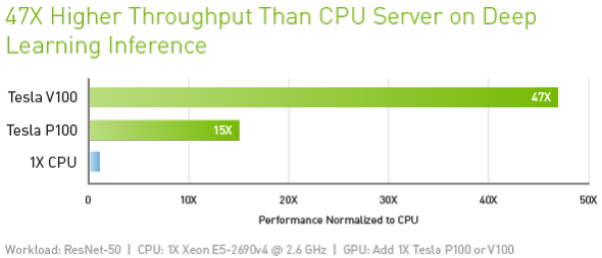
Introduction to Google Colab

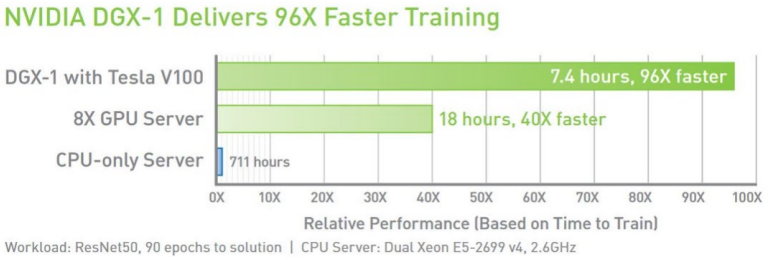
2022.04.26

Jonghwa lee

Deep learning requires a lot of hardware

- To efficiently train a deep learning model, your hardware should be able to deal with **a huge amount of data at high speed**
- CPUs are fast but their bandwidth is lower than GPUs
  - GPUs can perform a large number of matrix multiplications at a time
  - It allows you to build and train complex deep learning models on large dataset efficiently
- However, high-performance GPUs are expensive



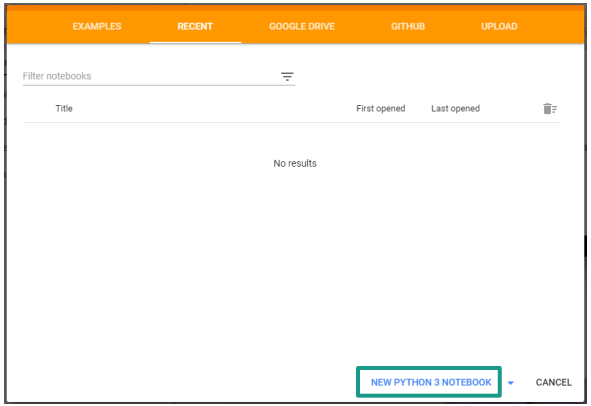


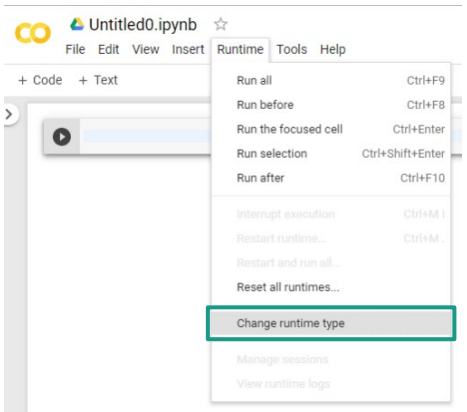
Google Colab to the rescue!

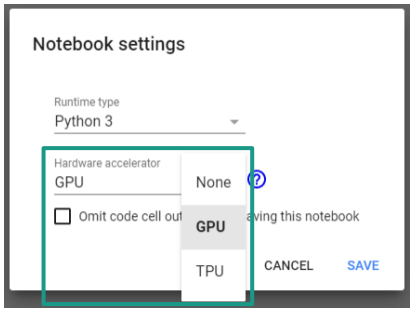
- Google Colab (https://colab.research.google.com) is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud
- It currently offers the computing services of an NVIDIA Tesla K80 GPU for free
- You can use the computing services for a maximum of 12 hours at a time
  - Basically, you are connected to a virtual machine that lives for a maximum of 12 hours at a time
  - All data including model parameters that aren't saved to the Google Drive before this period will be lost

