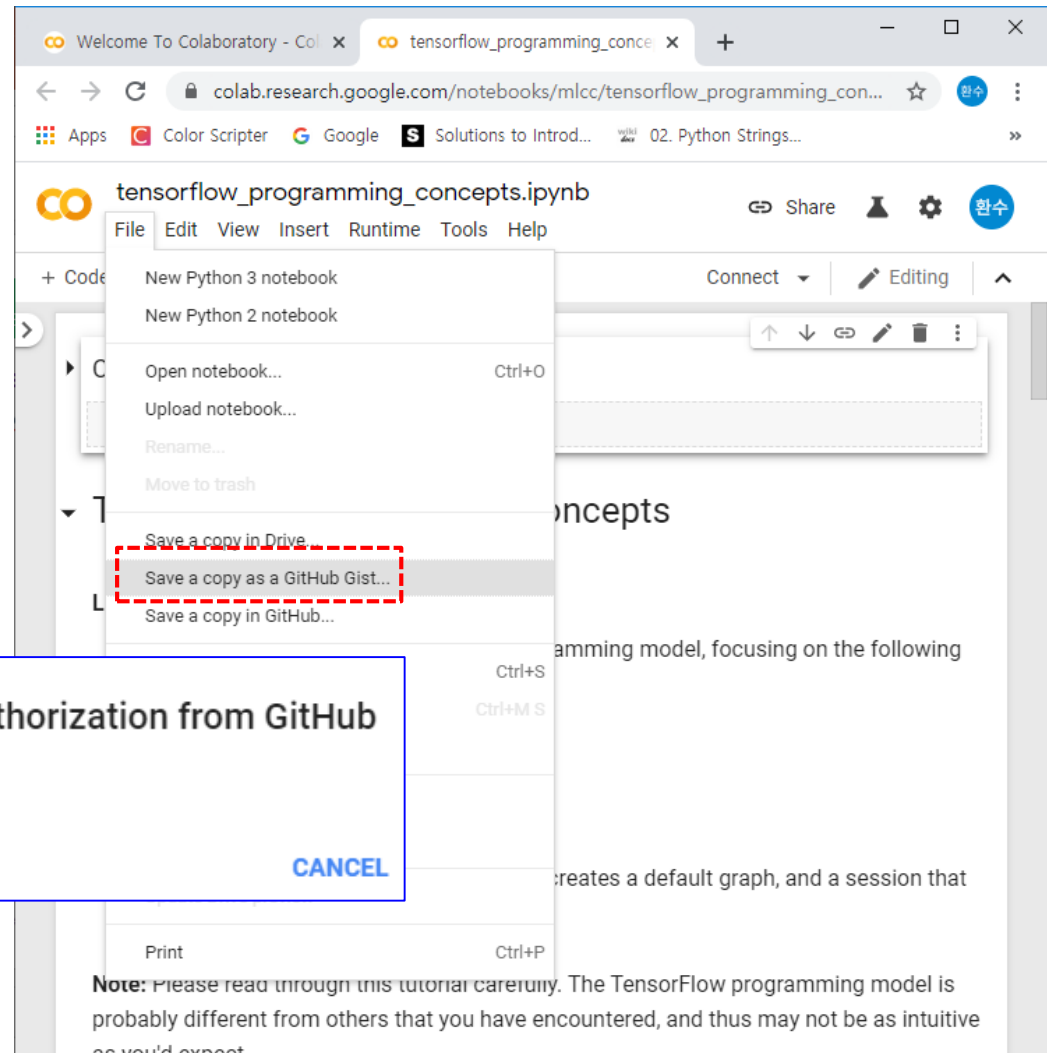
- [Pandas 소개](#)
- [TensorFlow 개념](#)
- [TensorFlow 첫걸음](#)
- [신경망 소개](#)
- [희소 데이터 및 임베딩 소개](#)

가속 하드웨어 사용하기

- [GPU를 사용한 TensorFlow](#)
- [TPU를 사용한 TensorFlow](#)

노트 파일에서 메뉴 선택

- 파일 하부
 - Github Gist로 사본 저장
- 깃허브 인증 필요





깃허브 인증

- 로그인

Sign in to GitHub · GitHub - Google Chrome

github.com/login?client_id=5036cf6d81e65aaa6340&...





Sign in to **GitHub**
to continue to **Colaboratory**

Username or email address

Password [Forgot password?](#)

Sign in

New to GitHub? [Create an account.](#)

자신의 깃허브 Gist에서 확인

- Gist 주소: gist.github.com/사용자id
 – gist.github.com/hskang7

비밀이라 공개는 안됨

Colab에서 소스 열기

hskang7's gists

gist.github.com/hskang7

Apps Color Scripter Google Solutions to Introd... 02. Python Strings... DeepLearningZero... TigerCow.Door :: 텐...

GitHub Gist Search... All gists Back to GitHub

All gists 7

Type: All Sort: Recently created

hskang7 / tensorflow_programming_concepts.ipynb Secret 1 file 0 forks 0 comments 0 stars

