

# 情報管理

## 第 1 回：Python環境の準備

# 今回の講義内容

以降の講義で使っていくPythonプログラミングの開発環境の準備と、Pythonプログラミングの基本について説明します。

具体的には以下の内容を説明します。

- Pythonの基本
- Jupyter Notebook
- Google colaboratory

レポートでは、Jupyter Notebook を動かしてもらい、簡単なプログラムとテキスト入力を行ってもらいます。

# Pythonとは？

人工知能やメディア情報処理分野で良く使われるプログラミング言語です。



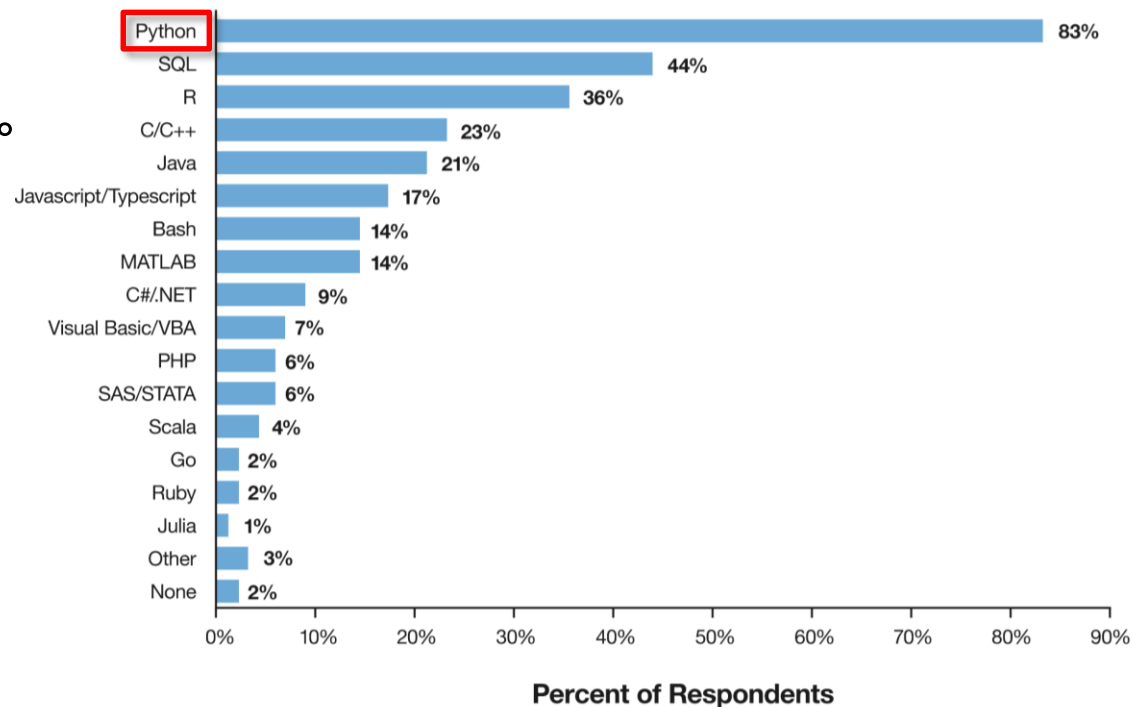
多くのデータは多次元のベクトルで表現されます。  
Pythonを使うとベクトルを使った計算が、とても簡単にできます。

Pythonプログラムについて少し勉強し、  
簡単なプログラムを作れるようになりましょう。

※本講義では、あくまでレポート課題をするための  
必要最低限の部分のみを教えます。  
Pythonの専門的な話は講義の対象外とします。

データサイエンティストにアンケートした、  
AI実装に用いるプログラミング言語の比率

(出典: <https://c3.ai/introduction-what-is-machine-learning/programming-languages-for-machine-learning/>)