Getting Started

- Go to https://colab.research.google.com
- Create a new Python 3 notebook
- From the '*Runtime > Change runtime type*' menu, assign a hardware accelerator to your notebook

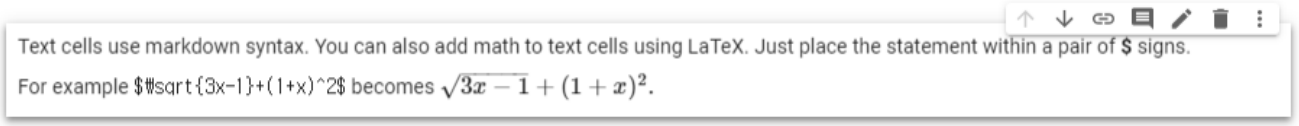




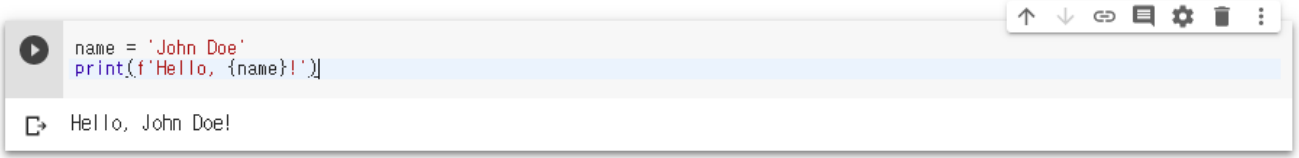


Notebook

- Notebook is **a list of cells** which contain either **explanatory text** or **executable code and its output**
- By clicking a cell, you can select the cell that you want to working with
- Text cells help you explain your notebook and code cells you wrote

