

コンピュータリテラシ発展 ～Pythonを学ぶ～

第5回：Excel作業を自動化しよう

情報学部 情報学科 情報メディア専攻

清水 哲也 (shimizu@info.shonan-it.ac.jp)

今回の授業内容

今回の授業内容

- 前回の課題解説
- 準備：Google Driveをマウントする
- 準備：PythonでExcelを操作する
- Excelの値を表示する
- 課題

前回の課題解説

前回の課題解説

- 前回の課題の解答例を示します
- 解答例について質問があればご連絡ください

解答例

(解答例を作成後URL貼り付け)

準備：Google Driveをマウントする

Colab上でExcelファイルを扱うためには？

ColabでExcelファイルを扱う方法は複数あります

1. プログラムでExcelファイルを読み込む方法
2. ローカルにあるExcelファイルを読み込む方法
3. Google Driveをマウントして読み込む方法
4. Colabに直接アップロードして読み込む方法

Colab上でExcelファイルを扱うためには？

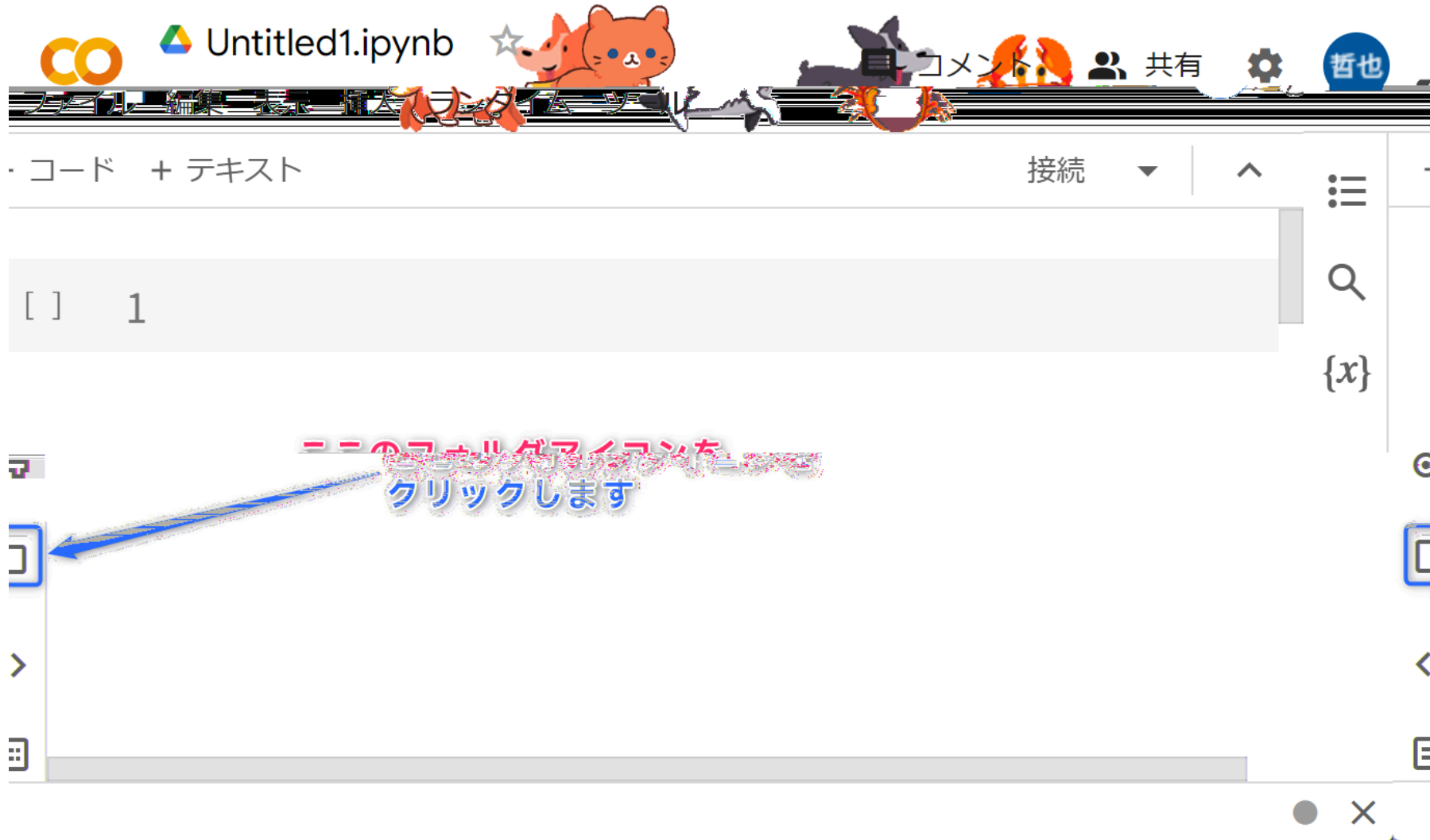
一番簡単な方法

- 4. 「Colabに直接アップロードして読み込む方法」
- ~~注意~~：ランタイムが終了するとファイルが削除されます

この授業では

- 3. 「Google Driveをマウントして読み込む方法」
- 作業しているファイルが削除されずに済むのでこの方法を採用します
- 基本的に今後この形式でファイル読み込みを行うので覚えておいてください

Google Driveをマウントする手順



Google Driveをマウントする手順

The screenshot shows the Google Colab interface. At the top, there's a header with the Google logo, a cartoon dog, and the text "Untitled1.ipynb". Below this is a menu bar with "ファイル", "編集", "表示", "挿入", "ランタイム", and "ツ". To the right of the menu bar are icons for "コメント", "共有", and a user profile with the name "哲也". Below the menu bar is a toolbar with icons for "ファイル", "コード", "テキスト", and "RAM". A green box highlights the Google Drive icon in the toolbar. A green arrow points from the text "このGoogle Driveマークがついたフォルダアイコンをクリックします" to the Google Drive icon. The file manager on the right shows a folder named "le_data" and a folder named "samp". At the bottom left, there's a status bar that says "80.66 GB が利用可能".

