

# はじめに

- 本日は「Python入門」
  - 演習講義で用いるプログラミング言語がPythonのため
- ある程度できる, という方は無理してでなくてOK
- 自由に入退出OK (単位が不要な方)
- 単位が必要な方へ
  - 「今日の講義」は出席カウントしないので無理して出なくてOK
  - 次回以降は遅刻厳禁 (Moodle 管理) & やむを得ない退出時は連絡
- わからないことがあれば気軽にどんどん聞いて下さい
  - 教員もTAもたくさんいます!
  - 私が喋っている途中にTAを捕まえて質問してもOK

# 次回以降

- 9/1 : 距離, クラスタリング, 統計的検定, 相関分析, 回帰
- 9/2 : 主成分分析, 画像処理基礎
- 9/3 : 機械学習

# 「DSことはじめ」

九州大学 大学院システム情報科学研究所  
情報知能工学部門  
末廣大貴, Diego Thomas, 正井克俊

# データサイエンスに必要なスキル

データ解析スキル  
検定、クラスタリング、回帰、  
主成分分析、・・・



データを集めた後、実際にこれらの分析を行うには？



「プログラミング」が必要  
解析のための具体的なデータ加工処理を  
コンピュータに指示

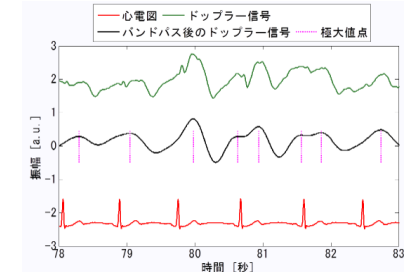
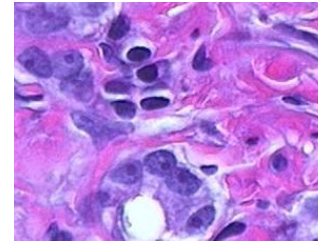
# データ解析

## 数値・カテゴリデータ 解析



数値データ  
身長・体重  
血圧  
カテゴリデータ  
アンケート  
男女  
喫煙の有無

## メディアデータ解析 画像 シグナル



細胞の大きさ、心拍間隔等  
画像データからは直接抽出  
できない場合が多く、事前  
処理が必要



本講義：

まず、数値・カテゴリデータを対象とした解析手法について学び、  
画像を例にメディアデータを対象とした解析手法を学ぶ

# Excel じゃダメ？

- 「まずは Excel で見てみる」は大事
  - データの集計，簡単な可視化
  - Excel で完結するのであればそれでヨシ
- もっと高度な分析がしたい！
  - 統計分析，機械学習
- エクセルじゃ開けない！
- 繰り返し処理などが必要！
- エクセルでは扱えないデータもたくさんある
  - 画像，言語，グラフ構造，etc.

# 本日の講義内容

- データサイエンス技術のプログラム実践前の準備
- プログラムの基本的な内容を勉強する
  - Google Colabの準備
  - コンピュータに命令する：コンソール上のコマンド
  - コンピュータにまとめて命令：プログラムコード
  - 演習問題

# プログラムの準備

## Python

# プログラムとは

- コンピュータに対する命令の記述
- 命令は何語？
  - C
  - Java
  - R
  - Python

# Python の利用

Google Colaboratoryを利用  
以下の web site へ

<https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja#scrollTo=-Rh3-Vt9Nev9>

# Google Colaboratory

← → ↻ https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja#scrollTo=xitplqMNk\_Hc ☆ 🌐 📄 🗄️ 🔍

Colaboratory へようこそ 🗄️  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

+ コード + テキスト | 📁 ドライブにコピー 接続 ▼ 編集

目次 コードスニペット ファイル ✕

Colaboratory の概要

はじめに

その他のリソース

機械学習の例: Seedbank

📁 セクション

Colaboratory へようこそ

Colaboratory は、完全にクラウドで実行される Jupyter ノートブック環境です。設定不要で、無料でご利用になれます。  
Colaboratory を使用すると、コードの記述と実行、解析の保存や共有、強力なコンピューティングリソースへのアクセスなどをブラウザからすべて無料で行えます。