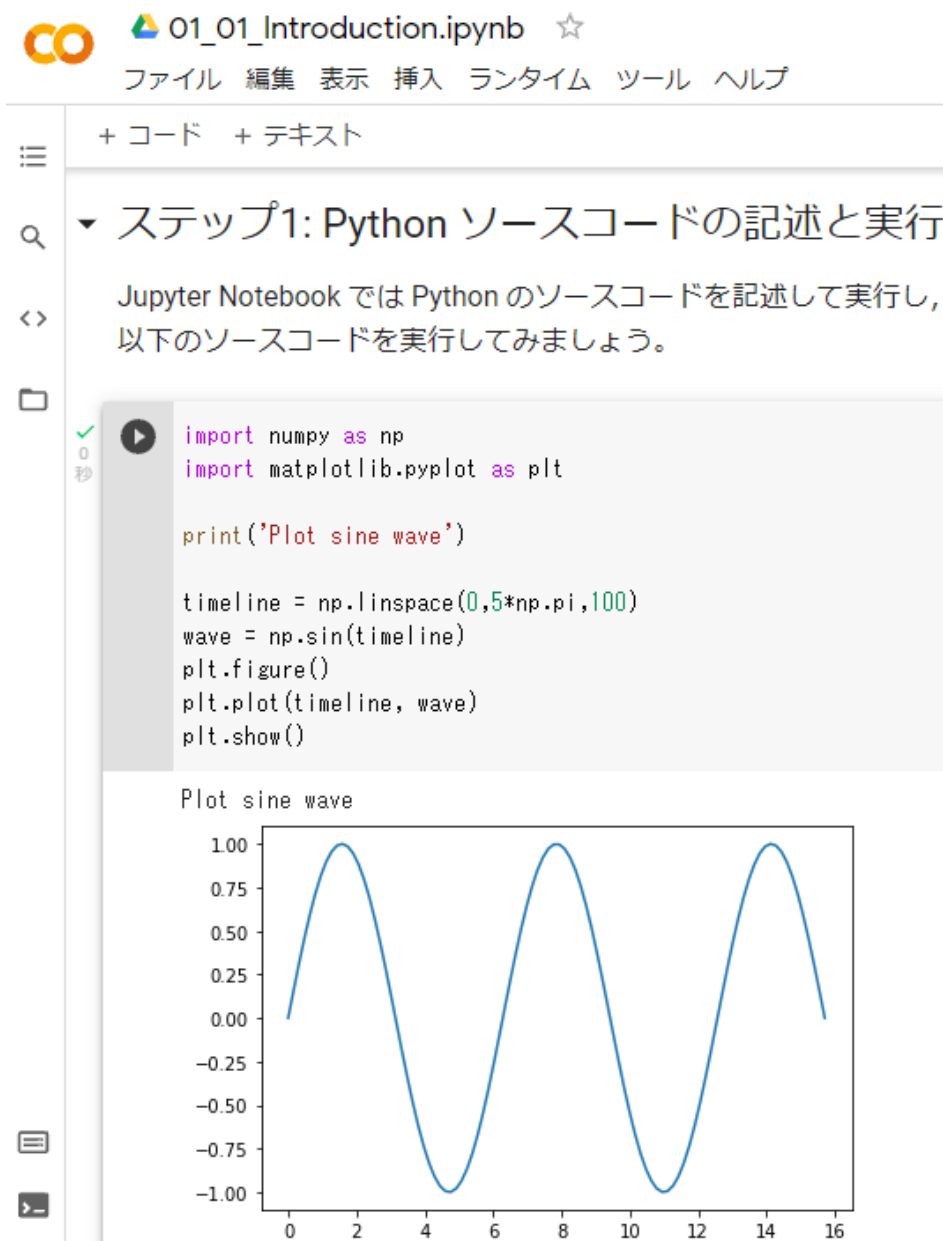


# Jupyter Notebook とは？

Pythonのソースコードと実行結果、  
またテキストメモなどを記録することが  
できるツールです。

特にPythonを用いた実験を行う場合に  
メモや実行結果を残せるので、便利な  
ツールです。

※ 正確には、本講義で使うのはGoogle colabが提供する  
Colabノートブックで、Jupyter Notebookではありません。  
基本的には Colabノートブックも Jupyter Notebook も使い方は同じです。



# Google colaboratory (Google colab) とは？

Google が提供しているオンラインの Python 開発環境です。

オンライン上で Jupyter Notebook (のColabバージョン)を利用できます。

PC のスペックに依存せず、また環境構築をスキップして、ブラウザ上で Pythonを動かせるので、本講義ではGoogle colab を使用します。

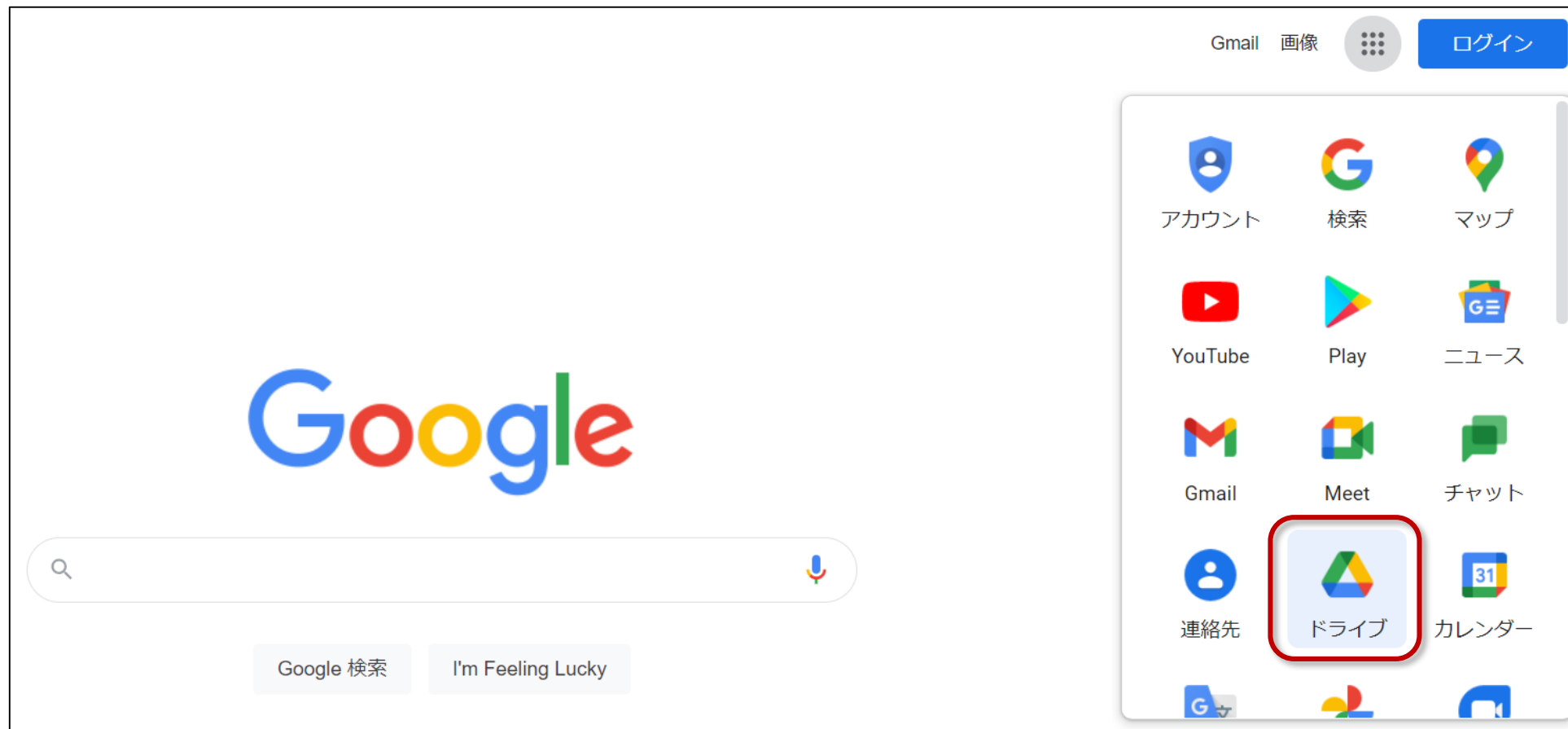
自身の PC に Jupyter Notebook をインストールしたい/既になっているという方は、そちらを使って演習に取り組んでも結構です。（ただしサポート外です。）