Untitled1.ipynb ☆

コメント 共有 哲也

ファイル 編集 表示 挿入 ランタイム ツー

+ コード + テキスト ✓ RAM ディスク

le_data

このGoogle Driveマークがついたフォルダアイコンをクリックします

80.66 GB が利用可能

ディスク

Google Driveをマウントする手順



The screenshot shows the Google Colab web interface. A modal dialog box is centered on the screen, asking for permission to access Google Drive files. The dialog contains the following text:

このノートブックに **Google** ドライブのファイルへのアクセスを許可しますか？

Google ドライブに接続すると、このノートブックで実行されたコードに対し、アクセス権が取り消されるまで Google ドライブ内のファイルの変更を許可することになります。

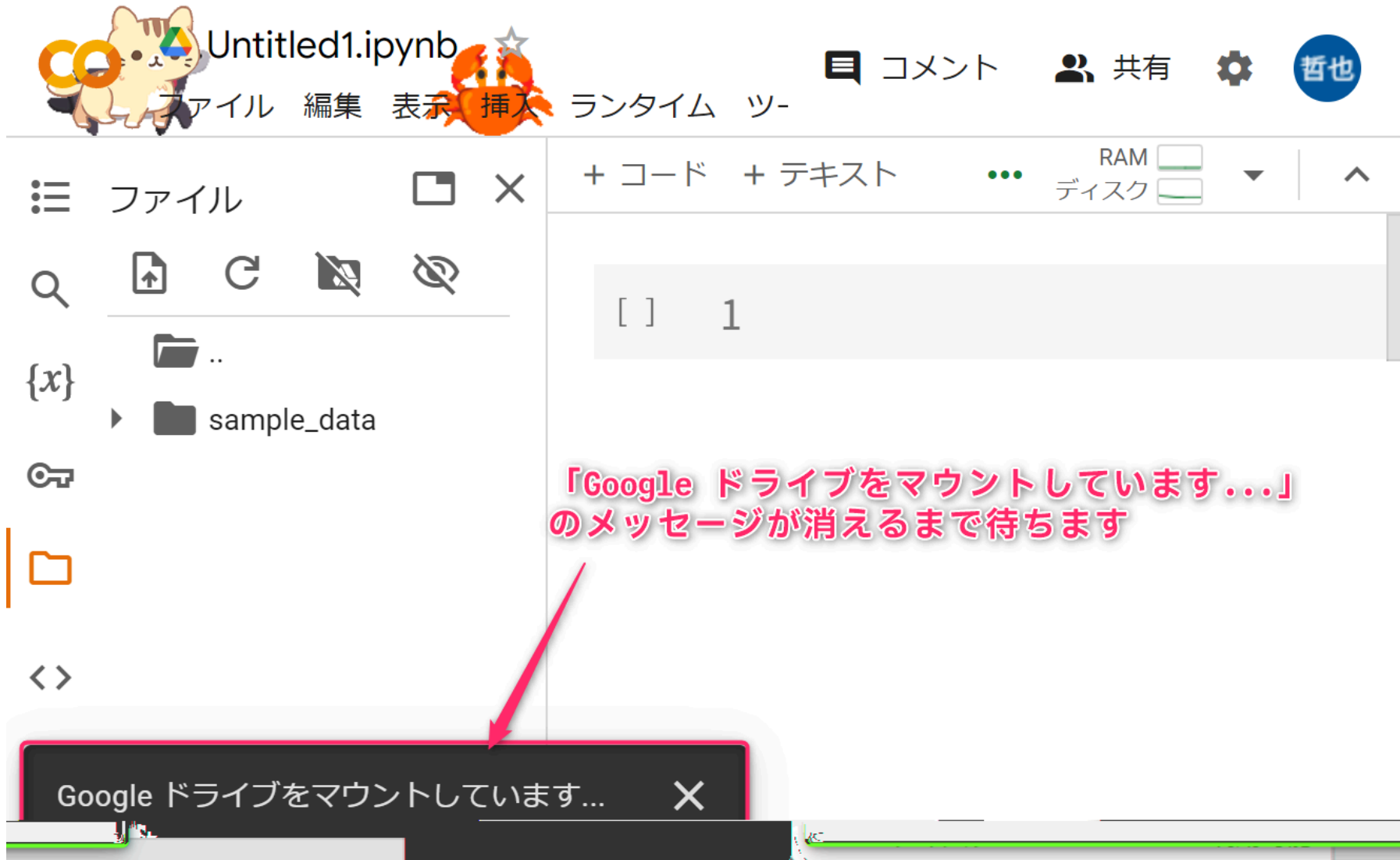
At the bottom of the dialog, there are two buttons: "スキップ" (Skip) and "Google ドライブに接続" (Connect to Google Drive). The "Connect to Google Drive" button is highlighted with a blue border and a blue arrow points to it from below.