🎥 Colaboratory の概要  
この 3 分間の動画では、Colaboratory の主な機能の概要をご紹介します。

📺 Get started with Google Colaboratory (Coding T... 後で見る 共有

Intro to Google Colab

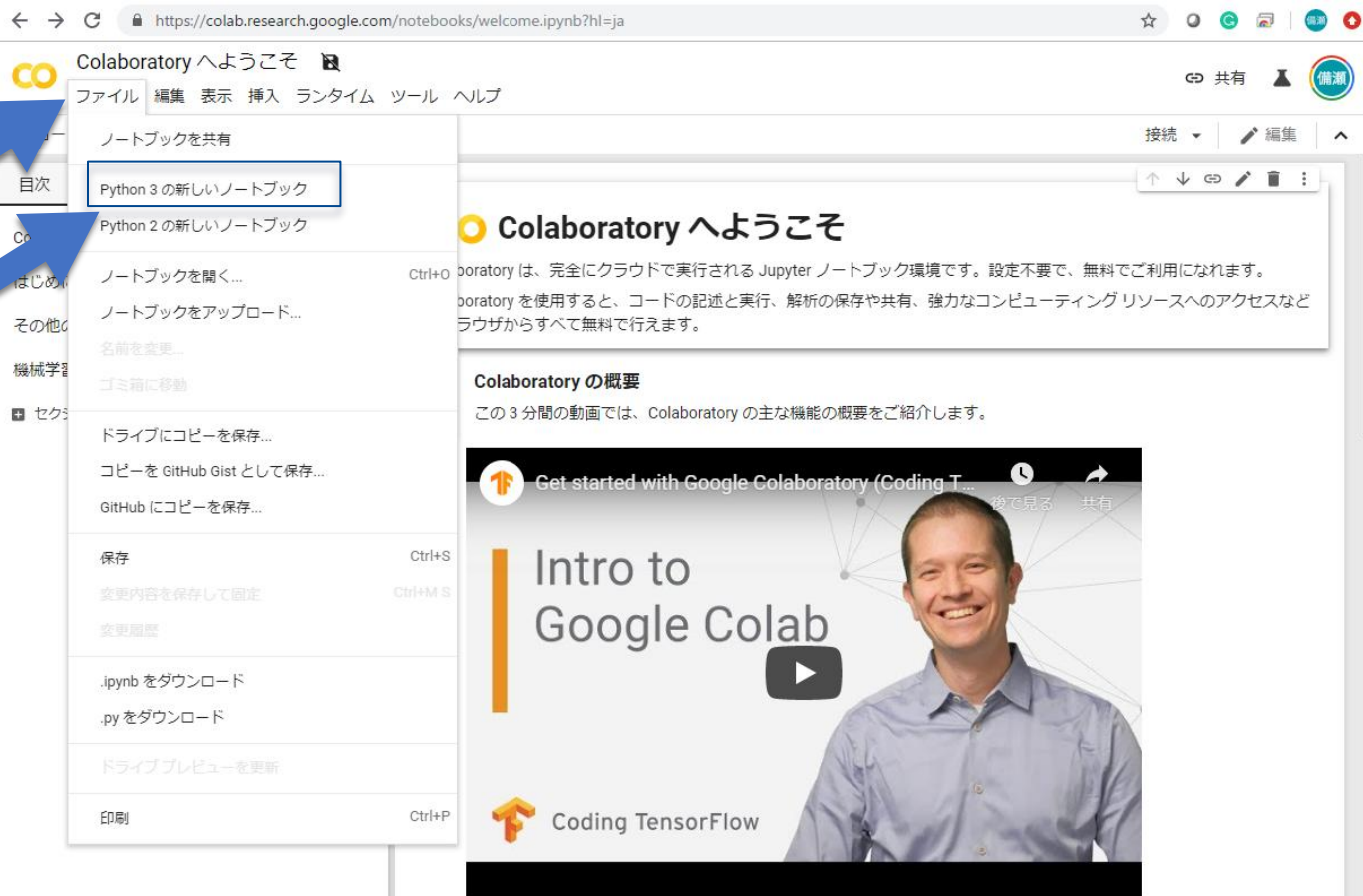
Coding TensorFlow

自分のGoogle  
アカウントで  
ログイン

ファイルを  
セーブできる  
ようになる

# Jupyter notebookを使う

## Jupyter notebook: Pythonの開発環境の一つ

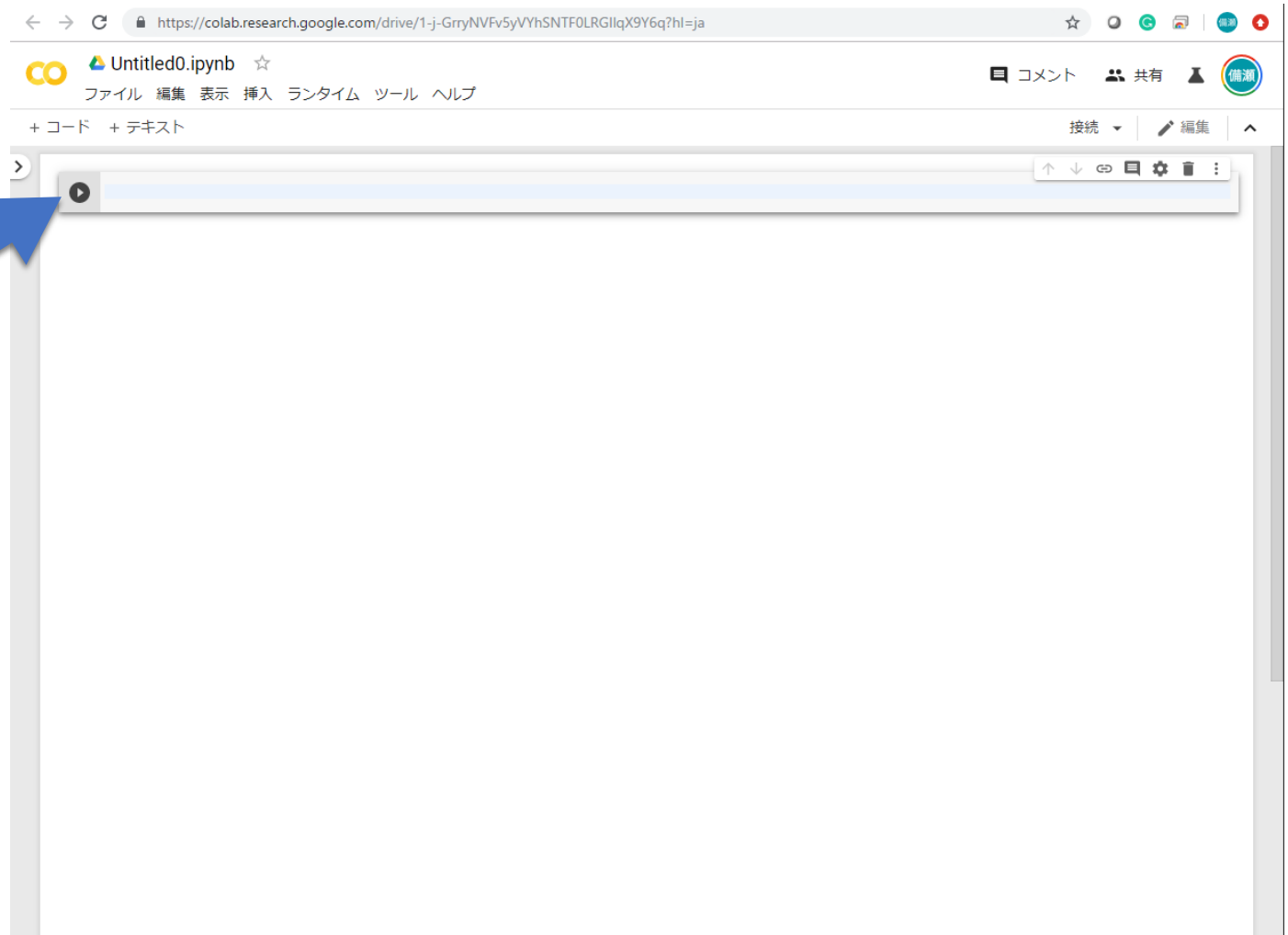


ファイルを  
クリック

Python3の新しい  
ノートブック  
をクリック

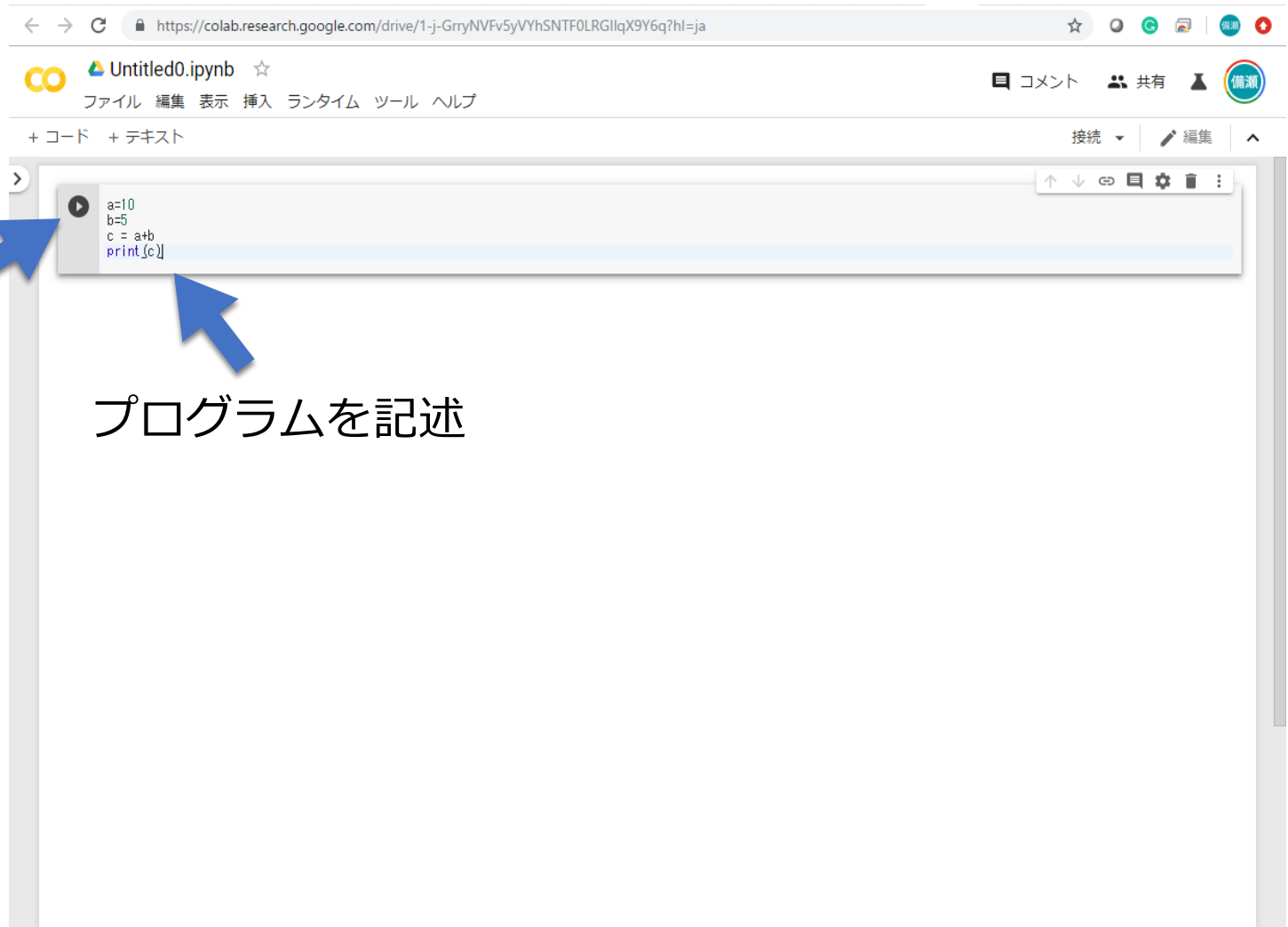
※) 数字はPython  
のバージョン

# Cell



Cell  
プログラムの  
入力フィールド  
Cellの単位で  
実行される

# Cellへの入力



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: <https://colab.research.google.com/drive/1-j-GrryNVFv5yVYhSNTF0LRGIlqX9Y6q?hl=ja>. The notebook title is "Untitled0.ipynb". The menu bar includes "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", and "ヘルプ". The toolbar shows "接続", "編集", and other icons. A code cell is selected, containing the following Python code:

```
a=10  
b=5  
c = a+b  
print(c]
```

Two blue arrows point to the code cell. The first arrow points to the play button icon on the left side of the cell, with the text "▶ボタンをクリックで実行" (Click the ▶ button to execute). The second arrow points to the code text inside the cell, with the text "プログラムを記述" (Describe the program).