# Google colab の導入

# Google Colaboratory 導入

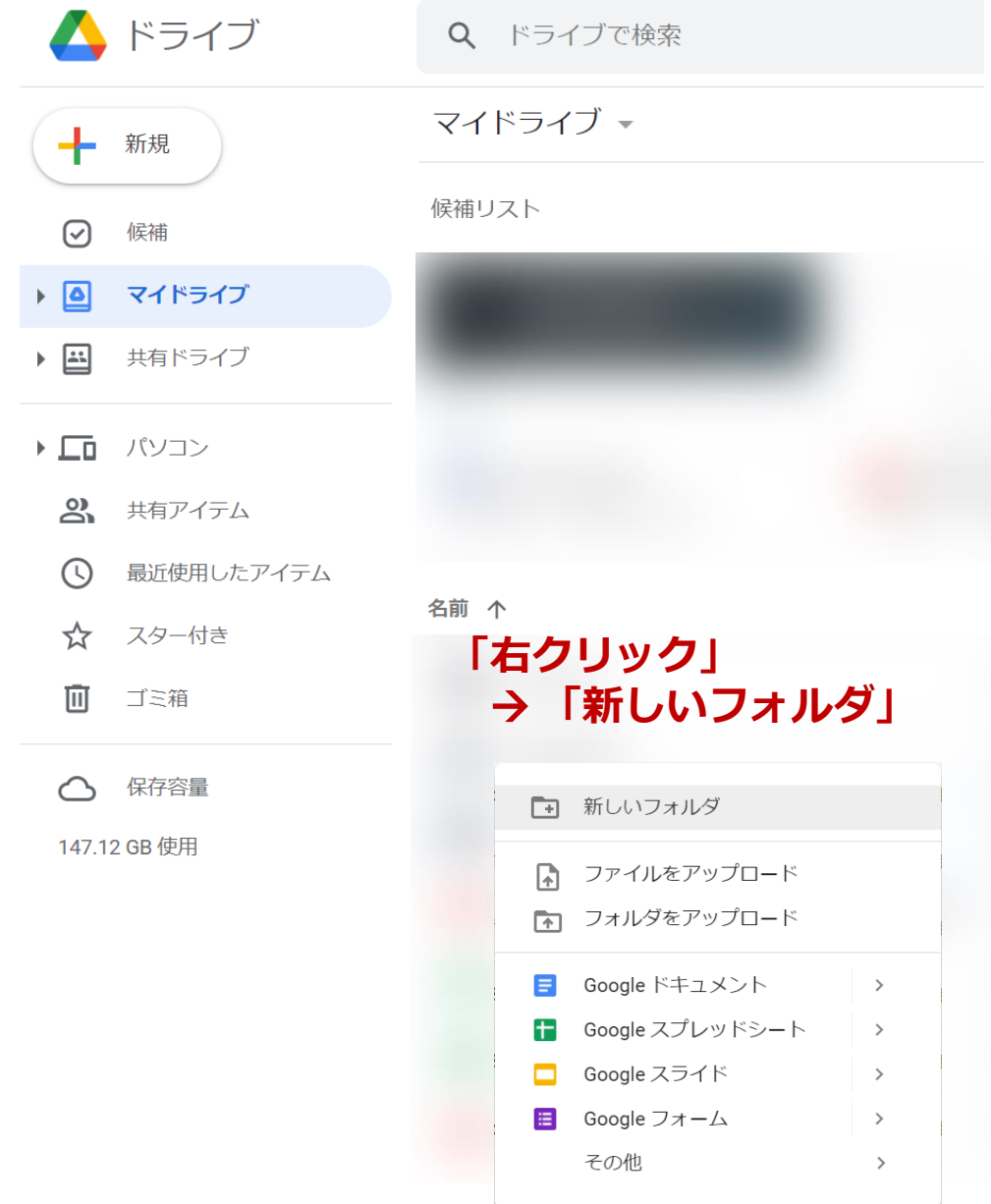
Gmailアカウントで Google ドライブにログインする



# Google Colaboratory 導入

Google ドライブに本講義の専用フォルダを作成する。

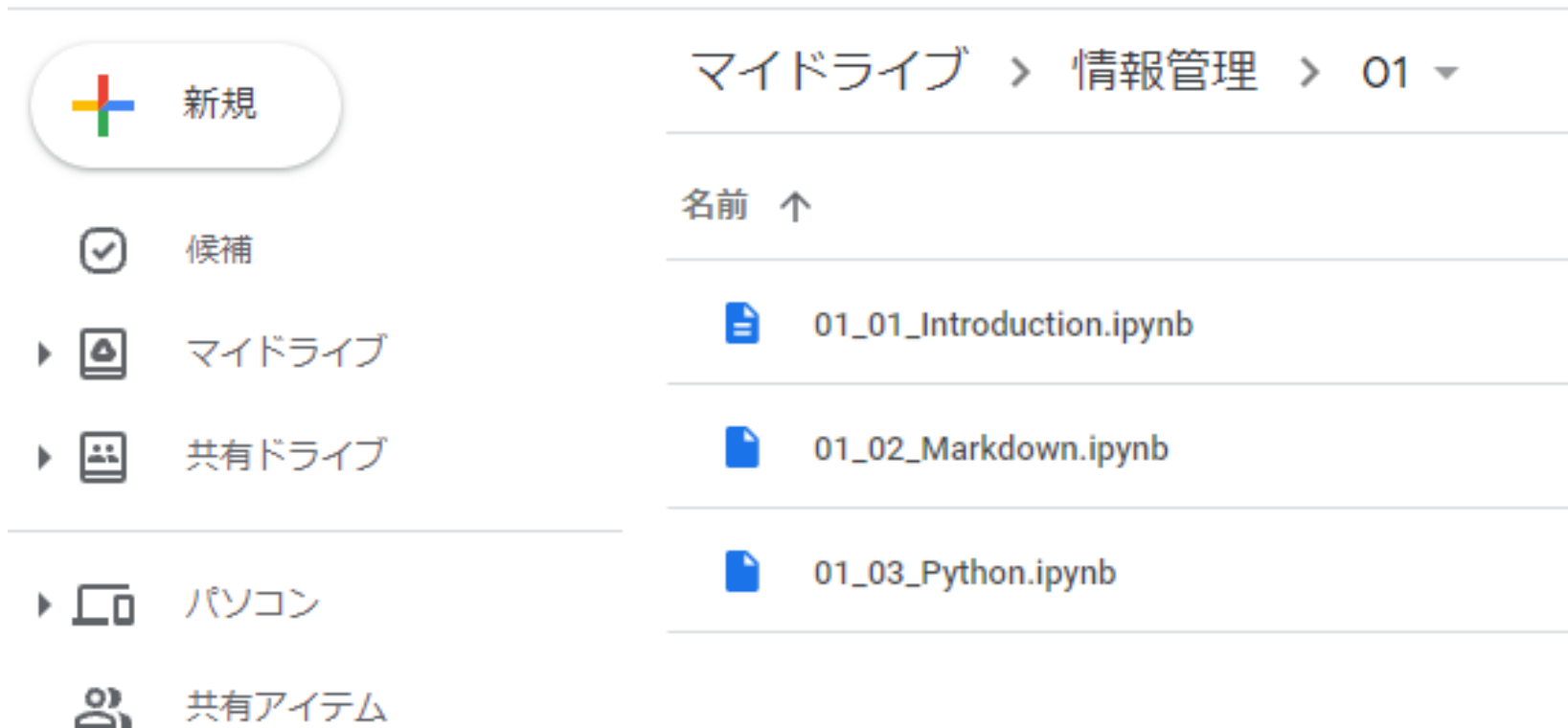
例えば「情報管理」など。