Created 6 minutes ago

tensorflow_programming_concepts.ipynb

hskang7

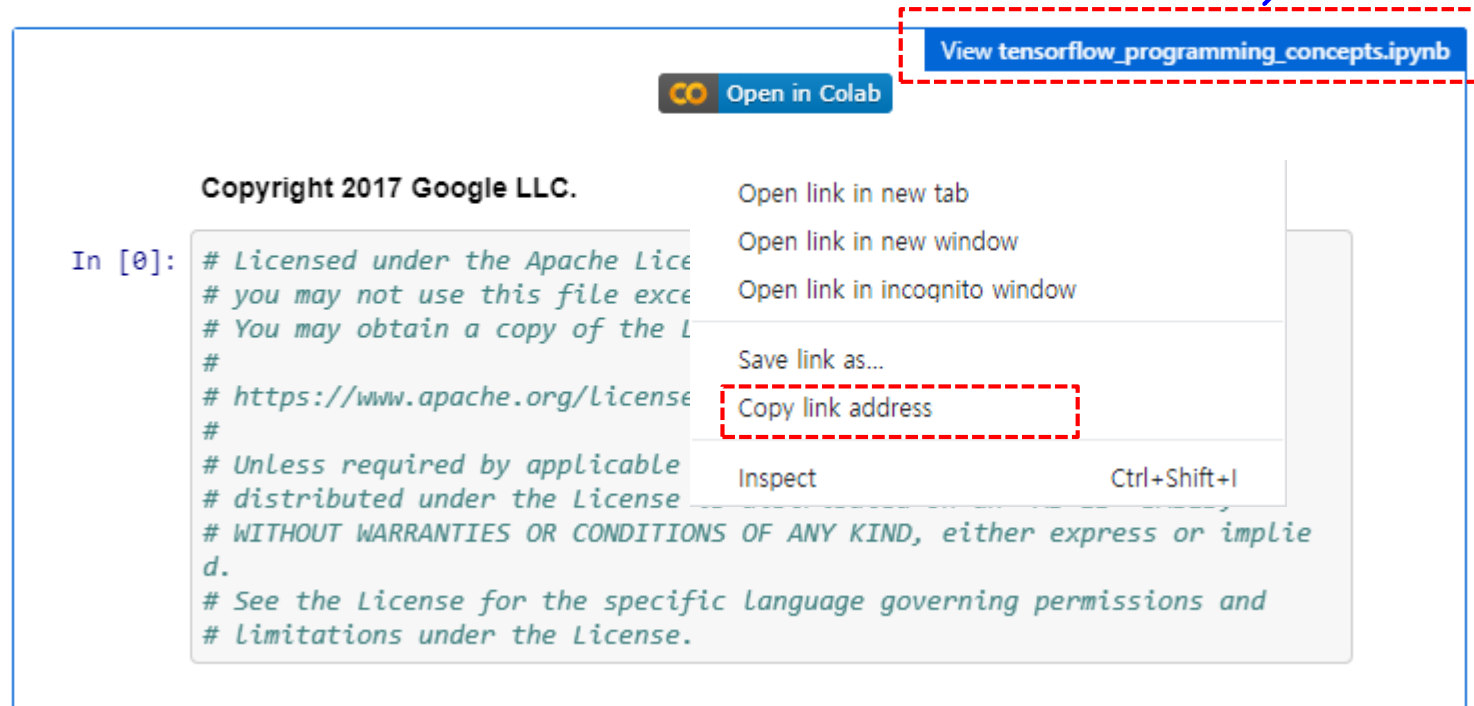
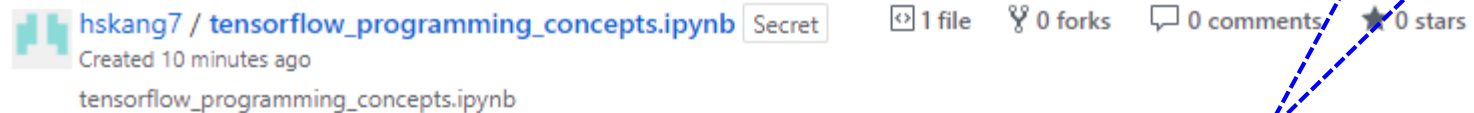
View GitHub Profile