▶ボタンをクリックで  
実行

プログラムを記述

# Cellの実行



The screenshot shows a web browser window with the URL <https://colab.research.google.com/drive/1-j-GrryNVFv5yVYhSNTF0LRGIq>. The notebook title is "Untitled0.ipynb". The menu bar includes "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", and "ヘルプ". Below the menu, there are buttons for "+ コード" and "+ テキスト". A code cell is shown with the following code:

```
a=10  
b=5  
c = a+b  
print(c)
```

Below the code, the execution result "15" is displayed. A blue arrow points from the Japanese text "実行結果が表示される" to the output "15".

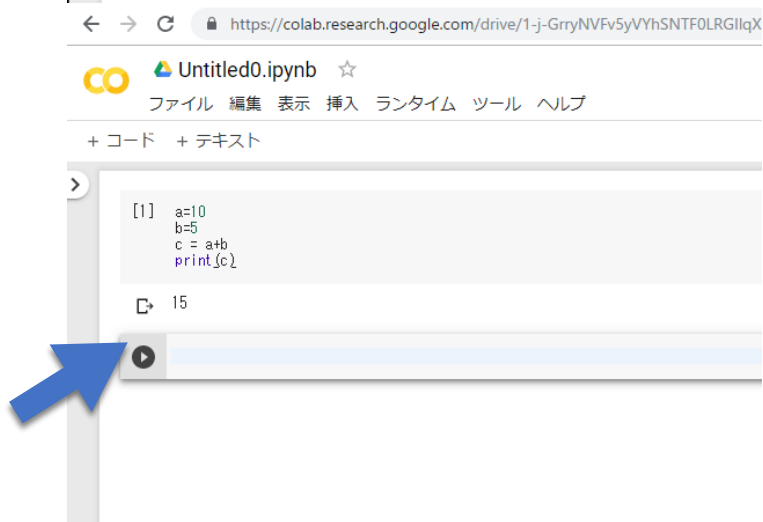
実行結果が  
表示される

# Cellの追加

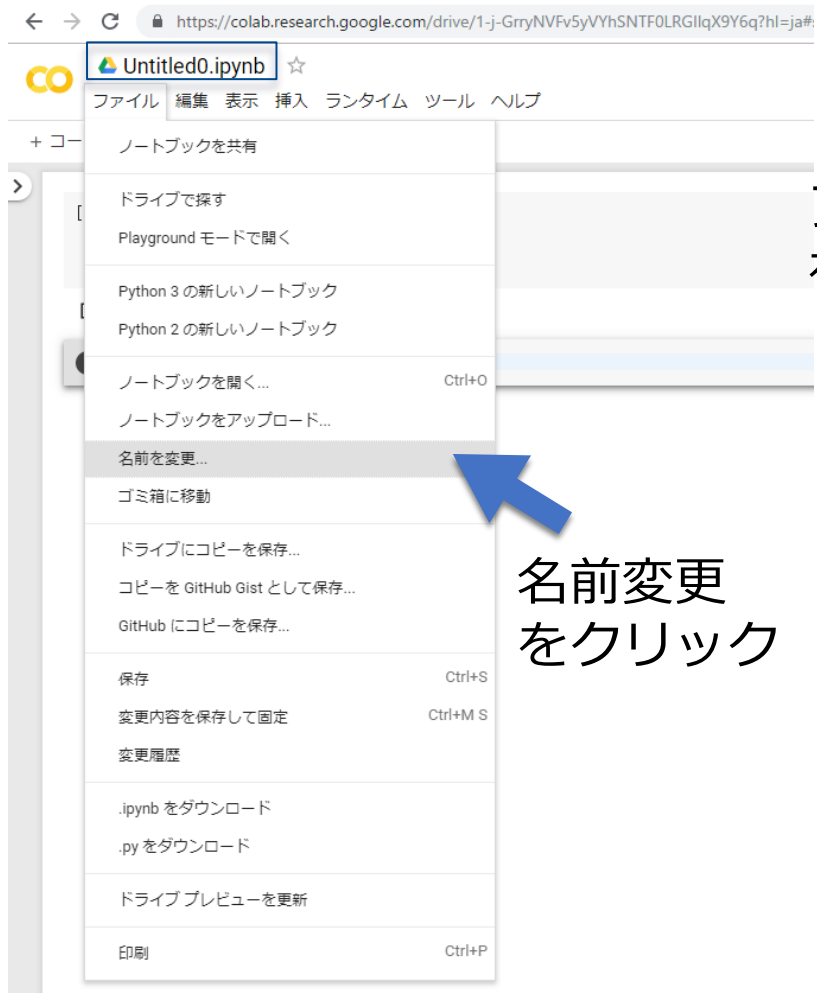
+コードを  
クリック



新たなCellが  
追加される

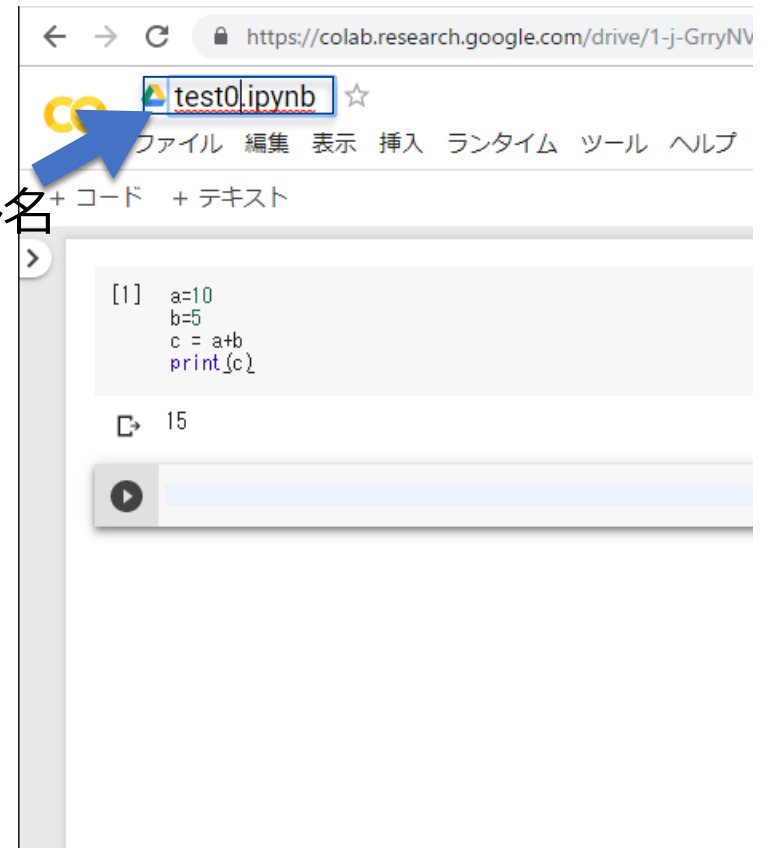


# ファイル名を変更

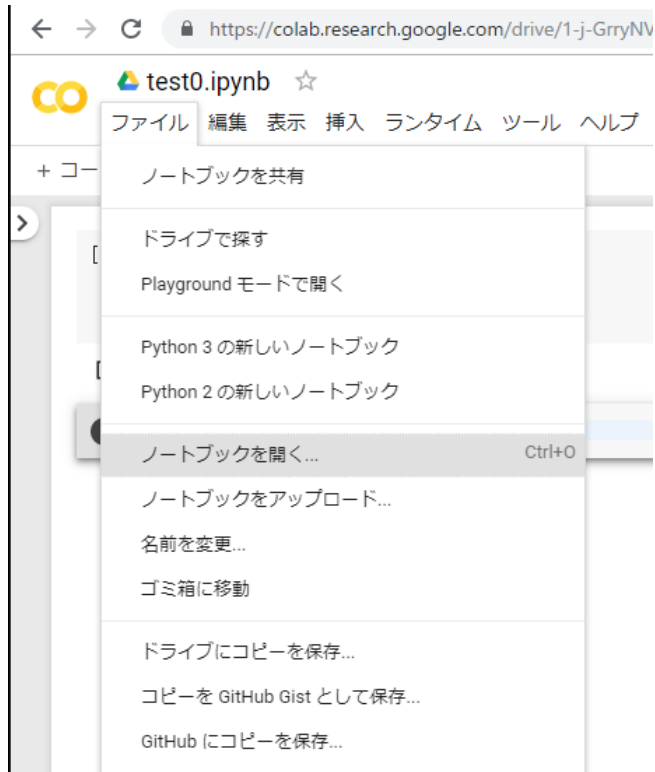


ファイル名  
を編集

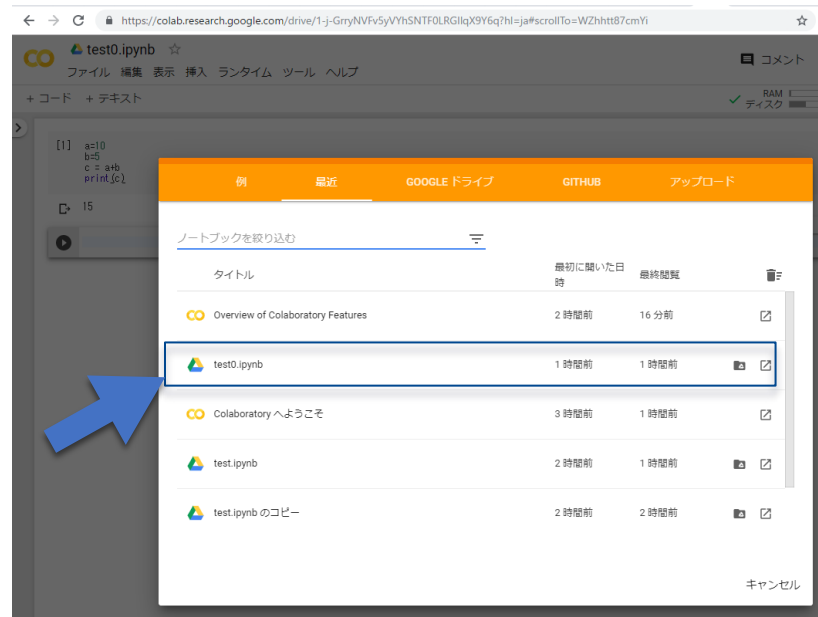