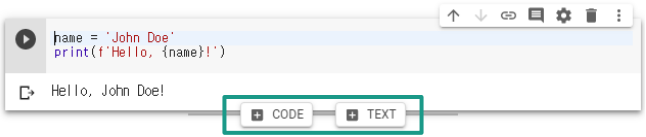


- Code cells contain executable Python code and its output

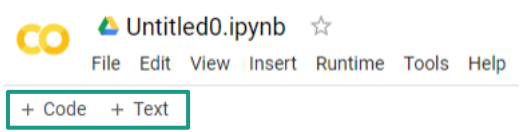


Working with Cells

- You can add new code or text cells using the **+CODE** or **+TEXT** buttons

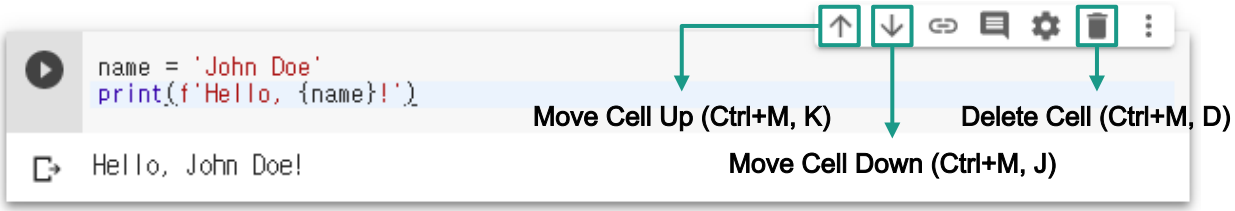


Buttons that appear when you hover between cells



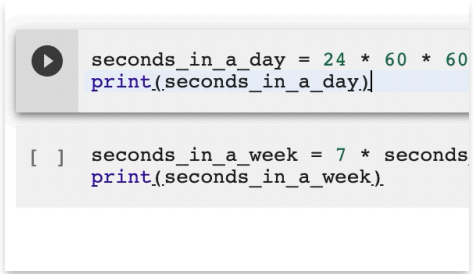
Buttons in the notebook toolbar

- By selecting a cell, you can move up, move down or delete the cell

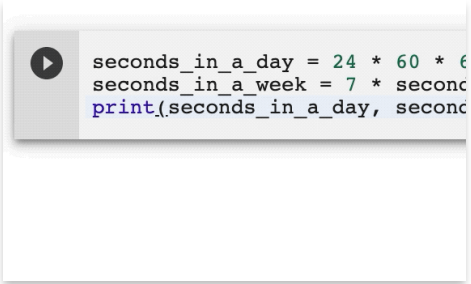


Working with Cells

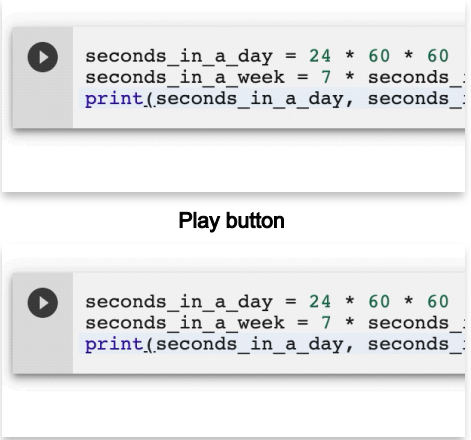
- You can execute a selected code cell in the following ways:
  - Click the **Play button** in the left of the cell
  - Type **Ctrl+Enter** to run the cell in place
  - Type **Shift+Enter** to run the cell and move focus to the next cell
  - Type **Alt+Enter** to run the cell and insert a new code cell immediately below it



Shift + Enter



Alt + Enter

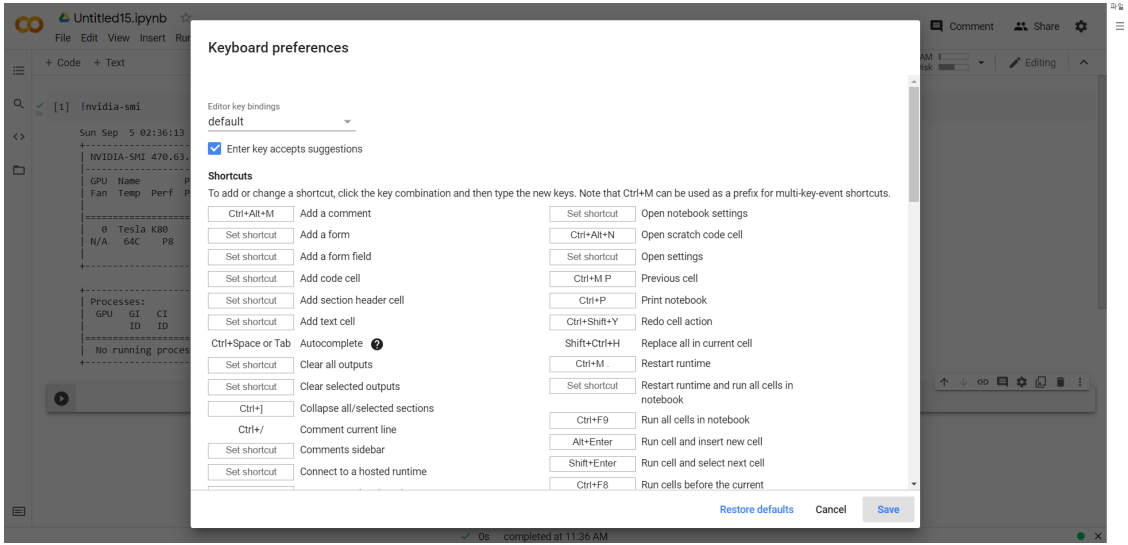


Play button

Ctrl + Enter

Working with Cells

- You can check list of keyboard shortcuts by typing ctrl + m + h



Working with Cells

- You can run any command on the system shell by prefixing it with `!`

```
[1]  !ls /
```

```
bin      datalab  home   lib64  opt   run   swift                tmp    var
boot     dev      lib    media  proc  sbin  sys                  tools
content  etc      lib32  mnt    root  srv   tensorflow-2.0.0-rc0  usr
```