Below the dialog, there is a red text overlay with a speech bubble pointing to the "Connect to Google Drive" button:

「Googleドライブに接続」をクリックします

The background interface shows the Colab header with "Untitled1.ipynb", navigation tabs (ファイル, 編集, 表示, 挿入, ランタイム, ツー), and user information (コメント, 共有, 哲也). The left sidebar shows a search icon, a code editor icon, and a file explorer icon.

Google Driveをマウントする手順



The screenshot shows the Google Colab interface for a notebook titled "Untitled1.ipynb". The top bar includes icons for Google Drive, a cat, and a star, along with tabs for "ファイル" (Files), "編集" (Edit), "表示" (View), "挿入" (Insert), "ランタイム" (Runtime), and "ツ" (Tools). The left sidebar shows a file explorer with a folder named "sample_data". The main area is a code editor with a single cell containing "[] 1". A red arrow points from a text box to a notification message at the bottom.

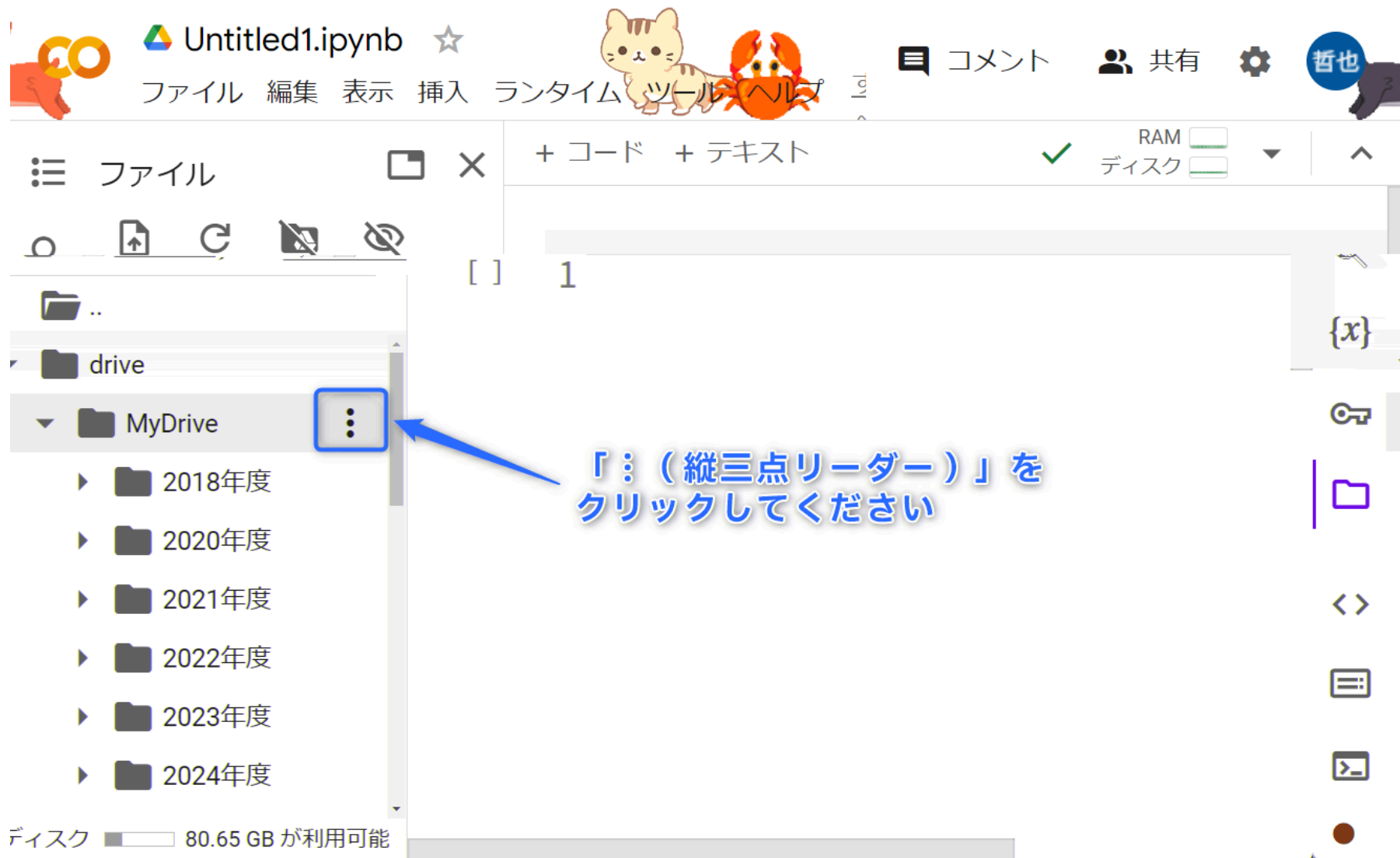
「Google ドライブをマウントしています...」
のメッセージが消えるまで待ちます