名前変更  
をクリック



# ファイルを開く



さっきの  
ファイルが  
選択できる



# ファイルの保存先 : Google Drive

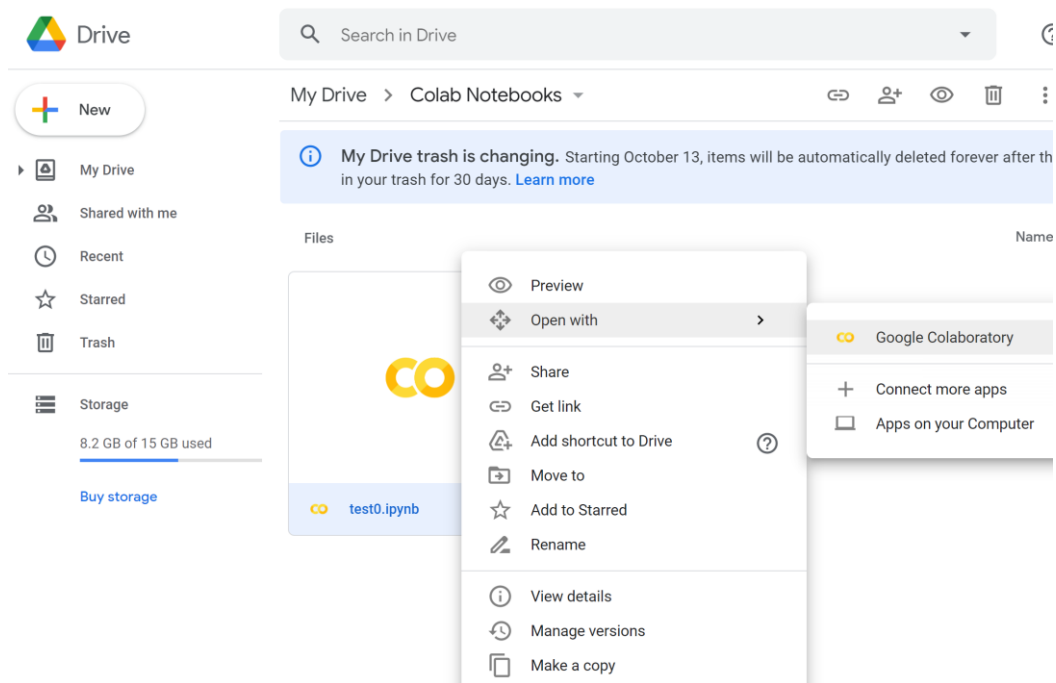
The screenshot shows the Google Drive interface. On the left, there is a navigation menu with options like 'New', 'My Drive', 'Shared with me', 'Recent', 'Starred', 'Trash', and 'Storage'. The 'My Drive' folder is selected. In the main area, there is a search bar, a notification about the trash, and a 'Quick Access' section. A folder named 'Colab Notebooks' is visible in the 'Folders' section and is highlighted with a red rectangular box.

自動的に同じアカウントの  
Google Drive の下に  
フォルダが作成されている

The screenshot shows the Google Drive interface with the 'Colab Notebooks' folder selected. The folder contains a single file named 'test0.ipynb'. The file icon is a yellow circle with 'CO' inside. The storage usage is shown as 8.2 GB of 15 GB used.

