

# データ科学基礎演習B（1）

データ科学科目部会



# 演習環境を整えよう

## Google Colaboratory の準備



# 演習を始める前に

- Pythonの演習はWEBブラウザ上で行います
- 下記ブラウザのいずれかを用意ください
  - Google Chrome
  - Firefox
  - ※ Microsoft Edge, Internet Exploreでは動作しません
- 参考書籍
  - 「プログラミング演習 Python 2019」
    - 喜多 一(京都大学)
    - <http://hdl.handle.net/2433/245698>



# まずはじめに演習の準備をしましょう(1)

- パソコンにログインし, ブラウザを開く
  - Googleアカウントが無い場合はアカウント作成
    - <https://accounts.google.com/signup>
  - 下記URLを入力してGoogle Colaboratoryにアクセス
    - <https://colab.research.google.com>





## Colaboratory とは

Colaboratory (略称: Colab) では、ブラウザから Python を記述し実行できるほか、次の特長を備えています。

- 構成が不要
- GPU への無料アクセス
- 簡単に共有

Colab は、**学生**、**データサイエンティスト**、**AI リサーチャー**の皆さんの作業を効率化します。詳しくは、[Colab のご紹介](#)をご覧ください。下からすぐに試してみることもできます。

### ▼ はじめに

ご覧になっているドキュメントは静的なウェブページではなく、**Colab ノートブック**という、コードを記述して実行できるインタラクティブな環境です。

たとえば次の**コードセル**には、値を計算して変数に保存し、結果を出力する短い Python スクリプトが含まれています。

```
[ ] 1 seconds_in_a_day = 24 * 60 * 60
    2 seconds_in_a_day
```

👤 86400

上記のセルのコードを実行するには、セルをクリックして選択し、コードの左側にある実行ボタンをクリックするか、キーボードショートカット「command+return」または「Ctrl+Enter」を使用します。コードはセルをクリックしてそのまま編集できます。

1 つのセルで定義した変数は、後で他のセルで使用できます。

```
[ ] 1 seconds_in_a_week = 7 * seconds_in_a_day
    2 seconds_in_a_week
```

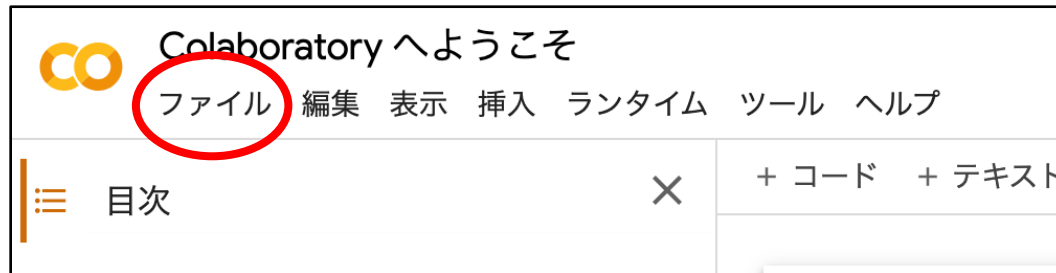
👤 604800

Colab ノートブックを使用すると、**画像**、**HTML**、**LaTeX** などと合わせて、**実行可能コード**と**リッチテキスト**を 1 つのドキュメントに結合できます。Colab ノートブックを独自に作成すると、Google ドライブアカウントに保存されます。Colab ノートブックは、同僚や友人と簡単に共有し、コメントの記入や編集をしてもらうことができます。詳細については [Colab の概要](#) をご覧ください。新しい Colab ノートブックを作成するには、上にある [ファイル] メニューを使用するか、[新しい Colab ノートブックを作成する](#) をクリックします。