Google ドライブをマウントしています... X

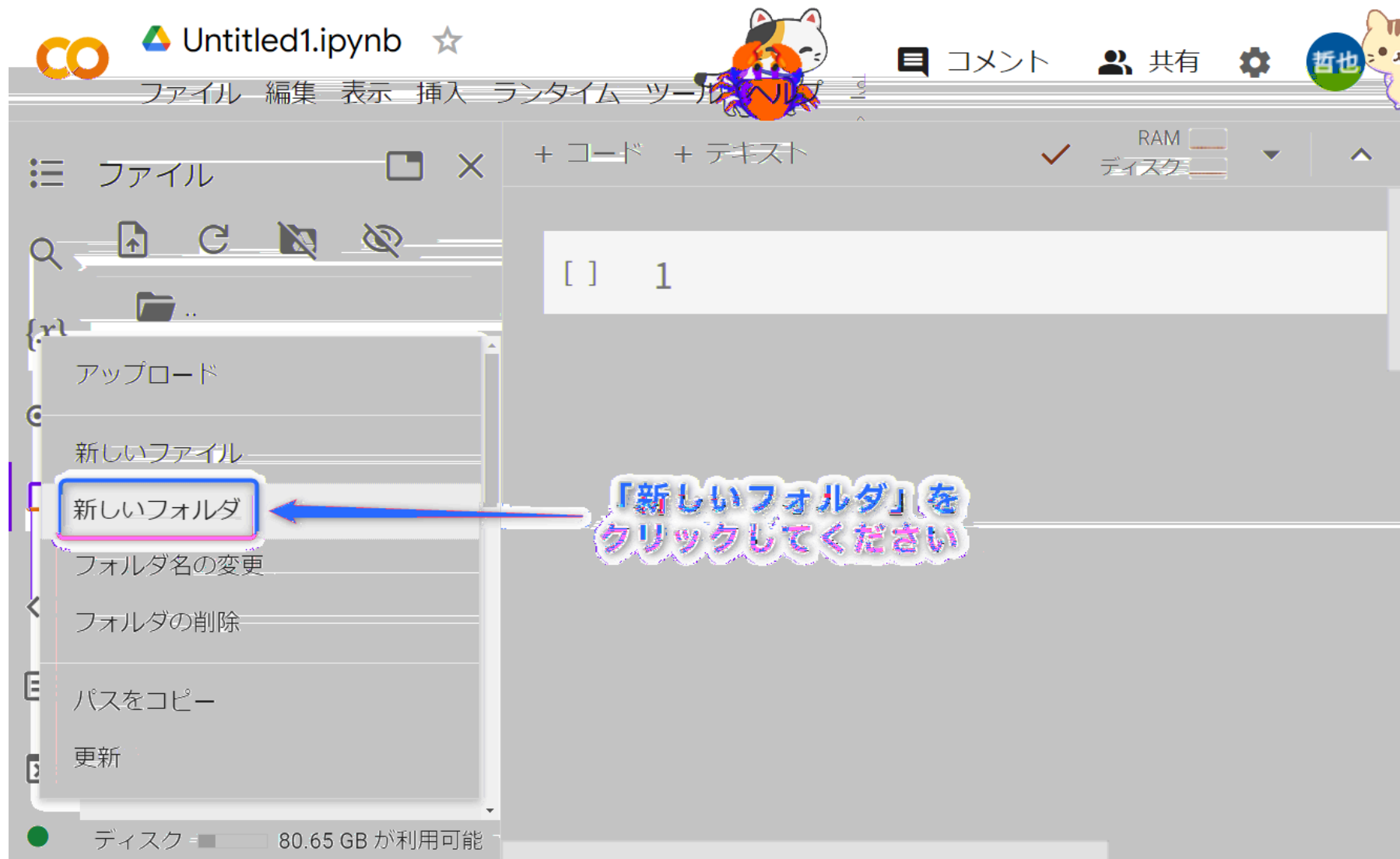
Drive上に作業用フォルダを作成

- MyDriveの直下にフォルダを作成します
- 作成方法
 - Colabから作成する方法
 - Driveから作成する方法
- フォルダ名はアルファベットのみにしてください
- 例：「Hatten」「CLHatten」 など