先ほどのファイルが保存されている

# ファイルの保存先 : Google Drive



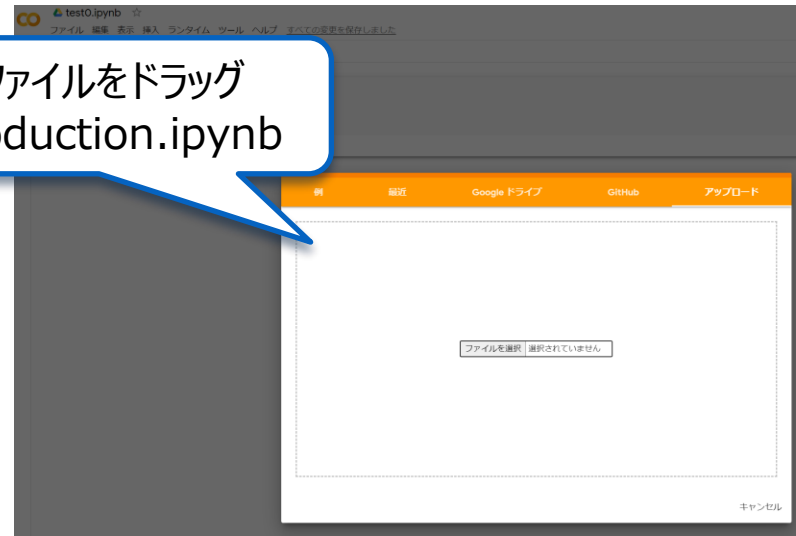
Google Driveから  
ファイルを開ける

ファイルの上で右クリックして、  
Open with -> Google Colaboratory をクリック

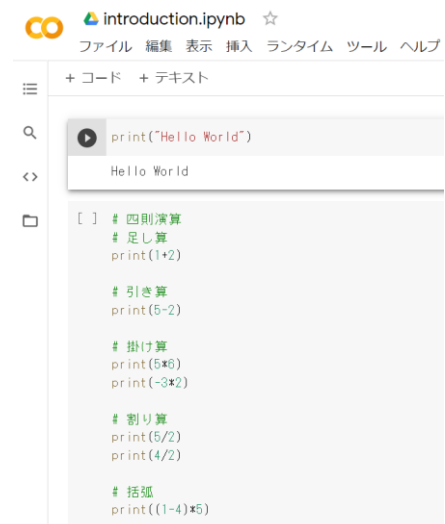
# Fileのアップロード



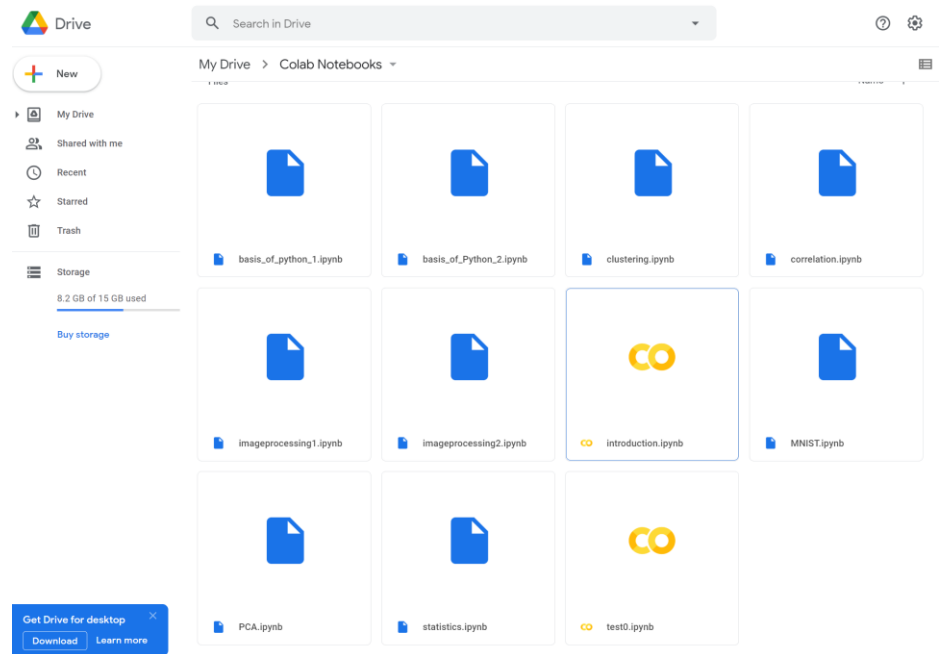
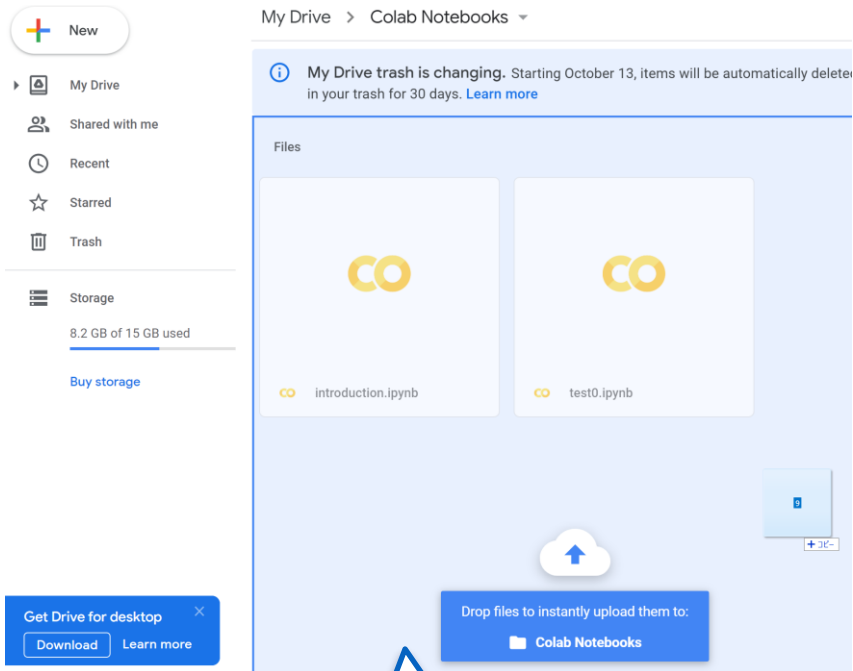
PCからファイルをドラッグ  
例: introduction.ipynb



Introduction.ipynbが開く



# DriveへのFileアップロード



複数ファイルまとめてDrop

# 新しいフォルダの作成

basis\_of\_python\_1.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

ファイル

- bin
- boot
- content
  - sample\_data
- datalab
- dev
- etc
- home
- lib
- lib32
- lib64
- media
- mnt
- opt
- proc
- root

+ コード + テキスト

```
[1] # list
a = [0, 1, 2]
print(a)

b = [2, 0.5, 4, 2, 1]

print(c)

e = [{"A", 52, 1.7}, {"B", 75, 1.6}]
print(e[1][1])
```

アップロード  
フォルダの削除  
ファイル名の変更  
新しいファイル  
新しいフォルダ  
パスをコピー  
更新

Contentの下に新しい  
フォルダ'data'を作成

フォルダができた

basis\_of\_python\_1.ipynb ☆  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

ファイル

- bin
- boot
- content
  - data
  - sample\_data
- datalab
- dev
- etc
- home
- lib
- lib32
- lib64
- media
- mnt

+ コード + テキスト

```
# list
a = [0,
print(

b = [2,
print(

b = [2,
print(
print(

list=[
list.ap
list.ap
print(

c = [{"
print(c
```

# データのアップロード

The image shows a JupyterLab interface on the left and a Windows Explorer window on the right. The JupyterLab interface displays a file browser on the left with the 'data' folder selected. A context menu is open over the 'data' folder, with 'アップロード' (Upload) highlighted. The main area shows a code cell with the following Python code:

```
# list
a = [0, 1, 2]
print( a )

b = [2, 0.5, 4, 2, 1]
print( b )
```

The Windows Explorer window shows a list of files in the 'Data' folder. The files are:

名前	更新日時	種類	サイズ
ais.csv	2019/08/16 8:35	Microsoft Excel CS...	15 KB
boston_house.csv	2019/08/16 8:35	Microsoft Excel CS...	62 KB
data.csv	2019/08/16 8:35	Microsoft Excel CS...	12 KB
grade_data1.csv	2019/08/16 8:35	Microsoft Excel CS...	1 KB
grade_data2.csv	2019/08/16 8:35	Microsoft Excel CS...	1 KB
grade_data3.csv	2019/08/16 8:35	Microsoft Excel CS...	1 KB
height_weight.csv	2019/08/16 8:35	Microsoft Excel CS...	4 KB
height_weight_small.csv	2019/08/16 8:35	Microsoft Excel CS...	1 KB
height_weight_small2.csv	2019/08/16 8:35	Microsoft Excel CS...	1 KB
height_weight_small3.csv	2019/08/16 8:35	Microsoft Excel CS...	1 KB
iris.csv	2019/08/16 8:35	Microsoft Excel CS...	5 KB
iris_2metrics1.csv	2019/08/16 8:35	Microsoft Excel CS...	4 KB
iris_2metrics2.csv	2019/08/16 8:35	Microsoft Excel CS...	4 KB
laborfor.csv	2019/08/16 8:35	Microsoft Excel CS...	1 KB
predicti.csv	2019/08/16 8:35	Microsoft Excel CS...	2 KB
some.csv	2019/08/16 8:35	Microsoft Excel CS...	1 KB
temperature_data.csv	2019/08/16 8:35	Microsoft Excel CS...	10 KB

The Explorer window also shows a file name field containing: "grade\_data3.csv" "height\_weight.csv" "height\_weight\_small.csv" "height\_weight\_small2.csv" "height\_weight\_small3.csv".