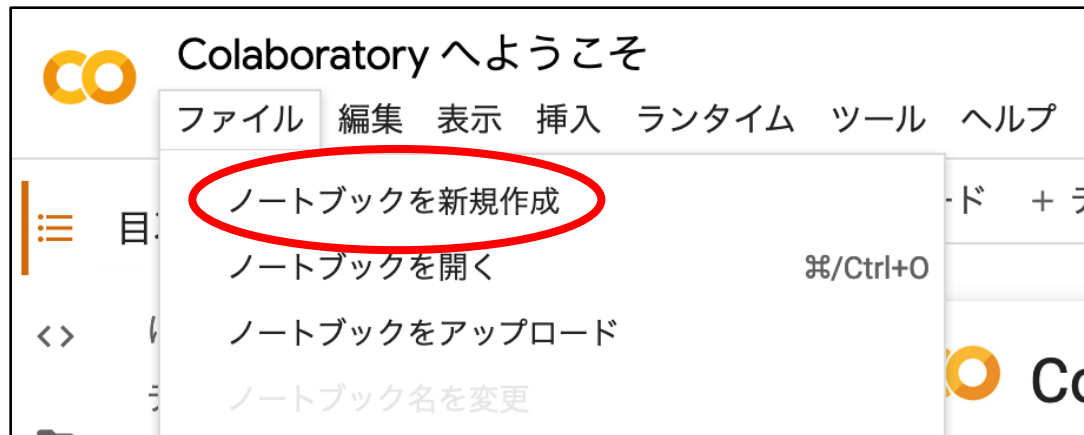


# まずはじめに演習の準備をしましょう(2)

- 演習用のノートブックを作成しましょう
  - 「ファイル」メニューをクリック



- 「ノートブックを新規作成」をクリック



# Google Colaboratory の使い方

The screenshot shows the Google Colaboratory interface. At the top, the file name `sample.ipynb` is highlighted with a red box, and a red arrow points to it with the text: **ファイル名を変更する場合はここをクリック**. Below the file name is a menu bar with options: `ファイル`, `編集`, `表示`, `挿入`, `ランタイム`, `ツール`, `ヘルプ`, and a status message `すべての変更を保存しました`.

On the left side, there is a sidebar with icons for a menu, code editor, and file explorer. The main area shows two code cells. The first cell, labeled `[5]`, contains the following code:

```
1 # 何かを入力して Shift + Enter で実行
2
3 x = 10
```

The second cell, labeled `[6]`, contains the following code:


```
1 # 変数だけを入力すると, その値が表示される
2
3 x
```

Below the second cell, the output `10` is displayed, indicated by a red arrow and the text: **実行結果はコードの下に表示**.

At the bottom, there is a toolbar with a play button (run icon) and a cursor. A red arrow points to the cursor with the text: **1. Pythonコードをここに入力  
2. 「Shift + Enter」でコードを実行**.



# Google Colaboratory の使い方(デモ)

 nulit2020\_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](#)

+ コード + テキスト

コメント 共有 設定

RAM ディスク 編集

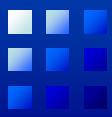
```
[2] 1 x = 10 + 1
    2 x

11
```

1 |

↑ ↓ 🔗 設定 📄 🗑️ ⋮





# Pythonの初歩

## プログラミングに触れてみよう



# 数の表現(1)

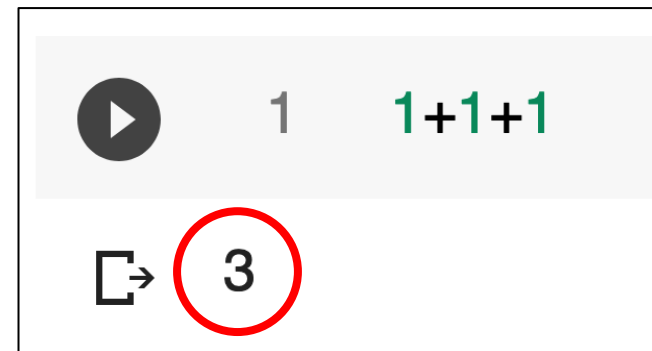
- コンピュータを使って次の計算を行うとどのような結果が得られるでしょうか？

$$1 + 1 + 1 = ?$$

- Google Colaboratory に下記のようなコードを入力して結果を確認してみましょう



Shift+Enter





## 数の表現(2)

- コンピュータを使って次の計算を行うとどのような結果が得られるでしょうか？

$$0.1 + 0.1 + 0.1 = ?$$

- Google Colaboratory に下記のようなコードを入力して結果を確認してみましょう



Shift+Enter



?