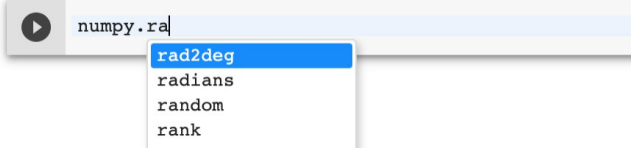
- You can install third-party libraries using package managers as needed

```
[ ]  !apt install -y openjdk-11-jdk
     !pip install tqdm
```

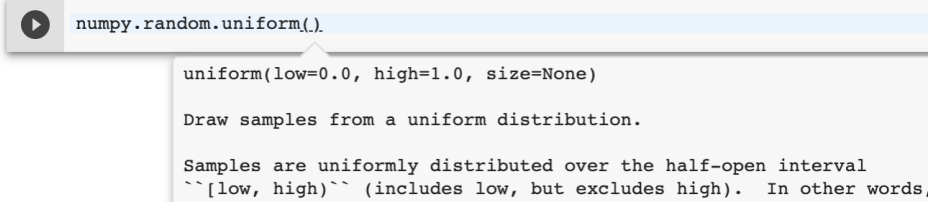
```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  openjdk-11-jdk-headless
Suggested packages:
  openjdk-11-demo openjdk-11-source visualvm
```

Working with Cells

- You can use tab completion to explore attributes of Python objects, as well as to quickly view documentation strings
  - As an example, if you press Tab after `numpy.ra`, you will see the list of available completions starting with `ra` within the `numpy` module

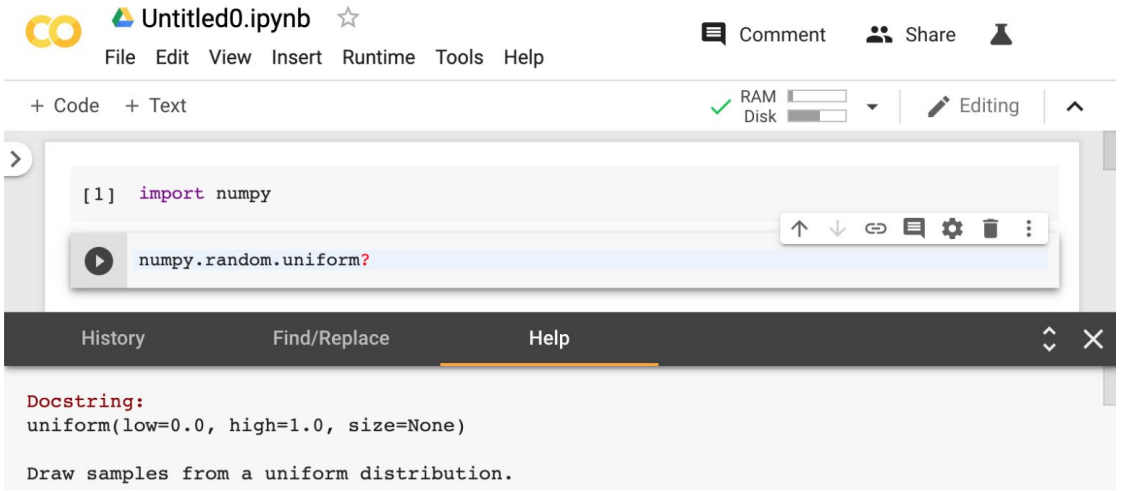


  - You can see a popup of its documentation string



Working with Cells

- To open the documentation in a persistent pane at the bottom of your screen, add ? after the object or method name and run the cell



Google drive in Colab

- You can mount your google drive in Colab
  - If you do NOT use the google drive, you may lose your all data and results when Colab's session expired.

Access to your google drive in Colab

```
from google.colab import drive

drive.mount('/gdrive')
```

```
# check what's in the mounted gdrive using Colab
import os

gdrive_root = '/gdrive/My Drive'
print('In gdrive :', os.listdir(gdrive_root))

notebook_dir = os.path.join(gdrive_root, 'ColabNotebooks')
print('In Colab Notebooks :', os.listdir(notebook_dir))
```

Fetching Notebooks from GitHub

- You can fetch notebooks from GitHub repositories by searching them
  - In this lecture, we will upload all the notebooks to repositories by keai - kaist