Data フォルダを選択してアップロードをクリック

配布資料のDataフォルダのパスを指定して、全てのファイルを選択

これでプログラムからファイルを読めるようになった

※) ただし接続が切れると初期化されるので、毎日アップする必要がある 24

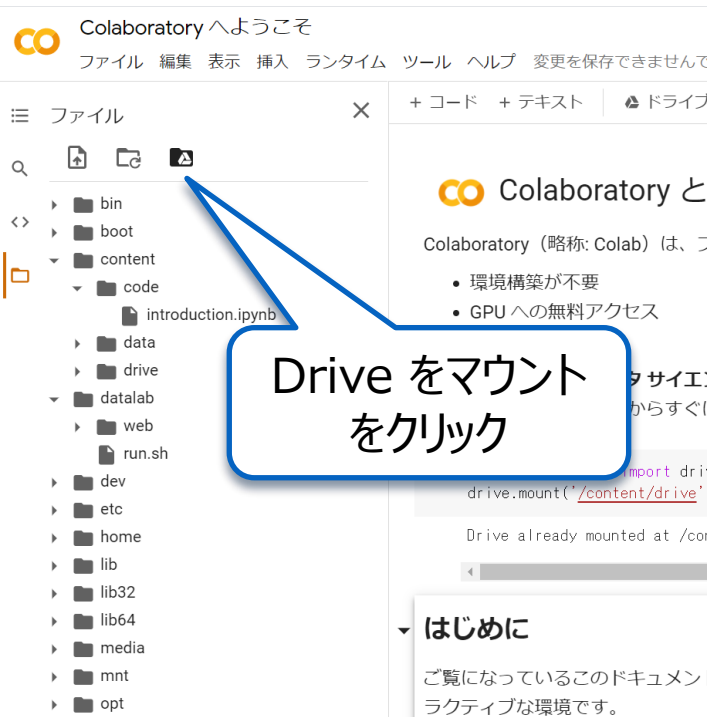
# Google Driveとの連携

# 準備 : Google Driveへフォルダごとアップ

The screenshot shows the Google Drive interface. On the left is a navigation sidebar with options: 'New', 'My Drive', 'Shared with me', 'Recent', 'Starred', 'Trash', and 'Storage' (8.2 GB of 15 GB used). The main area is titled 'My Drive' and contains a notification: 'My Drive trash is changing. Starting October 13, items will be automatically deleted forever after they've been in the trash for 30 days.' Below this is a 'Quick Access' section with three items: 'basis\_of\_python\_1.ipynb' (edited today), 'introduction.ipynb' (edited today), and 'imageprocessing1.ipynb' (uploaded today). At the bottom is a 'Folders' section with three folders: 'Colab Notebooks', 'Google Buzz', and 'ImgData'.

フォルダごとドラッグすれば、  
フォルダの中身も全てアップロード可能

# Google Driveとの接続



Colaboratory へようこそ  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 変更を保存できません

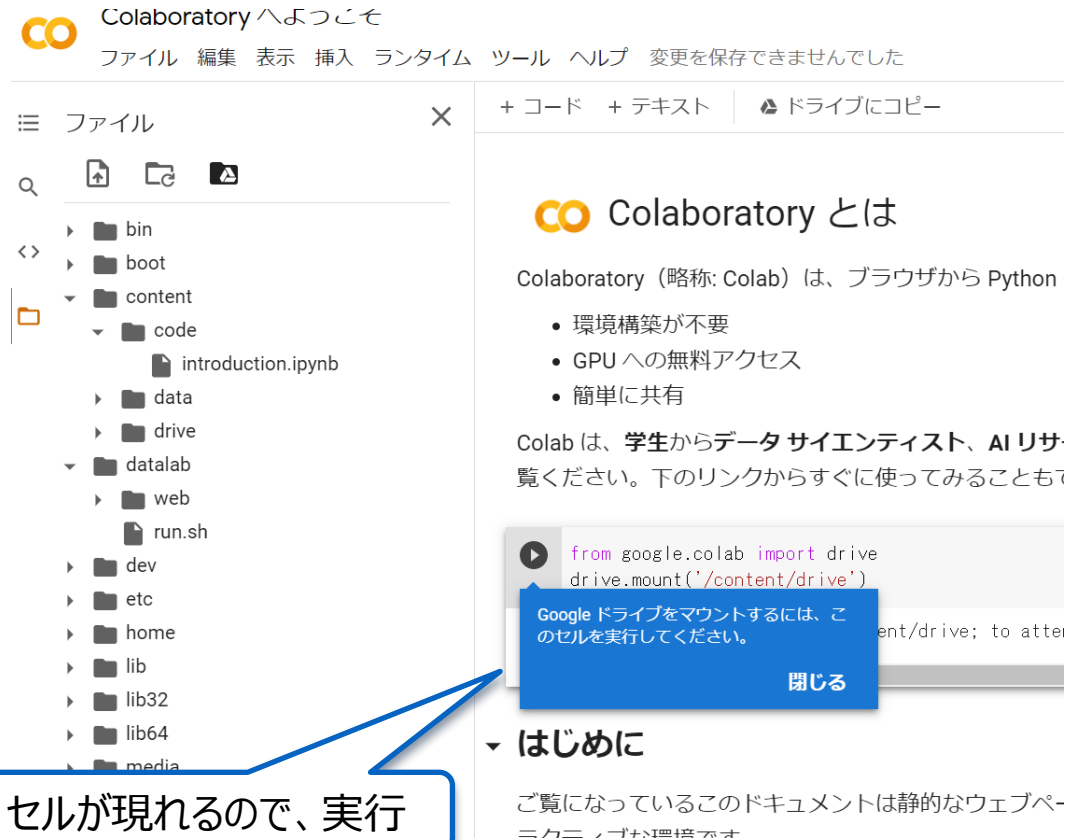
ドライブ

bin  
boot  
content  
code  
introduction.ipynb  
data  
drive  
datalab  
web  
run.sh  
dev  
etc  
home  
lib  
lib32  
lib64  
media  
mnt  
opt

Colaboratory とは  
Colaboratory (略称: Colab) は、  
• 環境構築が不要  
• GPU への無料アクセス

はじめに  
ご覧になっているこのドキュメントは静的なウェブページです。

Drive をマウント  
をクリック



Colaboratory へようこそ  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 変更を保存できませんでした

ドライブにコピー

bin  
boot  
content  
code  
introduction.ipynb  
data  
drive  
datalab  
web  
run.sh  
dev  
etc  
home  
lib  
lib32  
lib64  
media

Colaboratory とは  
Colaboratory (略称: Colab) は、ブラウザから Python  
• 環境構築が不要  
• GPU への無料アクセス  
• 簡単に共有

Colab は、学生からデータサイエンティスト、AIリサー  
ご覧ください。下のリンクからすぐに使ってみることも

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

Google ドライブをマウントするには、このセルを実行してください。

閉じる

はじめに  
ご覧になっているこのドキュメントは静的なウェブページです。

セルが現れるので、実行

# Google Driveとの接続

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

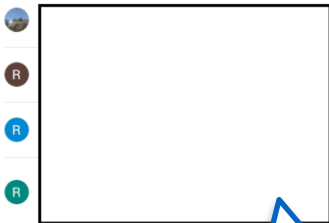
... Go to this URL in a browser: <https://accounts.google.com/o/oauth2/auth?c>

Enter your authorization code:

リンクをクリック

Google にログイン

アカウントの選択  
「Google Drive File Stream」に移動



Google Driveの  
アカウントを選択

Google にログイン

Google Drive File Stream が  
Google アカウントへのアクセス  
をリクエストしています

ryomab@gmail.com

Google Drive File Stream に以下を許可します:

- Google ドライブのすべてのファイルの表示、編集、作成、削除
- Google フォトの写真、動画、アルバムの表示
- プロフィールや連絡先などの Google ユーザー情報の表示
- Google ドライブのドキュメントの表示、編集、作成、削除

Google Drive File Stream を信頼できることを確認

機密情報をこのサイトやアプリと共有する場合があります。 Google Drive File Stream シーボリティーで、ユーザーの扱われるかをご確認ください。 詳細は、Google アカウントで

リスクの詳細

キャンセル

許可

許可

Google

ログイン

このコードをコピーし、アプリケーションに切り替えて貼り付けてください。

4/1AfDhmr i5BSRz jmYtv0eCPZGtk7xrktlcvoqllleCsrJ0 S3FEF40oeJs4gqg0

コピー

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

... Go to this URL in a browser: <https://accoun>

Enter your authorization code:

9CsrJ0S3FEF40oeJs4gqg0

ペースト

<>

```
> bin  
> boot  
v content  
  > drive  
  > sample_data  
> datalab  
> dev  
> etc
```

driveが  
追加された

コンピュータに指示を出す

Python で  
あそんでみよう

# 大事な「おまじない」

- 今後, jupyter を起動したらまずはじめに必ずやること！
- `import numpy as np` と打って enter キーを押す

```
In [2]: import numpy as np
```

```
In [3]:
```

コンピュータに指示を出す

文字の表示

お約束のHello World !