## 数の表現(3)

- コンピュータは小数点以下の数値を正しく扱うことができない！！

$$0.1 + 0.1 + 0.1 \neq 0.3$$

- コンピュータは2進数で数を表現している
  - 整数 → 2の指数乗の和で表現
    - $9 = 8 + 1 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (1001)_2$
  - 小数 →  $1/2$ の指数乗の和で表現 → 0.1にならない
    - $0.1 = \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \dots = 1 \times 2^{-4} + 1 \times 2^{-5} + 1 \times 2^{-8} + \dots$



# 四則演算(1)

- 数学の式とほぼ同じ記法(+, -)
  - 足し算 →  $1+2$
  - 引き算 →  $5-2$  や  $-2+5$
- 掛け算は \* (アスタリスク), 割り算は / (スラッシュ)
  - 掛け算 →  $11*35$
  - 割り算 →  $12/4$
- 特殊な計算(%, \*\*, //)
  - 剰余 →  $10\%3$
  - べき乗 →  $2**8$
  - 整数除算 →  $10//4$



## 四則演算(2)

- 次の式はどういう順番で計算しますか？

$$(1 + 2) \times (3 + 4) = ?$$

- Pythonでも()内の計算が優先される



1

(1+2)\*(3+4)



21



## 四則演算(3)

- 次の式の計算結果はどうなるでしょうか？

$$1 + 2 \times (3 + 4) = ?$$

- Pythonでも掛け算や割り算の計算が優先される



1

`1+2*(3+4)`



15



## 四則演算(4)

- 次の式の計算結果はどうなるでしょうか？

$$2 \times \sqrt{2} = ?$$

- 掛け算よりも乗算が優先される



1

$2 * 2^{**} 0.5$



2.8284271247461903





## 四則演算(5)

- 次の2つの式の計算結果を求めてみよう

$$12 \times (2 - \sqrt{3})$$

$$6\sqrt{2 - \sqrt{3}}$$



# 文字の表現

- Pythonの場合
  - 文字(列)を `"` (ダブルクォート)もしくは `'` (シングルクォート)で囲む

`"a"` もしくは `'a'`

- コンピュータは文字を数値に置換して記憶
  - 文字 `'a'` の数値を確認してみよう

文字 → 数値

`ord('a')`

数値 → 文字

`chr(97)`



# 特殊な文字(エスケープシーケンス)

- プログラムでは目に見えない文字も扱います
  - 改行, タブ, etc.
- エスケープシーケンスとは？
  - Windowsの場合 → ￥ (円マーク)
  - MacやLinuxの場合 → \ (バックスラッシュ)
- 特殊な文字はエスケープシーケンスを付ける
  - 改行 → ￥n
  - タブ → ￥t
  - エスケープシーケンス → ￥￥



# シングルクォートとダブルクォートの違い

- 基本的に同じ(他の言語では違う場合あり)
- シングルクォート 'a' で ' を出力する場合
  - ' の前にエスケープシーケンス → '¥'
  - ダブルクォートはそのまま使える → ''
- ダブルクォート "a" で " を出力する場合
  - " の前にエスケープシーケンス → "¥"
  - シングルクォートはそのまま使える → "'"