Copyright 2017 Google LLC.

```
In [0]: # Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

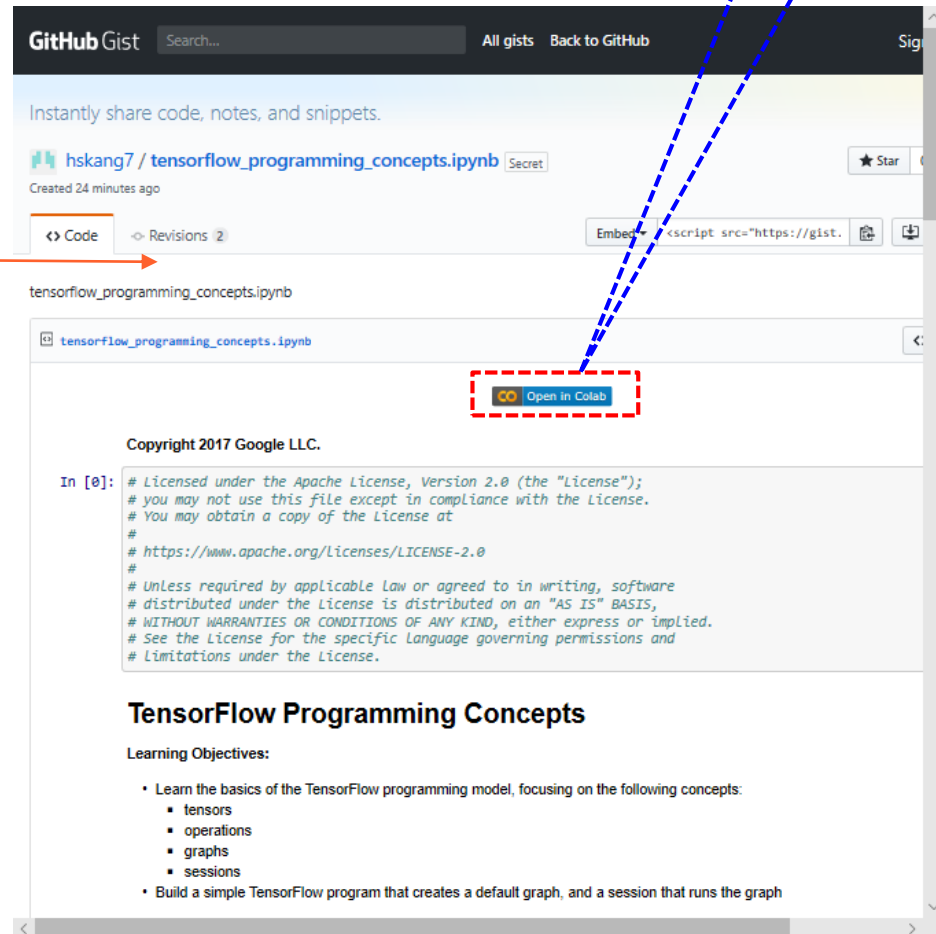
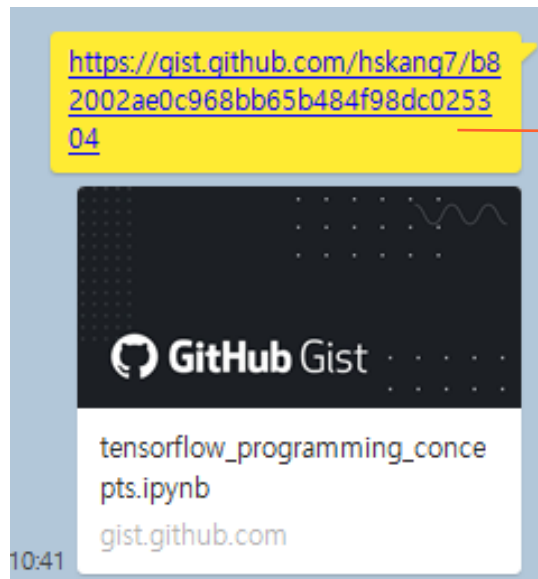
Gist 활용 방법

- 소스에서 오른쪽 팝업 메뉴에서 '링크 주소 복사' 선택



카톡 등에서 복사하면

- 바로 소스로 이동할 수 있는 링크 제공

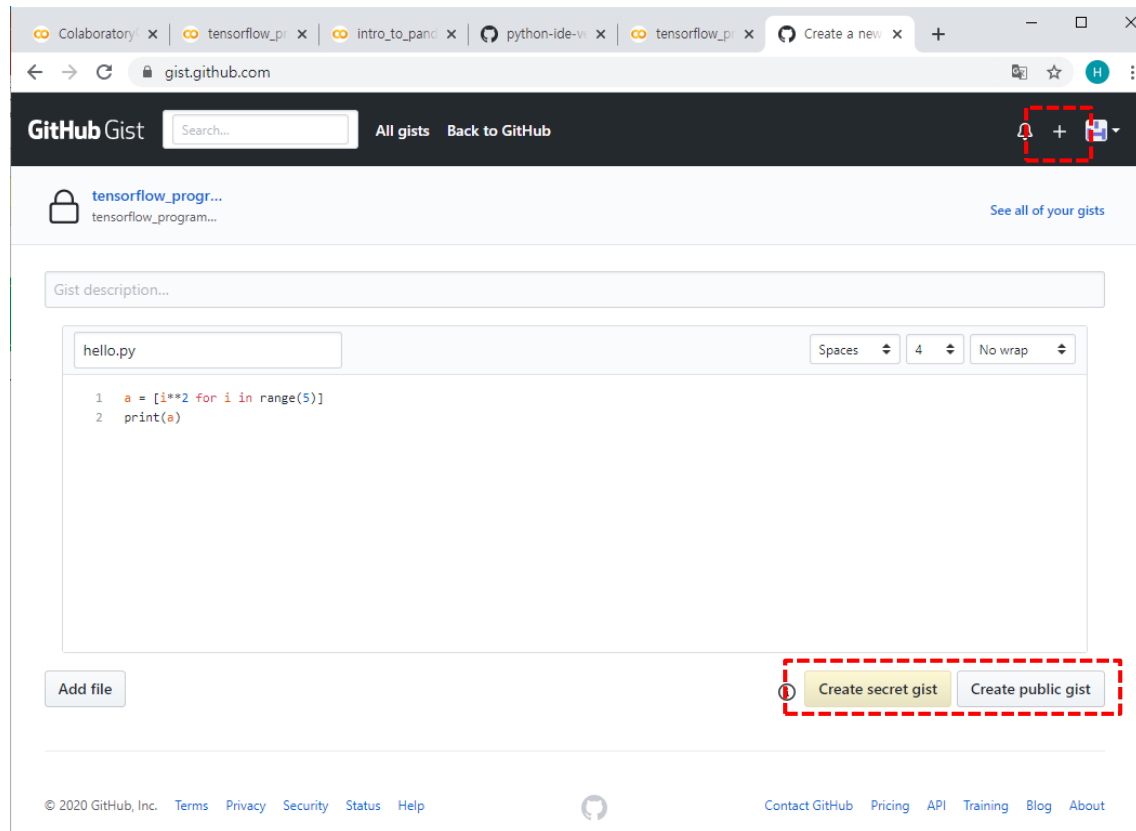


Github Gist 개념

- 깃허브에서 제공하는 링크 공유 서비스
 - 블로그 등에 부분 소스 코드(snippets)로 보기 좋게 붙여 놓는 방법
- Secret과 public
 - Secret
 - 검색 엔진에는 노출이 안되지만 당신이 URL을 알려준 사람에게는 보여주는 수준

간단한 gist 생성

- 파일 이름, 확장자 붙이기



Html 파일에 삽입

- 태그 복사하여 붙여넣기, 브라우저로 확인

The image is a composite of three screenshots illustrating the steps to embed a GitHub Gist into an HTML file and view it.

Top Screenshot: A browser window showing a GitHub Gist page for a file named `hello.py`. The URL is `gist.github.com/git7py/3f313c23f7e9899cdd9f223bb60676`. The page includes buttons for `Edit`, `Delete`, `Unsubscribe`, `Star` (0), and `Download ZIP`. A red dashed box highlights the `Embed` button and the generated `<script src="https://gist.github.com/git7py/3f313c23f7e9899cdd9f223bb60676"></script>` code snippet.

Bottom-Left Screenshot: A Windows Notepad window titled `hello.html - Windows 메모장`. The menu bar includes `파일(F)`, `편집(E)`, `서식(O)`, `보기(V)`, and `도움말(H)`. The text content shows the copied script tag: `<script src="https://gist.github.com/git7py/3f313c23f7e9899cdd9f223bb60676"></script>`. The status bar at the bottom indicates `Ln 1, Col 57` and `100%` zoom.

Bottom-Right Screenshot: A browser window displaying the rendered HTML file. The address bar shows `file:///D:/((0 나스`. A message bar states: `북마크바로 지정된 폴더에 북마크가 없습니다. 북마크를 추가해 보세요.북마크...`. The main content area displays the Python code from the gist:

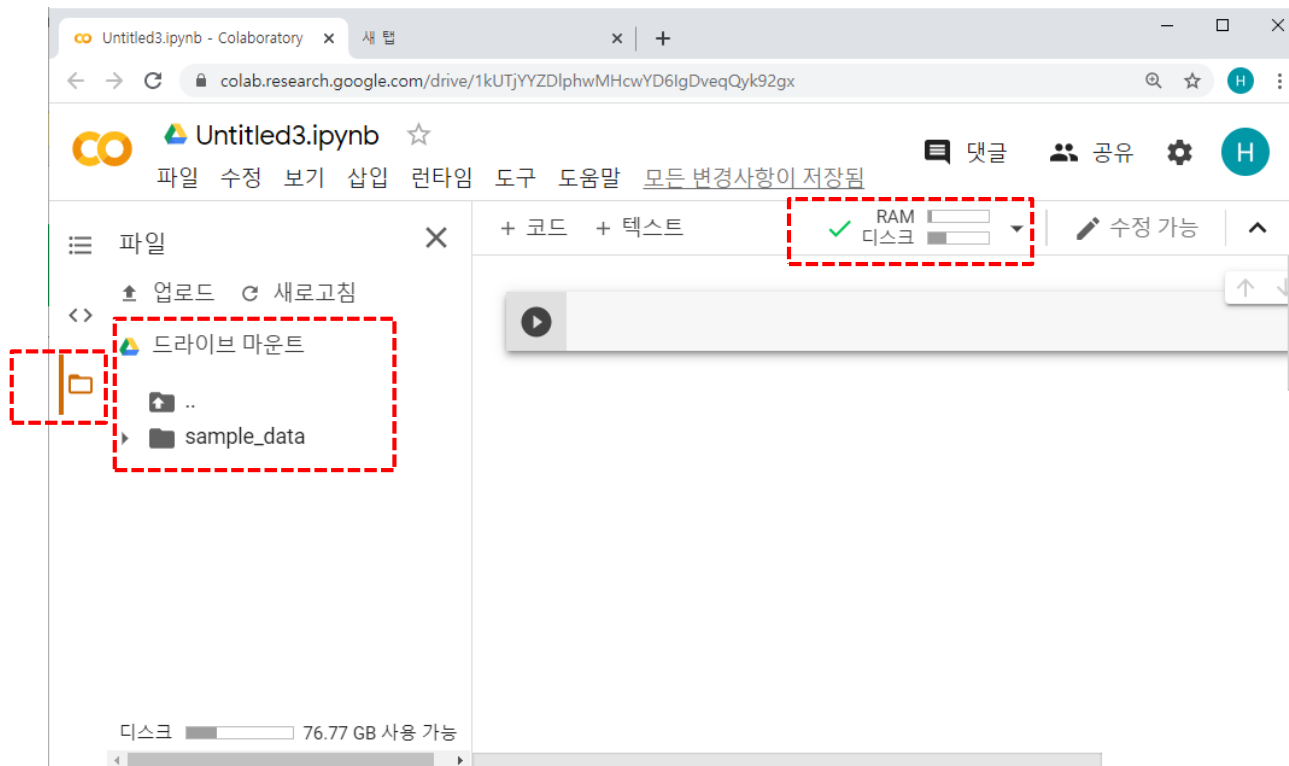
```
1 a = [i**2 for i in range(5)]
2 print(a)
```

 Below the code, it says `hello.py hosted with ❤ by GitHub` with a `view raw` link. The right sidebar contains navigation icons for home, search, and other functions.

Colab 서버의 폴더에
구글 드라이브를
연결(마운트) 하기

시작

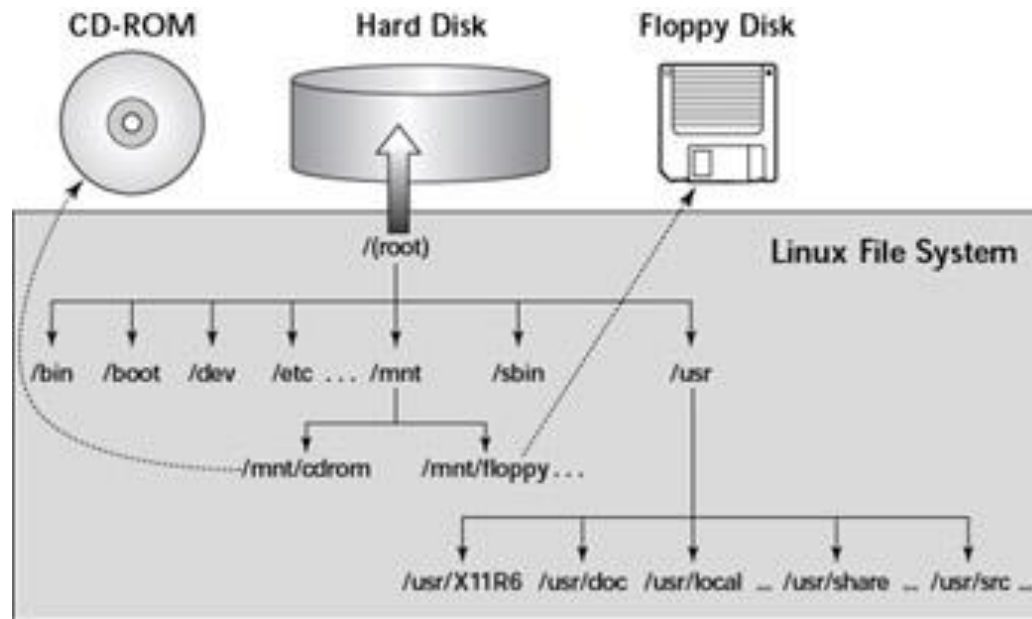
- 메뉴 파일 | 새 노트
 - 연결
 - 왼쪽 폴더 모양 클릭



마운트란?

- **마운트(mount) 의미**

- 물리적인 장치(예로 구글 드라이브)를 특정 위치 즉 디렉터리에 연결시켜주는 것



드라이브 마운트

• 자신의 구글 드라이브에 접속 후

- 왼쪽 드라이브 마운트 클릭

자신의 구글 드라이브를 colab 시스템의 drive 폴더와 연결

The image illustrates the process of mounting Google Drive in Google Colab. It consists of several parts:

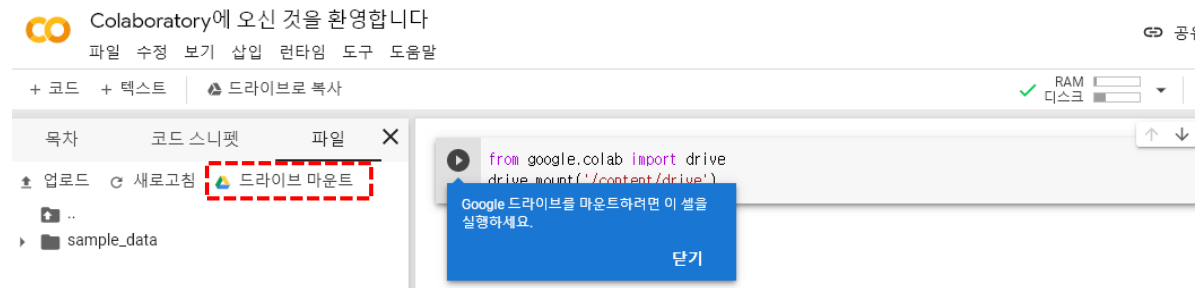
- File Explorer (Left):** Shows the initial state with a '드라이브 마운트' (Mount Drive) button highlighted by a red dashed box. Below it, the 'sample_data' folder is visible.
- Code Editor (Center):** Displays the code to mount the drive:


```
from google.colab import drive
drive.mount('/content/drive')
```
- Confirmation Dialog (Center):** A modal window asking for permission to access Google Drive files. It includes the text: "노트북이 Google 드라이브 파일에 액세스하도록 허용하시겠습니까?" and "Google 드라이브에 연결하면 이 노트북에서 실행된 코드가 Google 드라이브의 파일을 수정할 수 있습니다." The 'GOOGLE 드라이브에 연결' (Connect to Google Drive) button is highlighted with a red dashed box.
- Final File Explorer (Bottom Left):** Shows the state after mounting, with a '드라이브 마운트 해제' (Unmount Drive) button highlighted by a red dashed box. The 'drive' folder is now visible in the file list.
- Bottom Text Box:** A red-bordered box containing the text: "자동으로 연결되나 [아니오]를 누르면 소스로 마운트 할 수 있도록 코드를 삽입해 줌" (If it doesn't connect automatically, pressing [No] allows you to insert code to mount from the source).

드라이브 마운트

구글 드라이브와 서버 연결

- 버튼 '드라이브 마운트' 선택
- 마운트 기능의 스니펫 코드가 삽입

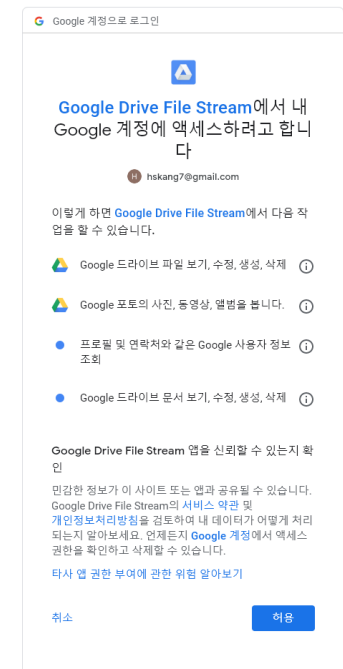
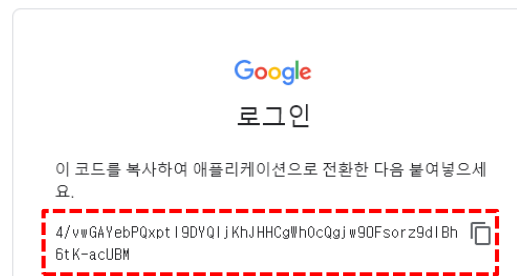
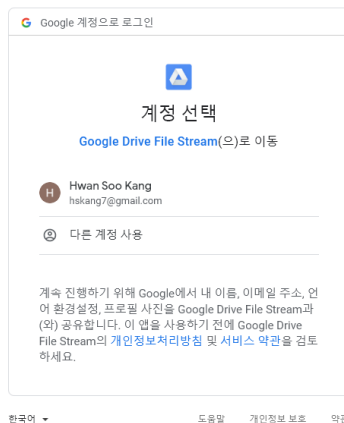


셀 실행 결과

- 인증 코드 복사