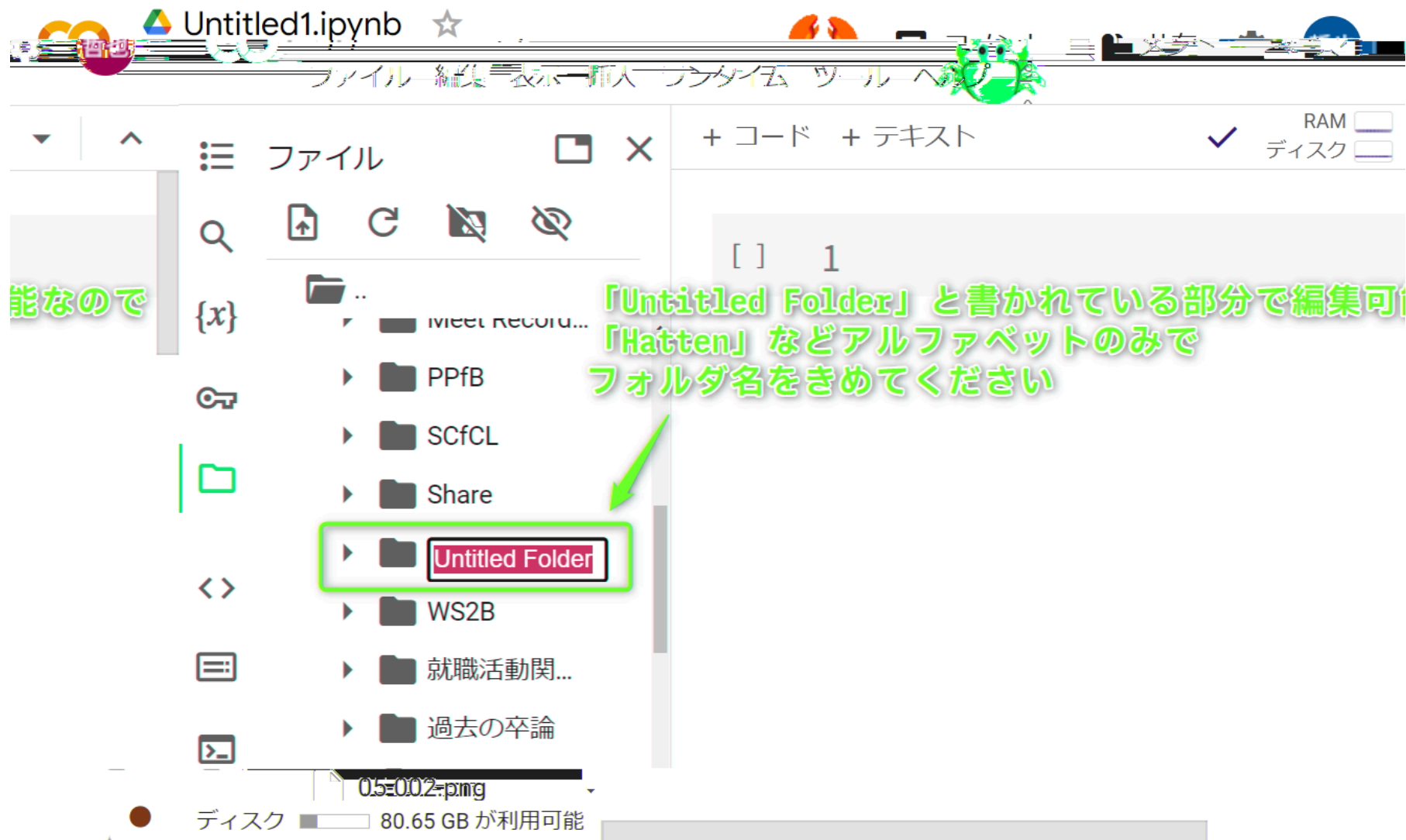
Colabからフォルダを作成する方法



Colabからフォルダを作成する方法



Colabからフォルダを作成する方法



準備：PythonでExcelを操作する

OpenPyXLについて

- PythonでExcelの読み書きをするためのライブラリです
- Pythonを使って次のような操作ができます
 - セルの値をPythonで読み書きする
 - 複数のExcelファイル間でデータをコピーする
 - フォント設定を変更する
 - Excelの行高や列幅を調整する
 - グラフを追加する

OpenPyXLについて

- OpenPyXLはサードパーティライブラリなので環境によってはインストールが必要になります
- Google Colabの場合はすでにインストールされているのでインストールは不要です

```
!pip install openpyxl
```

OpenPyXLについて

- OpenPyXLはライブラリなのでPython上で使うためにはインポートする必要があります
- インポートはNotebookのはじめに1回だけ実行すればそれ以降はインポートされた状態になります
- コーディングするうえで「`openpyxl`」を何度も書くのは面倒なので名前を「`op`」に変更してインポートします