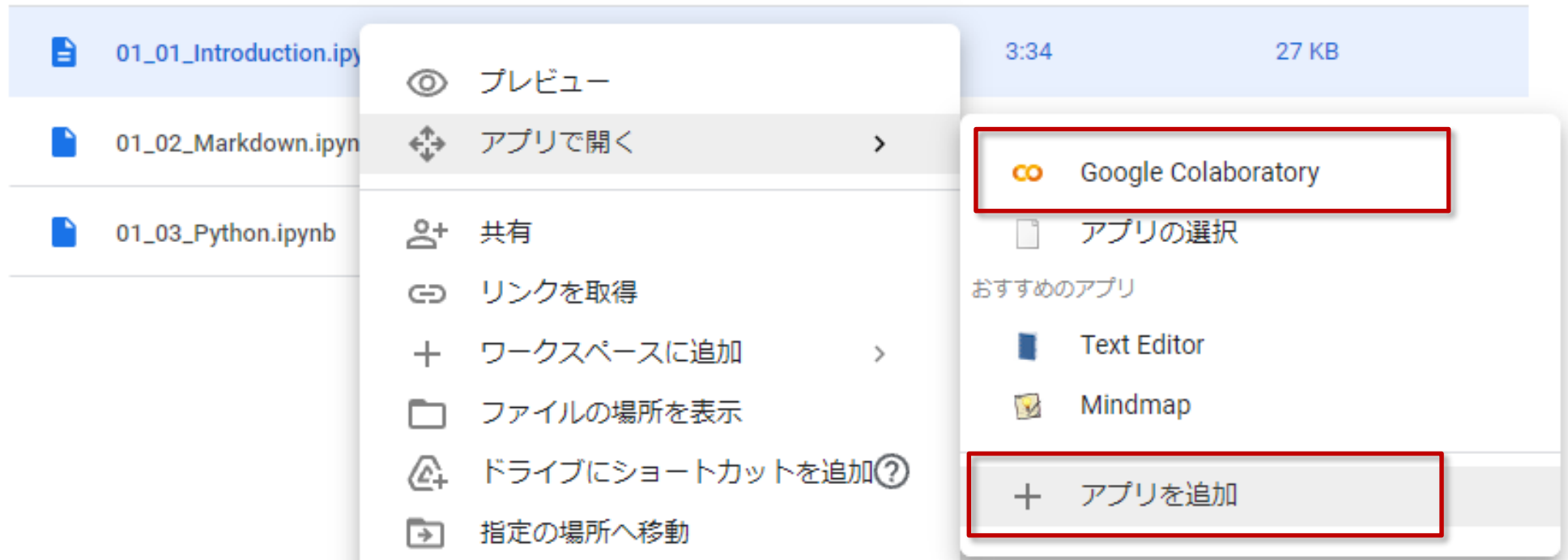
# Google Colaboratory 導入

BEEFにアップロードされているプログラム一式をフォルダに入れる。  
例では「マイドライブ → 情報管理 → 01」フォルダに入れている。



# Google Colaboratory 導入

- ファイルを右クリックして「アプリで開く」。
- 選択肢に「Google Colaboratory」があれば選択する。
- 無ければ、「アプリの追加」を選択する。