# 画面へ文字を出力してみよう

- print 関数を使うと画面に文字を出力できる

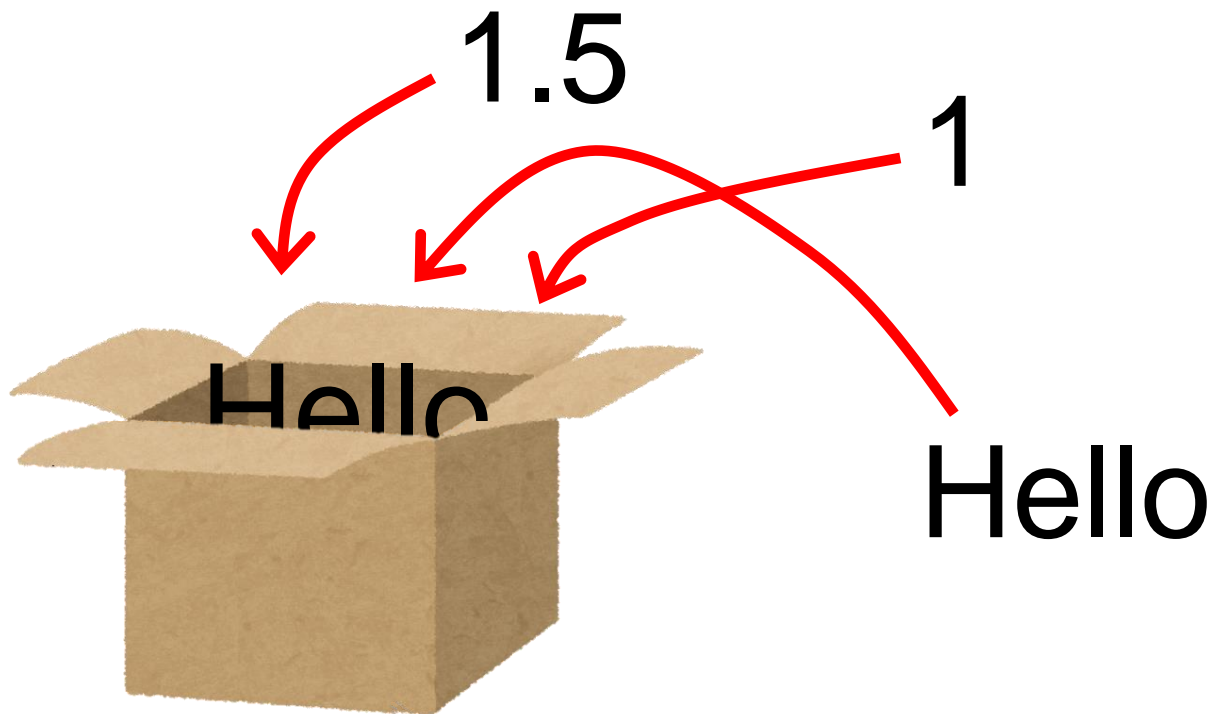
```
print('出力したい文字列')
```





# 変数

- 値(数値, 文字, etc.)を入れておく箱
  - 数学で使われる変数をより汎用的にしたイメージ





# 変数

- 値(数値, 文字, etc.)を入れておく箱
  - 数学で使われる変数をより汎用的にしたイメージ
- 変数は名前(ラベル)を付けて管理します
- 複数の変数でいろいろな値を記憶できます





## 代入(1)

- 次の式を見て  $y$  はいくつだと思いますか？

$$x = 3$$

$$y = 2x + 1$$





# 代入(1)

- 次の式を見て  $y$  はいくつだと思いますか？


$$x = 3$$

$$y = 2x + 1$$

- 「**=**」は数学の等号ではなく、変数への値の代入
  - 「右辺の値」を「左辺の変数」に**代入**する
  - 代入すると前の値は置き換わる
- 式(コード)は上から順に実行される
  - 式の順番を入れ替えると答えが変わる場合あり



# 代入(2)

 nulit2020\_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](#)