```
from google.colab import drive
drive.mount('/content/drive')
```

원하는 이름으로
수정 가능



연결된 드라이브 확인

- 셀 실행 이후
 - 복사된 인증 코드 붙여 넣고 Enter
- 메뉴 새로 고침 확인
 - 폴더 drive 자동 생성
 - 하부 구글의 My Drive 확인 가능



마운트된 구글 드라이브 활용

- 구글 드라이브의 파일을 서버에서 활용 가능
 - 구글 드라이브의 루트인 My Drive에 저장된 파이썬 소스 *.py 파일 실행 가능

The screenshot shows a Jupyter Notebook with three tabs: '목차' (Table of Contents), '코드 스니펫' (Code Snippets), and '파일' (Files). The '파일' tab is active, displaying a file explorer on the left and a code editor on the right.

File Explorer (Left Panel):

- 목차 (Table of Contents)
- 코드로 스니펫 (Code Snippets)
- 파일 (Files)
- 업로드 (Upload)
- 새로고침 (Refresh)
- 드라이브 마운트 (Drive Mount)
- .. (Parent Directory)
- drive (Mounted Drive)
- My Drive (Root of Drive)
- sample_data (Subdirectory)

Code Editor (Right Panel):

```
[1] from google.colab import drive
drive.mount('/content/drive')

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989...
Enter your authorization code:
Mounted at /content/drive

[2] pwd

'/content'

[5] cd drive/My Drive

/content/drive/My Drive

[7] !dir *.py

stringop.py

[8] cat stringop.py

print("원의 원주율 " + '3.141592')
print("python " + 'programming' + ' language')
print('파이썬 언어는' + " 강력하다")
print('파이썬 ' + "언어! " + 3 * "방가 ")