# Google Colaboratory 導入

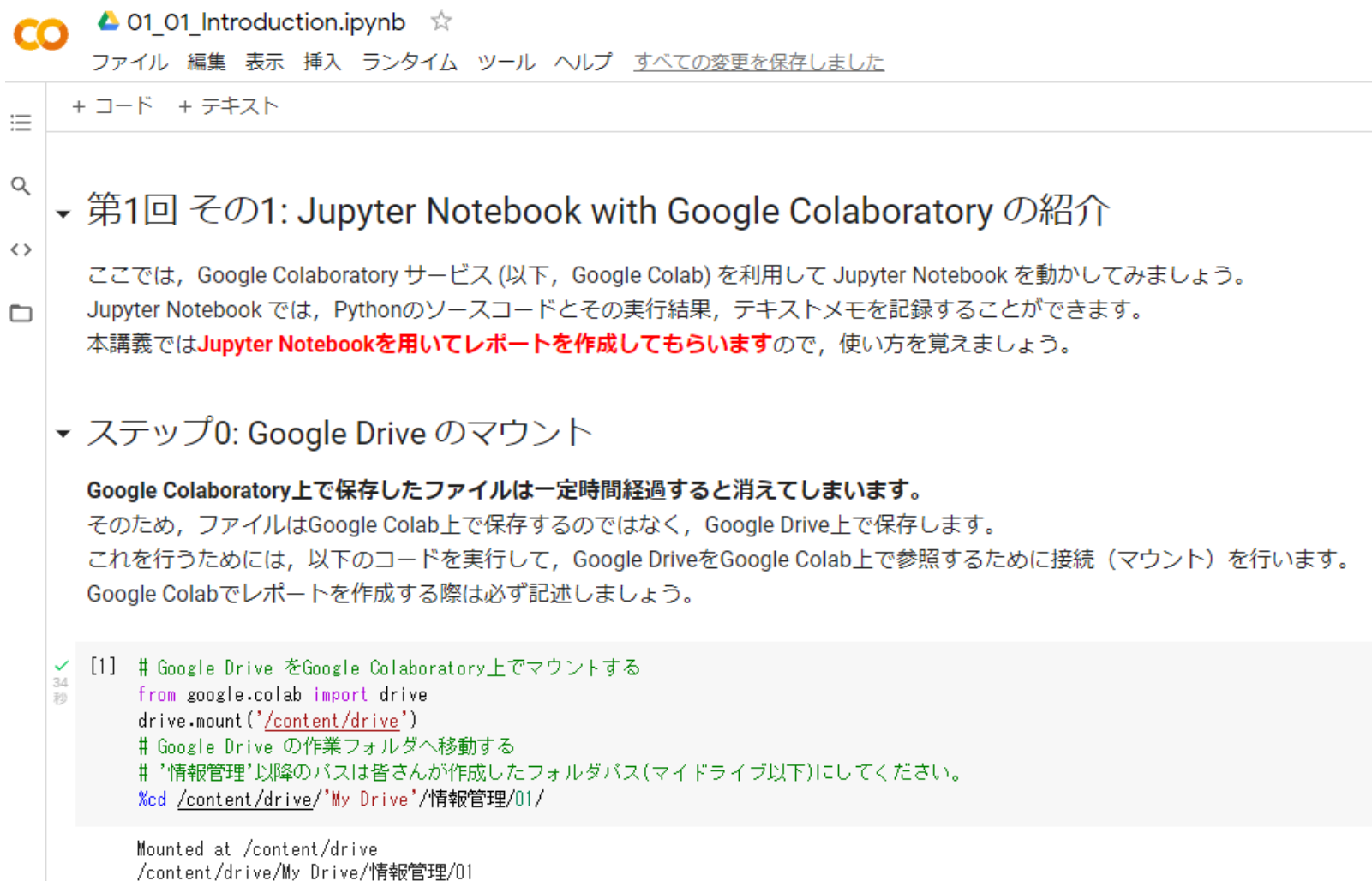
「アプリの追加」を選択した場合は、「Colaboratory」をインストール

The image shows a side-by-side comparison of the Google Workspace Marketplace interface. On the left, a grid of various apps is displayed, with the 'Colaboratory' app highlighted by a red rectangular border. The app's icon, a large orange 'CO', is prominent. Below the icon, the text reads 'Colaboratory', 'colab-team', and 'Colab is a Jupyter notebook environment that runs in the browser using Google Cloud.' The app has a 4.7-star rating and over 8 million downloads. Other visible apps include MindMap 2, Lucidpress K12, Grackle Docs, Ganttter, and Asana.