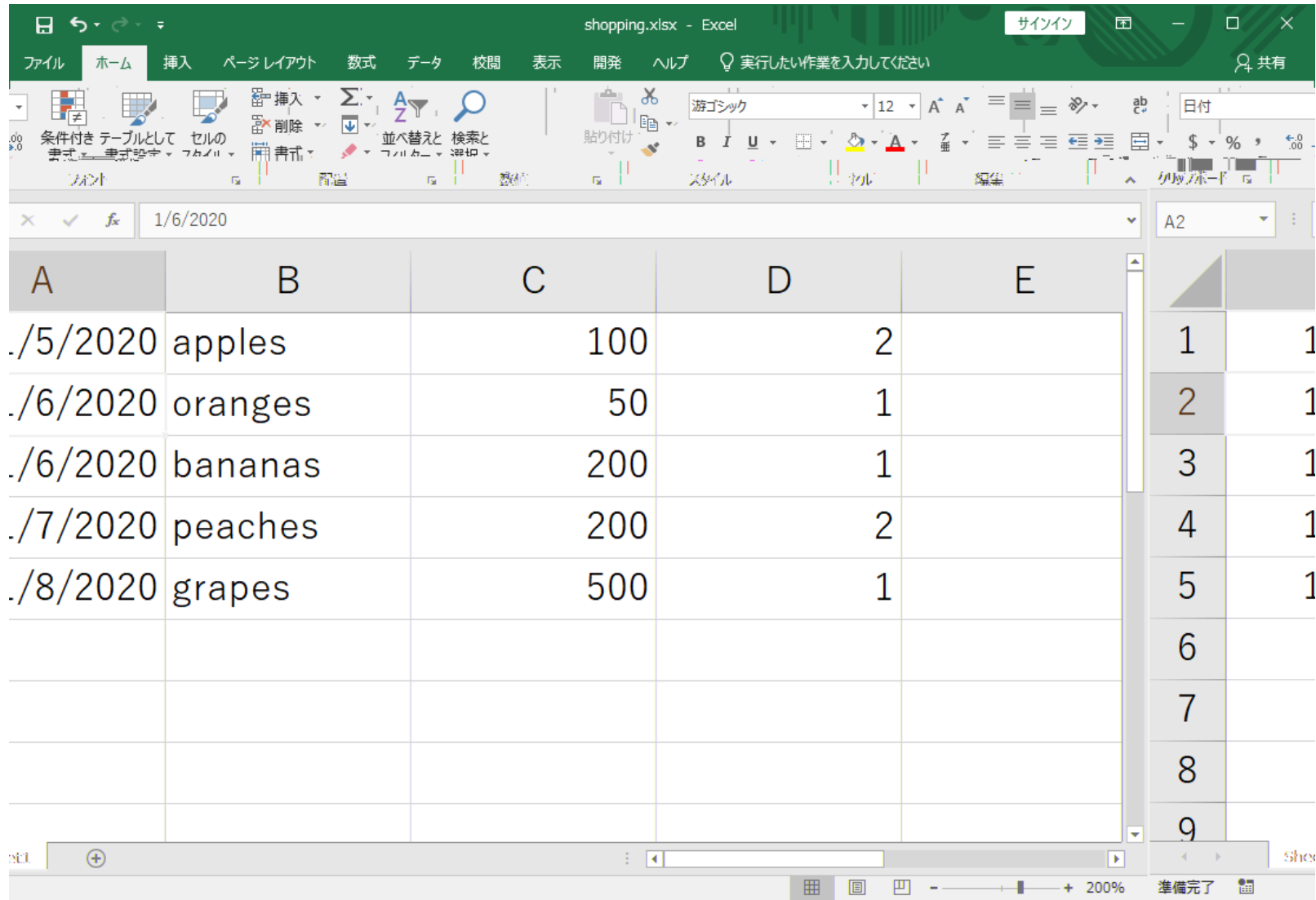
```
import openpyxl as op
```

Excelの値を表示する

Excelファイルの準備

- Moodleから「Shopping.xlsx」をダウンロードしてください
- MyDrive直下に作成した作業用フォルダにアップロードします
- アップロード方法
 - Colabからアップロードする方法
 - Driveからアップロードする方法

Excelファイルの中身を確認



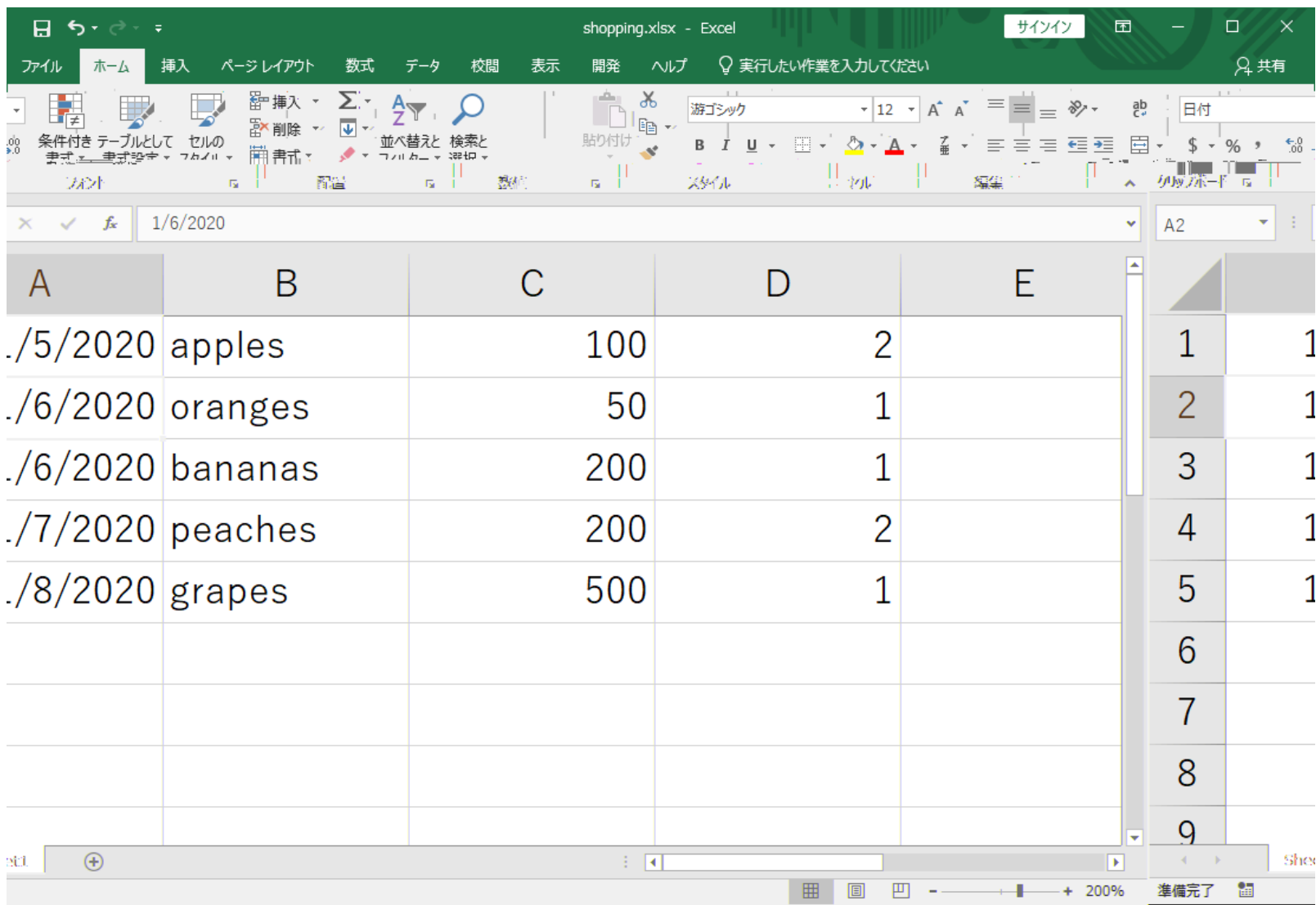
The screenshot shows an Excel spreadsheet with the following data:

A	B	C	D	E
./5/2020	apples	100	2	
./6/2020	oranges	50	1	
./6/2020	bananas	200	1	
./7/2020	peaches	200	2	
./8/2020	grapes	500	1	

The spreadsheet also includes a formula bar showing '1/6/2020' and a status bar at the bottom indicating '準備完了' (Ready).

ExcelとOpenPyXLのオブジェクト

- Excelファイル = Workbookオブジェクト



The screenshot shows the Microsoft Excel interface with a file named 'shopping.xlsx'. The ribbon is set to 'ホーム' (Home). The active cell is A2, containing the date '1/6/2020'. The table below lists items purchased over time.

A	B	C	D	E
./5/2020	apples	100	2	
./6/2020	oranges	50	1	
./6/2020	bananas	200	1	
./7/2020	peaches	200	2	
./8/2020	grapes	500	1	

ExcelとOpenPyXLのオブジェクト

- シート = Worksheetオブジェクト

shopping.xlsx - Excel

サインイン

ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表示 開発 ヘルプ 実行したい作業を入力してください 共有

貼り付け クリップボード フォント 配置 数値 スタイル セル 編集

A8

	A	B	C	D	E
1	1/5/2020	apples	100	2	
2	1/6/2020	oranges	50	1	
3	1/6/2020	bananas	200	1	
4	1/7/2020	peaches	200	2	
5	1/8/2020	grapes	500	1	
6					