[10] !python stringop.py

원의 원주율 3.141592
python programming language
파이썬 언어는 강력하다
파이썬 언어! 방가 방가 방가
```

Colab 서버의 폴더에
깃허브를 복사 하기

서버에 깃허브 복사

• 명령어

- !git clone https://github.com/neowizard2018/neowizard.git

Colaboratory에 오신 것을 환영합니다
파일 수정 보기 삽입 런타임 도구 도움말 오후 4:58 이후 저장되지 않은 변경사항

+ 코드 + 텍스트 + 드라이브로 복사

RAM 디스크

수정 가능

목차 코드 스니펫 파일 X

업로드 새로고침 드라이브 마운트

..

drive

neowizard

DataAnalysis

data

DA_LEC_01_Example.ipynb

DA_LEC_02_Example.ipynb

DA_LEC_03_Example.ipynb

DA_LEC_04_Example.ipynb

DA_LEC_05_Example.ipynb

DA_LEC_06_Example.ipynb

MachineLearning

README.md

sample_data

[11] cd ..

/content/drive

[12] cd ..

/content

[14] !git clone https://github.com/neowizard2018/neowizard.git

Cloning into 'neowizard'...

remote: Enumerating objects: 12, done.

remote: Counting objects: 100% (12/12), done.

remote: Compressing objects: 100% (12/12), done.

remote: Total 194 (delta 2), reused 0 (delta 0), pack-reused 182

Receiving objects: 100% (194/194), 269.92 MiB | 35.97 MiB/s, done.

Resolving deltas: 100% (62/62), done.

[15] ls

drive/ neowizard/ sample_data/

[16] cd neowizard

/content/neowizard

↑ ↓ ↻ ⚙️ 🗑️ ⋮

!ls

DataAnalysis/ MachineLearning/ README.md

디스크 79.24 GB 사용 가능

마운트된 구글 드라이브 하부 깃허브 복사

• 마운트된 My Drive로 이동

– 복사 명령

- `!git clone https://github.com/hunkim/DeepLearningZeroToAll`

Colaboratory에 오신 것을 환영합니다
파일 수정 보기 삽입 런타임 도구 도움말 오후 4:58 이후 저장되지 않은 변경사항

+ 코드 + 텍스트 + 드라이브로 복사

RAM 디스크

수정 가능

목차 코드 스니펫 파일 X

업로드 새로고침 드라이브 마운트

drive

My Drive

00. 개인

01. 재정

05. 집필

2017 New C로 배우는 프로그래밍 기초

2019 가을 2개의 논문을 위한 설문

Colab Notebooks

DeepLearningZeroToAll

chainer

ipynb

lab-01-basics.ipynb

lab-02-1-linear_regression.ipynb

lab-02-2-linear_regression_fee...

lab-02-3-linear_regression_tens...

lab-03-1-minimizing_cost_sho...

lab-03-2-minimizing_cost_gradi...

lab-03-3-minimizing_cost_tf_o...

lab-03-X-minimizing_cost_tf_gr...

lab-05-1-logistic_regression.ipy...

lab-08-tensor_manipulation.ipy...

디스크 79.23 GB 사용 가능

```
[14] remote: Compressing objects: 100% (12/12), done.
remote: Total 194 (delta 2), reused 0 (delta 0), pack-reused 182
Receiving objects: 100% (194/194), 269.92 MiB | 35.97 MiB/s, done.
Resolving deltas: 100% (62/62), done.