On the right, the full details page for the 'Colaboratory' app is shown. It features the 'Colaboratory' logo and a blue 'インストール' (Install) button. The description states: 'Colab is a Jupyter notebook environment that runs in the browser using Google Cloud.' The developer is listed as 'colab-team'. The app has a 4.7-star rating (3,425 reviews) and 8,280,898 downloads. It is compatible with Google Cloud. Below the details, there are tabs for '概要' (Overview), '権限' (Permissions), and 'レビュー' (Reviews). The '概要' tab is selected, showing a 'Welcome To Colaboratory' message and a preview of the Colab interface. The preview shows a menu with options like 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help', and a main area with the text 'What is Colaboratory?' and a list of features: 'Zero configuration required', 'Free access to GPUs', and 'Easy sharing'.

# Google Colaboratory 導入

Colaboratoryでファイルを開くと、Notebookの画面が表示されます。



The screenshot shows the Google Colaboratory interface for a notebook titled "01\_01\_Introduction.ipynb". The top bar includes the Google Colab logo, the file name, and a star icon. Below the bar, there are tabs for "ファイル" (Files), "編集" (Edit), "表示" (View), "挿入" (Insert), "ランタイム" (Runtime), "ツール" (Tools), and "ヘルプ" (Help). A notification says "すべての変更を保存しました" (All changes saved). The left sidebar has icons for a menu, search, code, and files. The main area shows a Jupyter Notebook with two sections: "第1回 その1: Jupyter Notebook with Google Colaboratory の紹介" and "ステップ0: Google Drive のマウント". The first section contains introductory text about Google Colab and Jupyter Notebook. The second section contains a code cell with Python and shell commands to mount Google Drive and navigate to a specific folder. The code cell is marked as successful with a green checkmark and a timer showing 34 seconds.

01\_01\_Introduction.ipynb ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ すべての変更を保存しました

+ コード + テキスト

▼ 第1回 その1: Jupyter Notebook with Google Colaboratory の紹介

ここでは、Google Colaboratory サービス (以下、Google Colab) を利用して Jupyter Notebook を動かしてみましょう。

Jupyter Notebook では、Pythonのソースコードとその実行結果、テキストメモを記録することができます。

本講義では**Jupyter Notebookを用いてレポートを作成してもらいます**ので、使い方を覚えましょう。

▼ ステップ0: Google Drive のマウント

**Google Colaboratory上で保存したファイルは一定時間経過すると消えてしまいます。**

そのため、ファイルはGoogle Colab上で保存するのではなく、Google Drive上で保存します。

これを行うためには、以下のコードを実行して、Google DriveをGoogle Colab上で参照するために接続（マウント）を行います。

Google Colabでレポートを作成する際は必ず記述しましょう。

```
[1] # Google Drive をGoogle Colaboratory上でマウントする
from google.colab import drive
drive.mount('/content/drive')
# Google Drive の作業フォルダへ移動する
# '情報管理'以降のパスは皆さんが作成したフォルダパス(マイドライブ以下)にしてください。
%cd /content/drive/My Drive/情報管理/01/
```

Mounted at /content/drive  
/content/drive/My Drive/情報管理/01

# Notebook の使い方

以下の内容は 01\_01\_Introduction.ipynb と 01\_02\_Markdown.ipynb  
でも解説していますので、実際に動かしながら  
確認することをお勧めします。

# Notebookの構成

Notebook には「コードセル」と「テキストセル」が存在します。

メニューバー下の + コード + テキスト をクリックするとセルを挿入できます。

セルを削除したい場合はゴミ箱ボタン  をクリックします。

## ▼ ステップ1: Python ソースコードの記述と実行 **テキストセル：テキストを記述する**

Jupyter Notebook では Python のソースコードを記述して実行し、その結果を記録することができます。  
以下のソースコードを実行してみましょう。

```
✓ [2] import numpy as np
import matplotlib.pyplot as plt

print('Plot sine wave')

timeline = np.linspace(0, 5*np.pi, 100)
wave = np.sin(timeline)
plt.figure()
plt.plot(timeline, wave)
plt.show()
```

## **コードセル：Pythonコードを記述する**

# Python ソースコードの実行

- 再生ボタンをクリック、あるいは Shift + Enter キーでPythonソースコードが実行されます。

以下のソースコードを実行してみましょう。



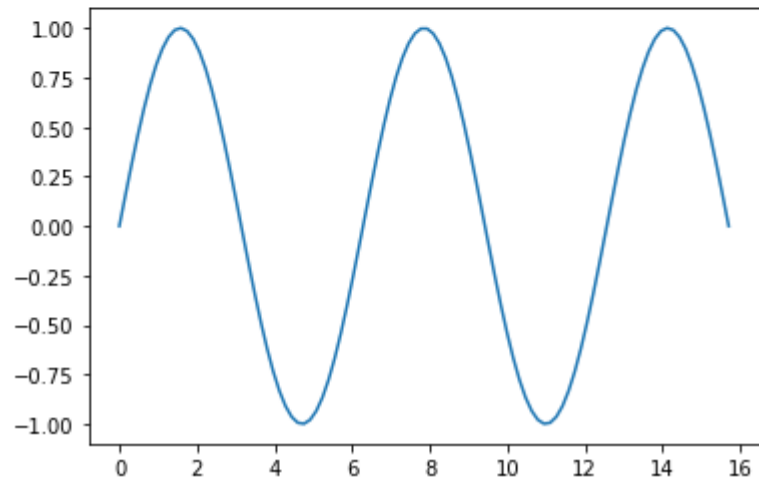
```
import numpy as np
import matplotlib.pyplot as plt
```

```
print('Plot sine wave')
```

```
timeline = np.linspace(0,5*np.pi,100)
wave = np.sin(timeline)
plt.figure()
plt.plot(timeline, wave)
plt.show()
```