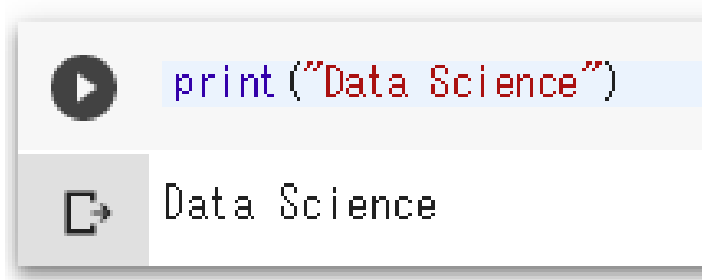
# Hello World

- Cellで  
print("Hello World") と入力
-  を押す（実行）



# 練習 1 : 文字を変えてみよう !

- コンソールのところで
- 好きな言葉を“ ”に入れて表示させよう  
print(“ ”) と入力
- return キーを押す (実行)



```
▶ print("Data Science")  
↳ Data Science
```

The image shows a code editor window with two lines. The first line is a Python print statement: `print("Data Science")`. The second line is the output of the code: `Data Science`. The code is highlighted in blue, and the output is shown in a separate box below the code.

コンピュータに指示を出す

四則演算

たす, ひく, かける, わる

# 足し算

- +

- 例

```
In [1]: 1+2  
Out[1]: 3
```

# 引き算

- -

- 例

```
In [2]: 5-2
```

```
Out[2]: 3
```

# 掛け算

- \* (アスタリスク) を使う

- 例

```
In [120]: 5*6  
Out[120]: 30
```

```
In [121]: -3*2  
Out[121]: -6
```

# 割り算

- / (スラッシュ) を使う

- 例

```
In [3]: 5/2  
Out[3]: 2.5
```

```
In [4]: 4/2  
Out[4]: 2
```

# 括弧

- $()$ 内を優先的に計算
- $(1-4)*5$

# 四則演算

- 足し算 :  $+$ 
  - $1+2$
- 引き算 :  $-$ 
  - $5-2$
- 掛け算 :  $*$  (アスタリスク)
  - $5*6$
- 割り算 :  $/$  (スラッシュ)
  - $5/2$
- 括弧 :  $()$ 内を優先的に計算
  - $(1-4)*5$

## 練習 2 : 四則演算

- 小学生の問題)

- A君が八百屋で100円のリンゴを5個と400円のイチゴを買いました。
- 合計いくらになるかプログラムで計算させてみよう！

- BMI をプログラムで計算してみよう

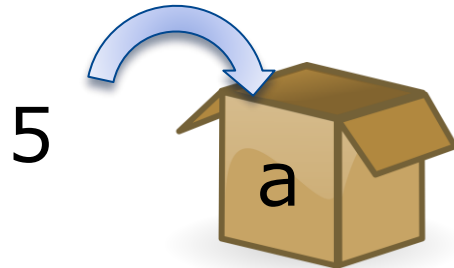
ヒント : BMI は  $\text{体重} \div (\text{身長} \times \text{身長})$

※体重は kg, 身長は m

# 変数

# 変数 (1)

- 変数：好きな数を保存する「箱」



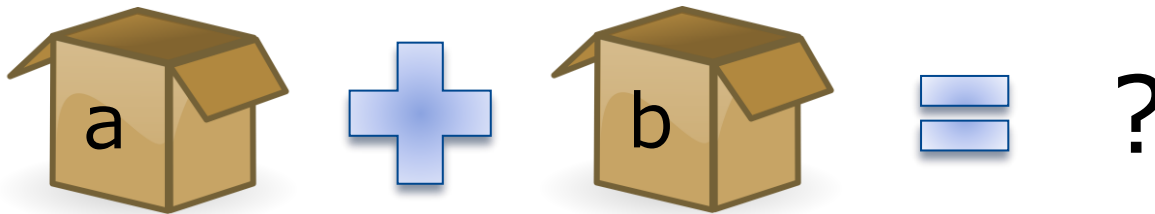
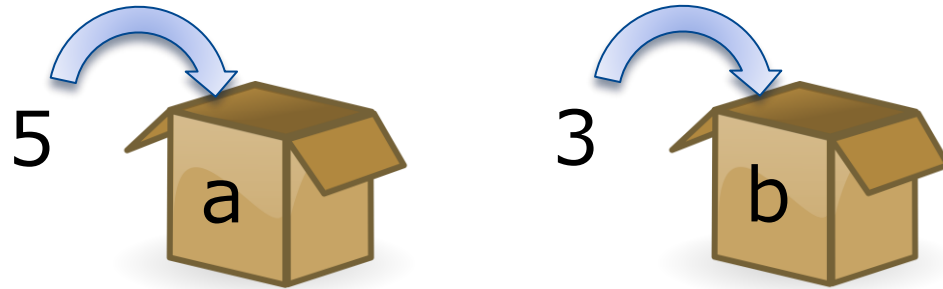
- =を使って保存  
(左側が「箱の名前」、右側が好きな数)
- $a=5$
- $b="Monday"$  ← 実は文字列でも OK !

# 変数：演算 (2)

- 変数同士の演算

- $a=5$

- $b=3$



```
In [293]: a + b
```

```
Out[293]: 8
```

```
In [294]: a - b
```

```
Out[294]: 2
```

# 変数 : 代入(2)

- 変数同士の演算
- $a=5, b=3$



$a = b$  とすると、  
a の中身を捨てて、  
b の中身を a に入れる

```
In [312]: a = b
```

```
In [313]: a  
Out[313]: 3
```

# 変数 (3)

- 変数の中身を変える

- $a=5$

 $a=20$ 

- $b=3$

 $b=10$ 

直接数字を書いて演算すると、毎回書き換える必要がある

```
In [301]: 5 + 3  
Out[301]: 8
```

```
In [302]: 20 + 10  
Out[302]: 30
```

変数で演算すると、同じ式でO.K.

```
In [293]: a + b  
Out[293]: 8
```

```
In [299]: a + b  
Out[299]: 30
```

# 変数 (4)

- データ 1

$$a=5, b=4, c=1, d=6, e=29, f=12$$

$$(5+4)*(1+6)/29 + 12*5*4*(6-1)$$

- データ 2

$$a=8, b=3, c=2, d=7$$

$$(8+3)*(2+7)/20 + 1$$

毎回すべて書き換える？

プログラムの大きなメリット！  
同じ命令文が違うデータに  
使える

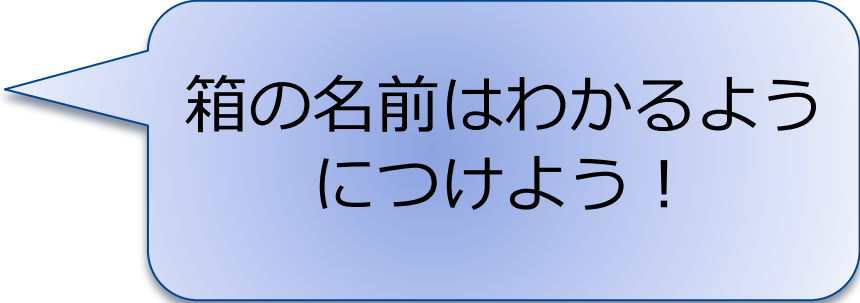
式で記述すれば、中身を変えるだけでOK

- $(a+b)*(c+d)/e + f*a*b*(d-c)$

# 変数 (5) : 名前の付け方

- 例 :

```
name="John"  
age=30  
weight=56  
height=1.68  
BMI=weight/(height*height)  
print( name )  
print( age )  
print( weight )  
print( height )  
print( BMI )
```