[15] ls

drive/ neowizard/ sample_data/

[16] cd neowizard

/content/neowizard

[17] !ls

DataAnalysis/ MachineLearning/ README.md

[18] cd ../drive/My Drive

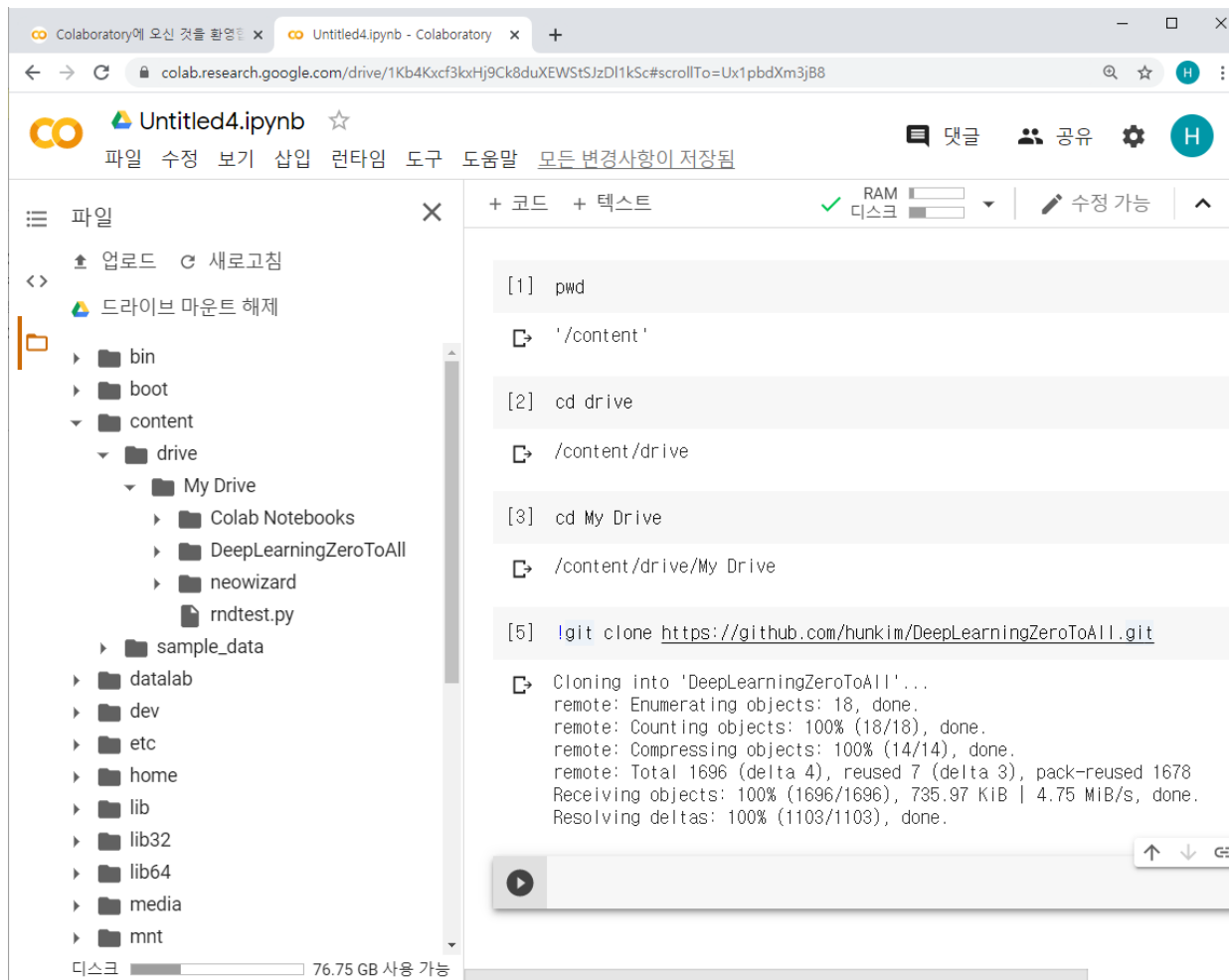
/content/drive/My Drive

[19] !git clone https://github.com/hunkim/DeepLearningZeroToAll

Cloning into 'DeepLearningZeroToAll'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 1688 (delta 2), reused 6 (delta 2), pack-reused 1678
Receiving objects: 100% (1688/1688), 733.45 KiB | 1.74 MiB/s, done.
Resolving deltas: 100% (1101/1101), done.
```

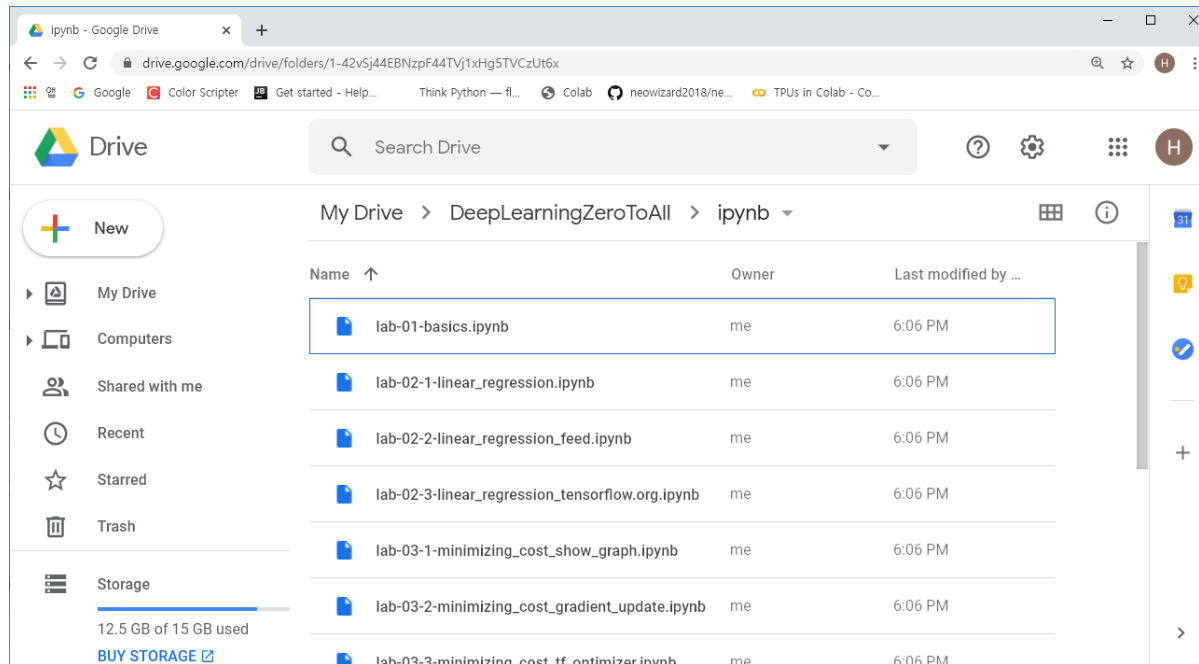
다른 샘플

• My Drive 하부에 바로 복사하는 예



구글 드라이브에서 확인

- 폴더 My Drive/DeepLearningZeroToAll/ipynb
 - 여러 파일 확인
- 파일 lab-01-basic.ipynb
 - Colab으로 열기



Colab으로 연 노트 파일

The screenshot shows a Google Colaboratory notebook interface. The browser address bar indicates the URL: `colab.research.google.com/drive/19U7Z-9irukVD-_Gh6N9VU-OHxEsKLNz`. The notebook title is `lab-01-basics.ipynb`. The interface includes a file manager, a code editor, and an output area.

The notebook content is as follows:

Getting Started With TensorFlow

https://www.tensorflow.org/get_started/get_started

Check TF version

```
[ ] import tensorflow as tf
    tf.__version__
```

Output: `'1.0.0'`

Hello TensorFlow!

```
[ ] # Create a constant op
    # This op is added as a node to the default graph
    hello = tf.constant("Hello, TensorFlow!")

    # start a TF session
    sess = tf.Session()

    # run the op and get result
    print(sess.run(hello))
```

Output: `b'Hello, TensorFlow!'`

Tensors

```
[ ] 3 # a rank 0 tensor: this is a scalar with shape []
    [1., 2., 3.] # a rank 1 tensor: this is a vector with shape [3]
    [[1., 2., 3.], [4., 5., 6.]] # a rank 2 tensor: a matrix with shape [2, 3]
    [[[1., 2., 3.], [7., 8., 9.]]] # a rank 3 tensor with shape [2, 1, 3]
```

필요 패키지 설치

- !pip install tensorflow==1.15.0
 - Start runtime 클릭