コメント 共有









+ コード + テキスト

RAM  ディスク  編集

[1] 1 y = 0  
2 y  
0

[3] 1 x = 3  
2 y = 2 \* x + 1  
3 y  
7

 1

# キーボードからの値の入力(1)

- input 関数でキーボードから値を入力できる

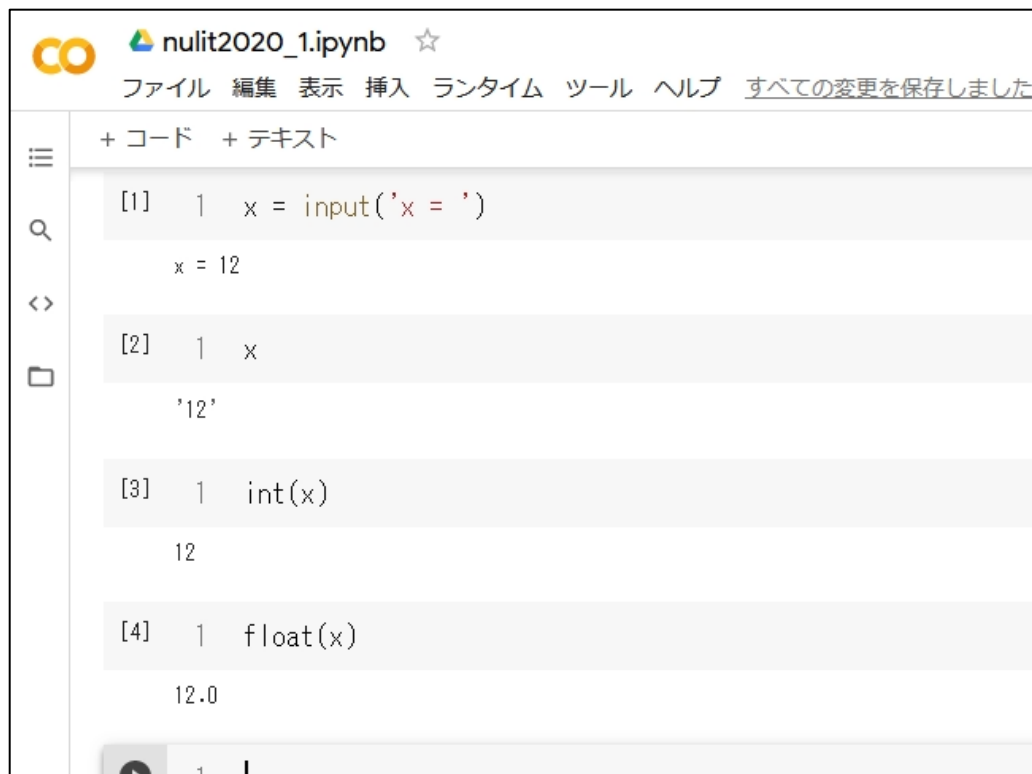
```
x = input('x = ')
```



The screenshot shows a Jupyter Notebook titled "nulit2020\_1.ipynb". The interface includes a menu bar with options: ファイル, 編集, 表示, 挿入, ランタイム, ツール, ヘルプ. Below the menu bar, there are tabs for "+ コード" and "+ テキスト". The notebook contains two code cells. The first cell, labeled "[1]", contains the code `x = input('x = ')` followed by `x = 12`. The second cell, labeled "[2]", contains the code `x` followed by `'12'`. At the bottom of the notebook, there is a play button icon and the number "1".

# ■ ■ ■ キーボードからの値の入力(2)

- input 関数の戻り値は文字(列)
  - 数値として扱う場合は変換が必要
    - 整数への変換: `int(x)`
    - 浮動小数への変換: `float(x)`



The screenshot shows a Jupyter Notebook interface with the title 'nulit2020\_1.ipynb'. The notebook contains four code cells demonstrating the use of the `input()` function and its conversion to numerical types.

```
[1] 1 x = input('x = ')
      x = 12

[2] 1 x
      '12'

[3] 1 int(x)
      12

[4] 1 float(x)
      12.0
```



# キーボードからの値の入力(3)

nulit2020\_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

コメント 共有 編集

+ コード + テキスト

RAM  
ディスク

```
[1] 1 x = input('x = ')
    2 y = input('y = ')

```

```
x = 1
y = 2

```

```
[2] 1 x + y
    '12'

```

```
[3] 1 x = int(x)
    2 y = int(y)