箱の名前はわかるように  
つけよう！

# 変数 (6)

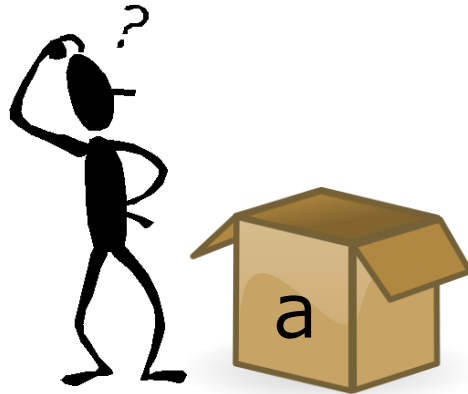
- 良くない例 :

a="John"

b=30

c=56

d=1.68



- えっと BMI の計算式は . . .  $e=c/(d*d)$

アドバイス : 変数名はなんでもいいけど,  
できる限り意味がわかりやすいように用意する

## 練習 3 : 変数

- 変数を使って、BMI をプログラムで計算してみよう

ヒント : BMI は  $\text{体重} \div (\text{身長} \times \text{身長})$

weight = 自分の体重

height = 自分の身長

※体重は kg, 身長は m

コンピュータに指示を出す

関数

# 「材料」と「道具」(1)

材料 (変数)



道具 (関数)

切る



皮をむく



煮る



print や  
+や-も  
道具の一種



# 関数(1)

例)

- 平方根 ( $\sqrt{\quad}$ ) を計算する関数 (道具)
  - sqrt (平方根は、英語で'square root'でその略)
- 道具を使うためのおまじない
  - import math

```
In [319]: math.sqrt(4)
```

```
Out[319]: 2.0
```

```
In [320]: math.sqrt(9)
```

```
Out[320]: 3.0
```

ライブラリ  
道具はまとめて整理されている  
調理する道具は台所、  
トンカチ等の工具は工具箱  
mathにsqrtがある  
仕舞っている処から出して、  
道具を使える状態にしている

## 関数(2)

- 他にもいろいろなライブラリ、関数がある
- `import numpy`
- `numpy.round(3.8)`
  - 小数点以下を四捨五入

```
In [328]: numpy.round(3.8)
Out[328]: 4.0
```

- 毎回ライブラリ名をフルネームで書くのは面倒 : 短縮してしまおう
- `import numpy as np`

```
In [330]: np.round(3.8)
Out[330]: 4.0
```

# 練習 4 : 関数の利用

- 関数を使って計算しよう！
  - 81の平方根を計算しよう
  - 256の平方根を計算しよう
  - 9.4を四捨五入しよう
  - 9.7を四捨五入しよう
- 自分で関数をWebで調べて使ってみよう！
  - 絶対値を求める関数を調べてみよう
  - -5の絶対値を関数を使って計算しよう

ヒント) Python 絶対値  
でGoogleで検索してみよう

# 基本注意事項

※知ってる人は遠慮なく飛ばしてください

# 全角文字，日本語入力を避ける

- 日本語入力するとエラーが出る

a \_ + \_ b  
全角スペース

```
In [314]: a + b
File "<ipython-input-314-c469e43fc799>", line 1
  a + b
    ^
SyntaxError: invalid syntax
```

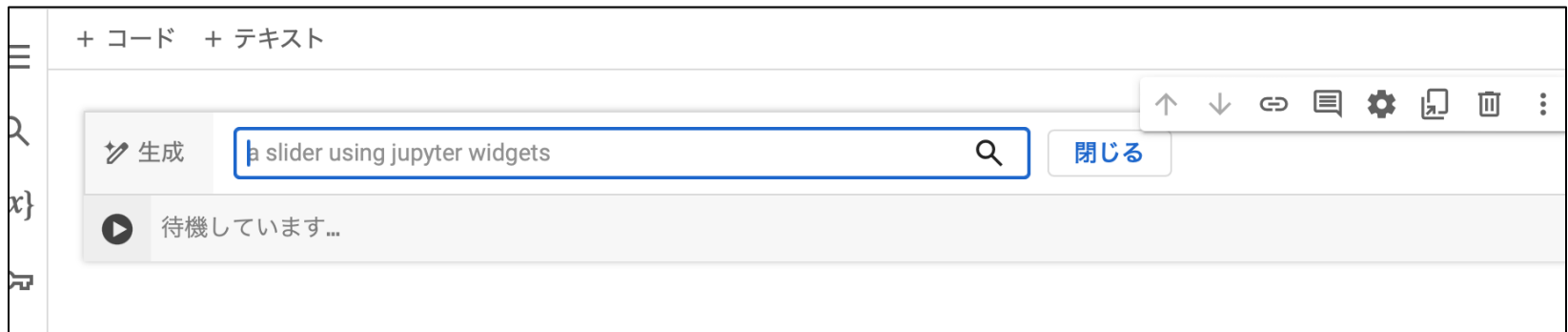
## 全角文字，日本語入力を避ける（２）

- できる限り全角文字，日本語は使わない
  - 全角の空白（スペース）に注意！（エラーになる＆見つけづらい）
  - ファイル名も半角英数字にしよう
    - OK：shincho\_taiju.xlsx, height\_weight.csv
    - NG：身長体重データ.csv
- 半角カタカナ（アイウ）とかもやめよう

# コメントアウト

- メモ書き
- 書いたけど一時的に実行したくない場合
- 行頭に # で一行コメントアウト
- ''' または """" で囲んだ部分をコメントアウト

# 補足：LLMを使ったプログラミング



# 補足：LLMを使ったプログラミング



The screenshot shows a user interface for generating code. At the top, there is a search bar with the text "サイコロをプログラミングしてください" (Please program a dice) and a "閉じる" (Close) button. Below the search bar, there are navigation icons and a warning message: "注意してコードをご使用ください" (Please use the code with caution). The main area displays the generated code, which is a Python function to roll a dice. The code is as follows:

```
# prompt: サイコロをプログラミングしてください

import random

def roll_dice():
    """サイコロを振って結果を返す関数"""
    result = random.randint(1, 6)
    return result

# サイコロを振る
dice_result = roll_dice()
print("サイコロの結果:", dice_result)
```

簡単なプログラムなら命令をすれば作ってくれる

# 動作確認は必須

## 生成コード支援

一部の Google サービスで、生成 AI 技術を利用したコード支援機能の試験運用版にアクセスできるようになりました。このページは、これに関連するサービスと機能に適用されます。

## 責任を持って使用する

Google の生成コード機能はまだ試験運用中であるため、提案されたコードまたはコーディングの説明はご自身の責任で使用してください。すべてのコードについて、エラー、バグ、脆弱性がな  
いか、ご自身の裁量でテストやレビューを慎重に行うことをおすすめします。

## ※ Google からのメッセージ

### 正しく動作するかの確認は必須

- 使い手がプログラムを読み書きできることを想定
- プログラミングに慣れた人には便利
- ヒントとして使うのも良い

# 本講義での利用

- 使う分には全然構いません
  - 当然ですが演習のプログラムをAIに書かせても何の演習にもならないので、自分自身の学習・理解の補助としてうまく使ってください
  - 最初は自分の力 & TAへの質問だけで試行錯誤して書いてみて、AIの生成と比較してみる、などもあり

# 演習資料

Playing with the Body-mass index

# 演習1：BMI

- 変数を使ってBMIを計算するコードを書いてみよう

ヒント：BMI は  $\text{体重} \div (\text{身長} \times \text{身長})$

※体重は kg, 身長は m

- 変数の値を変えて、それぞれ計算してみよう

- A君：1.7m, 62kg

- B君：1.6m, 75kg

- C君：1.8m, 73kg

## 演習2：関数の呼び出し

- 求めたBMIを四捨五入してみよう
  - ヒント：round
- 求めたBMIを切り捨ててみよう
  - ヒント：「python 切り捨て」でGoogleで検索
- 求めたBMIを切り上げてみよう
- 変数の値を変えて、それぞれ計算してみよう
  - A君：1.7m, 62kg
  - B君：1.6m, 75kg
  - C君：1.8m, 73kg

# 演習：進みが早い人用

## 演習3：print文の応用

- 変数aに5を代入して、
- print文で「aの中身は5です。」と表示してみよう。
  - ヒント) 「print format python」で検索

## 演習4：print文の応用

- 演習3で作成したエディタファイルに以下のprint文を追加してまとめて実行してみよう。
  - 「A君のBMIはxxxです。」と表示
  - 「B君のBMIはxxxです。」と表示
  - 「C君のBMIはxxxです。」と表示
  - ※) xxxの部分はそれぞれ計算したBMIが入る