Plot sine wave



# テキストの記述方法

テキストセルにはテキストを記述できます。

テキストは **Markdown** と呼ばれる書式で記述します。


01\_02\_Markdown.ipynb に基本的な記述方法を解説していますので、参照してください。

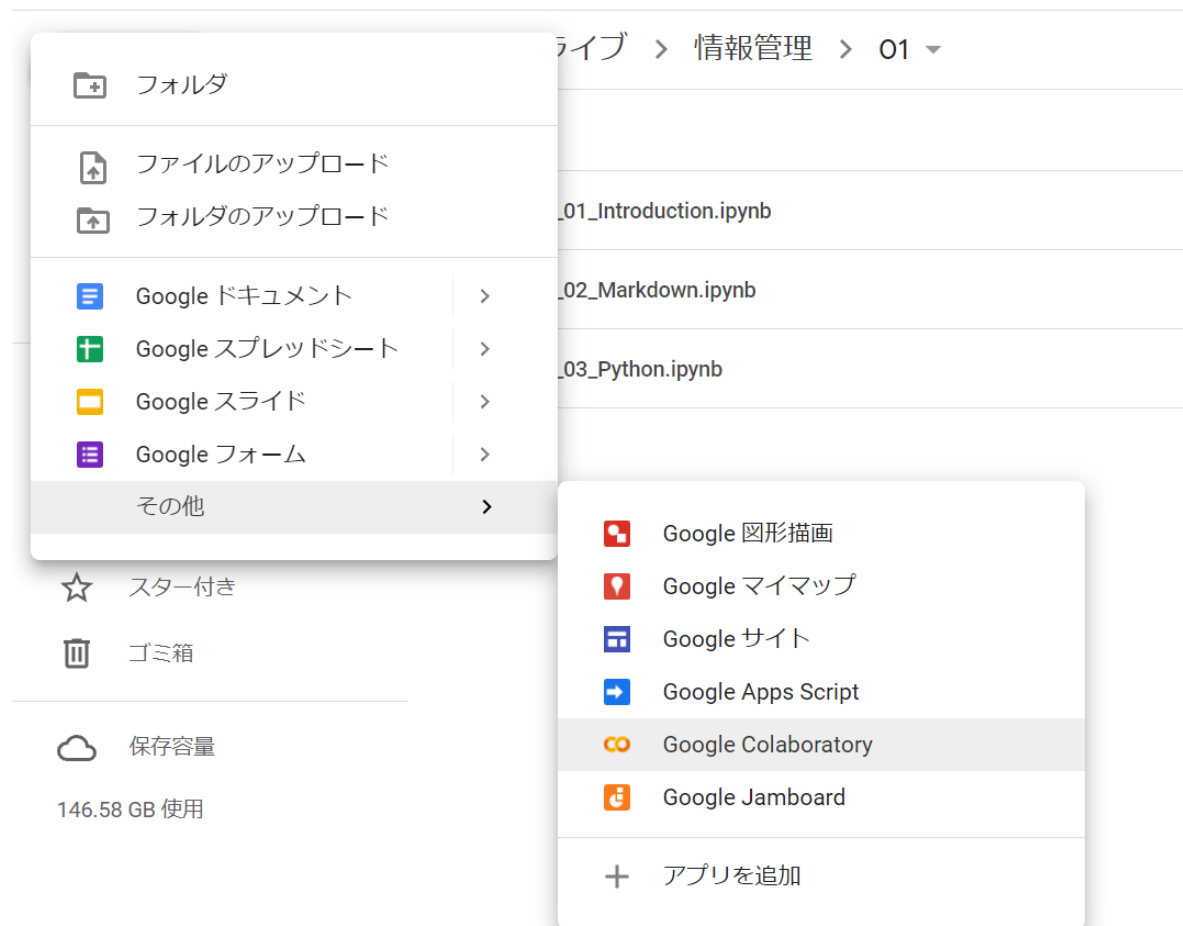
まずは改行の仕方を覚えましょう。

「改行 = スペース二つ」



# 新規Notebookファイルの作成法

Googleドライブの  新規 ボタンをクリックするか、右クリックし、「その他 → Google Colaboratory」を選択すると、新規ファイルが作成されて、Google Colabで開かれます。



# Pythonプログラムの基本

以下の内容は 01\_03\_Python.ipynb でも解説していますので、実際に動かしながら確認することをお勧めします。

# 算術演算

加算や乗算などの算術計算は、次のように行います。

```
>>> 2 + 3      加算: "+"  
5  
>>> 2 - 3      減算: "-"  
-1  
>>> 2 * 3      乗算: "*"  
6  
>>> 7 / 2      除算: "/"  
3.5  
>>> 7 // 2     整数の商: "//"  
3  
>>> 7 % 2      整数の余り: "%"  
1  
>>> 7 ** 2     累乗 (72): "**"  
49
```

# データ型

C言語と同様に、Pythonにもデータの型があります。

`type()` という関数を使うことで、データの型を調べることができます。

```
>>> type(10)
<class 'int'>      整数
>>> type(1.5)
<class 'float'>    小数
>>> type("hello")
<class 'str'>      文字列
```

# 変数

Pythonも変数を定義して、計算を行います。

C言語と異なり、変数を定義する際はデータ型を指定しません。

変数のデータ型は値や計算内容に応じて自動的に決定されます  
(このようなプログラミング言語は「動的型付き言語」と呼ばれます)。

```
>>> x = 10          変数 x を定義し、初期値に10を代入
>>> print(x)
10
>>> type(x)
<class 'int'>      変数 x は int型になっている。
>>> x = x * 1.5
>>> print(x)
15.0
>>> type(x)
<class 'float'>    xに小数の値をかけたことで、自動的にfloat型になった。
```