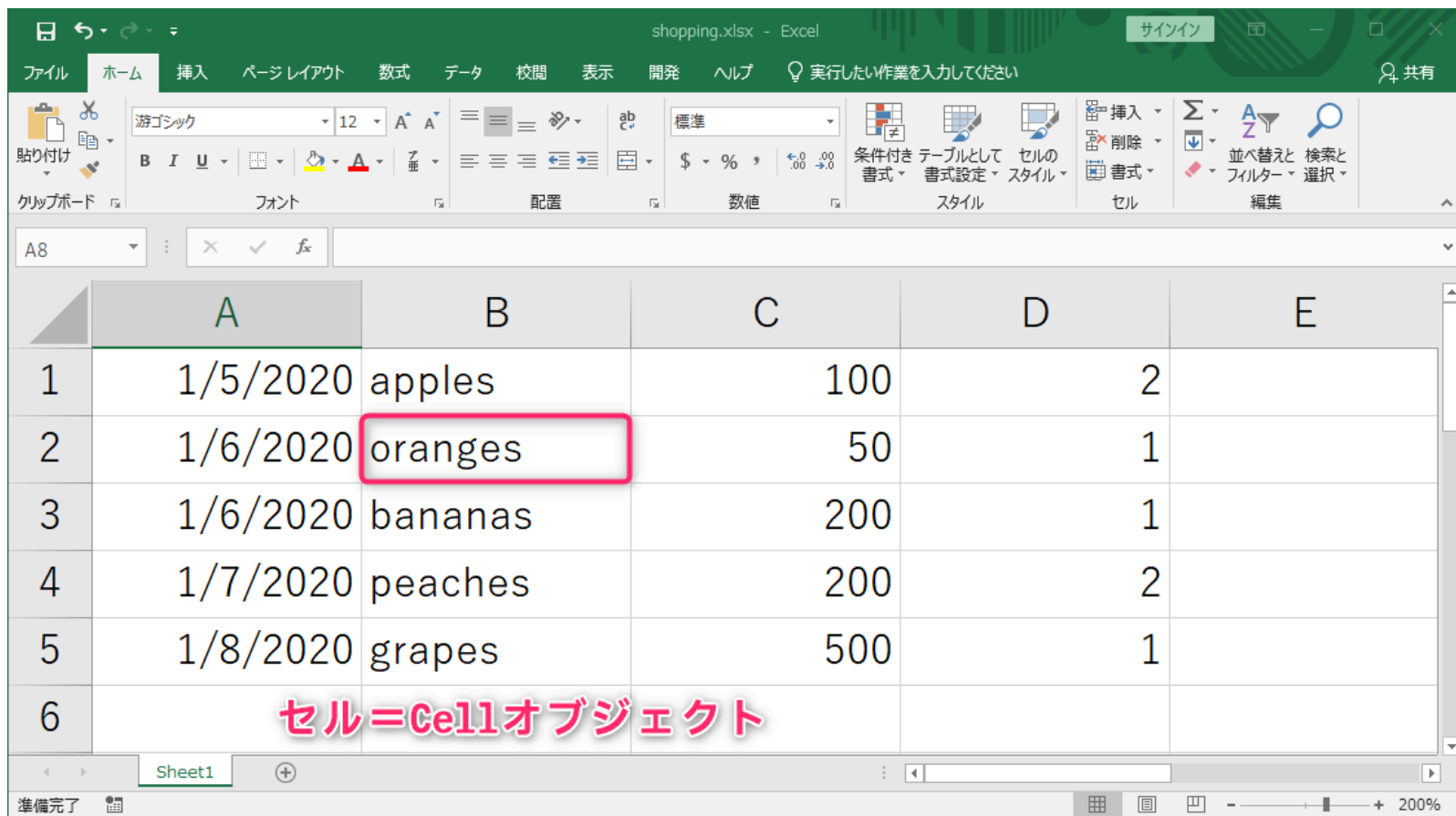
シート=Worksheetオブジェクト

Sheet1

準備完了 200%

ExcelとOpenPyXLのオブジェクト

- セル = Cellオブジェクト



shopping.xlsx - Excel

サインイン

ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表示 開発 ヘルプ 実行したい作業を入力してください 共有

貼り付け クリップボード フォント 配置 数値 スタイル セル 編集

	A	B	C	D	E
1	1/5/2020	apples	100	2	
2	1/6/2020	oranges	50	1	
3	1/6/2020	bananas	200	1	
4	1/7/2020	peaches	200	2	
5	1/8/2020	grapes	500	1	
6					

セル=Cellオブジェクト

Sheet1

準備完了 200%

指定したセルの値を取得する

- shopping.xlsx内のB1セルの値を取得して表示するプログラムを作成します
- 手順は以下の通りです
 - i. `load_workbook()` メソッドでExcelファイルを読み込みます
 - ii. ワークシートとセルを読み込みます
 - iii. 読み込んだ値を表示します

指定してセルの値を取得する

- `load_workbook()` メソッドでExcelファイルを読み込みます
 - セルを取得するためにExcelファイルを取得する必要があります
 - `load_workbook()` メソッドの引数にファイル名を指定します
 - ファイル名だけではなくファイルまでのパスも必要です
 - Colabの場合：対象ファイルを右クリックして「Copy path」でOK

指定したセルの値を取得する

- ワークシートとセルを読み込みます
 - ワークブックを読み込んだのでワークシートとセルを指定します
 - 対象のシートは「**Sheet1**」, セルは「**B1**」です
 - さらに値を取得するので `value` プロパティを指定します

指定したセルの値を取得する

1. `load_workbook()` メソッドでExcelファイルを読み込みます
2. ワークシートとセルを読み込みます
3. 読み込んだ値を表示します

```
wb = op.load_workbook('/content/drive/MyDrive/???/shopping.xlsx')  
value = wb['Sheet1']['B1'].value  
print(value)
```

「???」は各自作成したフォルダ名を入れてください

複数のセルをまとめて取得する

- shopping.xlsxの**B**列をすべて取得してみます
- データが入っているセルがすでにわかっている場合
 - 対象：セルB1～B5
 - 繰り返し処理を利用します
 - セルの行と列を個別に指定します

複数のセルをまとめて取得する

- セルの行と列を個別に指定する方法
 - row (行)と colum (列)を使って個別に指定します

Workbookオブジェクト.Cell(row=行数番号, colum=列数番号)

列数番号

	1	2	3	4	5
	A	B	C	D	E
1	1/5/2020	apples	100	2	
2	1/6/2020	oranges	50	1	
3	1/6/2020	bananas	200	1	
4	1/7/2020	peaches	200	2	
5	1/8/2020	grapes	500	1	
6					

複数のセルをまとめて取得する

- shopping.xlsxのB列をすべて取得してみます
- データが入っているセルがすでにわかっている場合
 - 対象：セルB1～B5
 - 繰り返し処理を利用します
 - セルの行と列を個別に指定します

```
wb = op.load_workbook('/content/drive/MyDrive/???/shopping.xlsx')
sheet = wb['Sheet1']

for i in range(1,6):
    print(sheet.cell(row=i, column=2).value)
```

list()関数を使ってセルをまとめて取得する

- range() 関数
 - 最終行を事前に調べる必要があります
 - 最終行を指定してCellオブジェクトを取得します
- list() 関数
- 指定した列番号や行番号に存在するExcelの Cell オブジェクトをまとめて取得します

list()関数を使ってセルをまとめて取得する

- list() 関数の使い方

```
list(Worksheetオブジェクト.columns)[列番号]
```

~~注意~~：listを使う場合列番号はPython上の数え方になりますので**B列**の場合は「**1**」を指定します

- list() 関数を使ってB列のCellオブジェクトをまとめて取得します

```
wb = op.load_workbook('/content/drive/MyDrive/???/shopping.xlsx')
sheet = wb['Sheet1']

for cell in list(sheet.columns)[1]:
    print(cell.value)
```

課題

課題

- Moodleにある「SCfCL-5th-prac.ipynb」ファイルをダウンロードしてColabにアップロードしてください
- 課題が完了したら「File」>「Download」>「Download .ipynb」で「.ipynb」形式でダウンロードしてください
- ダウンロードした **.ipynb** ファイル をMoodleに提出してください
- 提出期限は **5月23日(木) 20時まで** です