```
colab.research.google.com/drive/15N9Khnt4gizy6z-ElQPRJnZV-oLLP-#scrollTo=S6lgBwgs4aP8
lab-01-basics.ipynb
파일 수정 보기 삽입 런타임 도구 도움말 모든 변경사항이 저장됨
+ 코드 + 텍스트
RAM 디스크 수정 가능
Check TF version
[4] !import tensorflow as tf
tf.__version__
'2.2.0'
[5] !pip install tensorflow==1.15.0
Collecting tensorflow==1.15.0
  Downloading https://files.pythonhosted.org/packages/3f/98/5a99af92fb911d7a8aa0005ad5005135b4c1ba8d75fba0d726c936e6/tensorflow-1.15.0-cp36-cp36i-macosx_10_9_universal2.whl (412.3MB)
Requirement already satisfied: wrapt>=1.11.1 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (1.12.1)
Requirement already satisfied: absl-py>=0.7.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (0.9.0)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (1.12.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (3.2.1)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (1.1.0)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (0.34.2)
Requirement already satisfied: numpy>=2.0.1 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (1.18.5)
Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (1.29.0)
Requirement already satisfied: protobuf>=3.6.1 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (3.10.0)
Collecting tensorboard<1.16.0,>=1.15.0
  Downloading https://files.pythonhosted.org/packages/1e/e9/d3d747a97f7188f48aa5eda486907f3b345cd409f0a0850468ba657db246/tensorboard-1.15.0-py3-none-any.whl (3.8MB)
Requirement already satisfied: google-pasta>=0.1.6 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (0.2.0)
Requirement already satisfied: keras-applications>=1.0.8 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (1.0.8)
Collecting tensorflow-estimator==1.15.1
  Downloading https://files.pythonhosted.org/packages/d6/2ee9cd74c9fa21a450877847ba560b260f50fb70ee0595203082dafcc9d/tensorflow_estimator-1.15.1-py3-none-any.whl (512kB)
Requirement already satisfied: astor>=0.6.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (0.8.1)
Collecting gast==0.2.2
  Downloading https://files.pythonhosted.org/packages/4e/35/11749b199c2d4e3cccb4d55ca22590b0d7c2c62b9de38ac4a4714687421/gast-0.2.2.tar.gz
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (1.1.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (47.1.1)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (0.16.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (3.1)
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (2.10.0)
Requirement already satisfied: importlib-metadata: python_version < "3.8" in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (3.10.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.15.0) (3.10.0)
Building wheels for collected packages: gast
  Building wheel for gast (setup.py) ... done
  Created wheel for gast: filename=gast-0.2.2-cp36-none-any.whl size=7540 sha256=ea190f2b454adb0770effde16bfdbadb4b282a66a378190245242dd1c37ed587
  Stored in directory: /root/.cache/pip/wheels/5c/2e/7e/atd4d4f6e6c381f378ce7743a3ced3699f8eb89c6fbdadadd
Successfully built gast
ERROR: tensorflow-probability 0.10.0 has requirement gast>=0.3.2, but you'll have gast 0.2.2 which is incompatible.
Installing collected packages: tensorboard, tensorflow-estimator, gast, tensorflow
  Found existing installation: tensorboard 2.2.2
  Uninstalling tensorboard-2.2.2:
    Successfully uninstalled tensorboard-2.2.2
  Found existing installation: tensorflow-estimator 2.2.0
  Uninstalling tensorflow-estimator-2.2.0:
    Successfully uninstalled tensorflow-estimator-2.2.0
  Found existing installation: gast 0.3.3
  Uninstalling gast-0.3.3:
    Successfully uninstalled gast-0.3.3
  Found existing installation: tensorflow 2.2.0
  Uninstalling tensorflow-2.2.0:
    Successfully uninstalled tensorflow-2.2.0
Successfully installed gast-0.2.2 tensorboard-1.15.0 tensorflow-1.15.0 tensorflow-estimator-1.15.1
WARNING: The following packages were previously imported in this runtime:
[gast, tensorboard, tensorflow]
You must restart the runtime in order to use newly installed versions.
RESTART RUNTIME
```


오류나던 소스가 실행

The screenshot shows a Google Colaboratory notebook titled 'lab-01-basics.ipynb'. The code cell contains the following Python code:

```
# Create a constant op
# This op is added as a node to the default graph
hello = tf.constant("Hello, TensorFlow!")

# start a TF session
sess = tf.Session()

# run the op and get result
print(sess.run(hello))
```

The output shows an `AttributeError` with the message: `AttributeError: module 'tensorflow' has no attribute 'Session'`. The error traceback indicates it occurred in the code cell at line 6. A search bar for 'SEARCH STACK OVERFLOW' is visible at the bottom.

The screenshot shows the same Google Colaboratory notebook, but the code is now running successfully. The output of the code cell is:

```
b'Hello, TensorFlow!'
```

Below the code cell, there is a section titled 'Tensors' which lists various tensor shapes and their corresponding data types:

```
[ ] 3 # a rank 0 tensor; this is a scalar with shape [ ]
[1., 2., 3.] # a rank 1 tensor; this is a vector with shape [3]
[[1., 2., 3.], [4., 5., 6.]] # a rank 2 tensor; a matrix with shape [2, 3]
[[[1., 2., 3.]], [[7., 8., 9.]]] # a rank 3 tensor with shape [2, 1, 3]
```

At the bottom, there is a list of tensors: `[[[1.0, 2.0, 3.0]], [[7.0, 8.0, 9.0]]]`.

구글 Colab 서버 정보 확인

서버 정보 확인 명령

[1] `!cat /etc/issue.net`

↳ Ubuntu 18.04.3 LTS

[2] `cat /etc/issue.net`

↳ Ubuntu 18.04.3 LTS

[4] `!head /proc/cpuinfo`

```
↳ processor      : 0
   vendor_id     : GenuineIntel
   cpu family    : 6
   model         : 79
   model name    : Intel(R) Xeon(R) CPU @ 2.20GHz
   stepping      : 0
   microcode     : 0x1
   cpu MHz       : 2200.000
   cache size    : 56320 KB
   physical id   : 0
```

[6] `!head /proc/meminfo`

```
↳ MemTotal:      13335188 kB
   MemFree:      10804652 kB
   MemAvailable: 12509160 kB
   Buffers:      70536 kB
   Cached:       1790100 kB
   SwapCached:   0 kB
   Active:       692488 kB
   Inactive:     1593188 kB
   Active(anon): 402272 kB
   Inactive(anon): 312 kB
```

[8] `!df -h`

```
↳ Filesystem      Size  Used Avail Use% Mounted on
   overlay         108G   29G   75G   28% /
   tmpfs           64M    0   64M    0% /dev
   tmpfs           6.4G    0   6.4G    0% /sys/fs/cgroup
   tmpfs           6.4G   8.0K   6.4G    1% /var/colab
   /dev/sda1       114G   30G   85G   26% /etc/hosts
   shm             6.0G   4.0K   6.0G    1% /dev/shm
   tmpfs           6.4G    0   6.4G    0% /proc/acpi
   tmpfs           6.4G    0   6.4G    0% /proc/scsi
   tmpfs           6.4G    0   6.4G    0% /sys/firmware
```

[9] `pwd`

↳ `'/content'`

[10] `ls -al`

```
↳ total 16
   drwxr-xr-x 1 root root 4096 Jan 13 16:38 ./
   drwxr-xr-x 1 root root 4096 Jan 21 08:43 ../
   drwxr-xr-x 1 root root 4096 Jan 13 16:38 .config/
   drwxr-xr-x 1 root root 4096 Jan 13 16:38 sample_data/
```

[12] `!python --version`

↳ Python 3.6.9

[13] `import tensorflow as tf`
`print(tf.__version__)`

```
↳ The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
   We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via
   the %tensorflow_version 1.x magic: more info.
   1.15.0
```