# リスト（配列）

C言語と同じような感覚で配列を定義できます。

```
>>> x = [1, 2, 3, 4, 5]    リストは"[]"を使って定義する。
>>> print(x)
[1, 2, 3, 4, 5]
>>> len(x)                リストの要素数は、len関数により得られる。
5
>>> print(x[0])           要素の番号は 0 から始まる。
1
>>> x[4] = 10             特定の要素に代入。
>>> print(x)
[1, 2, 3, 4, 10]
```

C言語の配列とは異なり、単一の要素だけでなく、複数の要素をまとめて（サブリスト）アクセスすることができます。これを**スライシング**と呼びます。

また、要素番号に“-”（マイナス）をつけることで、後ろから数えることもできます。

```
>>> print(x[0:2])         0番目から2番目(2番目は含まない!)まで取得。
[1, 2]
>>> print(x[1:])          1番目から最後まで取得。
[2, 3, 4, 5]
>>> print(x[:3])          最初から3番目(3番目は含まない!)まで取得。
[1, 2, 3]
>>> print(x[:-2])         最初から最後の要素のふたつ前まで取得。
[1, 2, 3]
```

# If 文

C言語ではif文の範囲を{} (カッコ)で囲っていましたが、Pythonではカッコを使わず、代わりに ":" (コロン)とインデントを使います。

(おそらくPythonを使い始めて最初に違和感を覚える点だと思います。)

```
a = 5
if a == 5:
    print("a=5")
elif a < 5:
    print("a<5")
else:
    print("a>5")
```

If 条件: (コロンを忘れずに)  
ifの中は、かならずインデント(空白)を入れる。  
elif は C言語の else if に相当する。  
else は C言語の else に相当する。

a=5

分岐の開始は: (コロン)、分岐の中はインデント(空白)を忘れないようにしましょう！

# for 文

for 文は以下のサンプルのように書きます。

range(N) は、「0～N-1までの整数」という意味です。

“for n in range(N)” は、「n が 0～N-1まで1ずつ増えながら、ループする」という意味です。  
つまり、C言語で言う “for( n=0; n<N-1; n++ )” に相当します。

```
for n in range(5):  
    print(n)
```

if と同様に、コロンとインデントを忘れずに！

```
0  
1  
2  
3  
4
```



# おわりに

以降の講義でプログラムを動かすための準備を行いました。

レポートは基本的には Notebook 形式で提出してもらいますので、色々試して使い慣れてください。

次回はPythonを使って実際にデータを触り、プロットなどをしてみます。

# レポート課題

- 新しく Notebook ファイルを作成してください。
- テキストセルに「(見出し1)レポート課題 1」、その下に「学籍番号」、さらにその下に「氏名」を記述してください。  
(見出し、改行ができていない場合は減点)
- for と if 文を使って、整数 0 ～ 9 までの、奇数の総和と、偶数の総和を計算して表示するプログラムを、コードセルに記述し、実行してください。
- 作成した Notebook (.ipynb) ファイルを**HTMLファイルに変換して** BEEFで提出してください。  
(その際、感想や質問がある場合は書いてください。)

レポート提出期限：4/19(火) AM10:30

# レポート課題 提出例

レポートの提出例を以下に示します。

赤字はチェック項目です。満たしていない場合は減点対象です。



## ▼ レポート課題 1 ← 見出し 1 の書式を使うこと



学籍番号 : 0123456789

氏名 : 山田太郎

改行がされていること（改行の方法は何でも良い）



✓  
0  
秒



```
sum_odd = 0
sum_even = 0
for
  if
  el

print('sum_even = ' + str(sum_even))
print('sum_odd = ' + str(sum_odd))
```

sum\_even = 20  
sum\_odd = 25

正しい実行結果が表示されていること

# レポート課題の HTML への変換方法

BEEFにアップしてあるファイル一式内の「convert\_report.ipynb」を使用してください。  
指示通りに動かせば、Googleドライブ上に、HTML ファイルが作成されます。

 convert\_report.ipynb ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ [すべての変更を保存しました](#)

+ コード + テキスト

≡

🔍

{x}

📁

▼ レポート課題提出用のプログラム

これはレポート課題で作成したNotebookファイル (.ipynb) を、提出するためのhtmlファイルに変換するプログラムです。

使用するためには、以下の情報を入力してから、上から順番に実行してください。

- work\_dir = 'My Drive/課題が保存されているフォルダ'
- file\_name = 'ファイル名.ipynb'

✓ 0 秒 [1] # 以下の2変数を設定してください。  
work\_dir = 'My Drive/情報管理/01'  
file\_name = 'report\_01.ipynb'

Google ドライブをマウント

(このとき、Google colabからGoogleドライブに接続するための確認などが出てきますが、指示に従ってGoogle ドライブに接続してください。)

✓ 1 分  from google.colab import drive  
drive.mount('/content/drive')

Mounted at /content/drive

ここを実行したタイミングで、Google ドライブからセキュリティ確認の通知が出ます。「Googleドライブに接続」→ アカウントの指定 → 「許可」で接続できます。(このソースコードの意味は次回説明します。)

# レポート提出時の注意点

（初回のレポートを除いて）全てのレポートは、課題に対する考え方や結果に対する考察を含んでいます。

単なるプログラムのレポートではありません。従って、以下のような文章としてまとまっていないレポートは未完成として扱います。

- 途中で切れているレポート  
（途中までしか出来なかったとしても、「～～ができませんでした」等の説明をして、文章として完結させてください。）
- ソースコードだけが羅列されたレポート  
（実行結果に対する説明が一切ない。）
- 課題内容と全く関係の無い文章やソースコードが入っている。  
（例：授業のサンプルプログラムを流用し、無関係なサンプルコードや説明文などが入ったまま提出している。）