```

```
[4] 1 x + y
    3

```

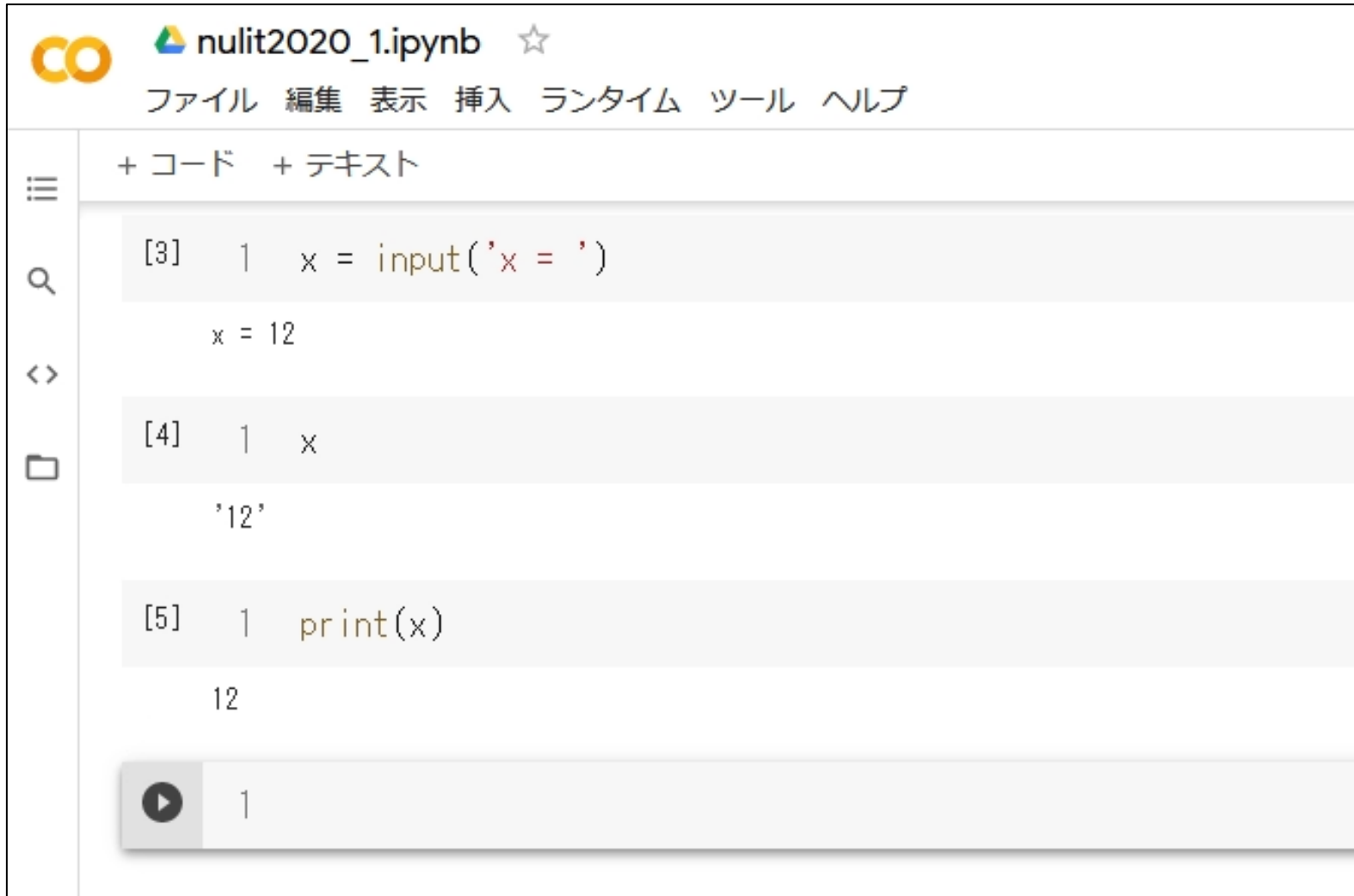
```
1 |
```

↑ ↓ ↻ ⌨ ⚙ 📄 🗑 ⋮



# 画面へ文字を出力してみよう(再び)

- print 関数で変数の値を画面に出力できる



The screenshot shows a Jupyter Notebook titled "nulit2020\_1.ipynb". The interface includes a top bar with the Colab logo, the file name, and a star icon, followed by a menu bar with options: ファイル, 編集, 表示, 挿入, ランタイム, ツール, ヘルプ. Below the menu bar, there are tabs for "+ コード" and "+ テキスト". The notebook contains three code cells:

```
[3] 1 x = input('x = ')
```

`x = 12`

```
[4] 1 x
```

`'12'`

```
[5] 1 print(x)
```

`12`

At the bottom, there is a run button (a play icon) and a text input field containing the number "1".



# 本日の課題

- 下記URLから演習課題のノートブックを自身のGoogle Driveにコピーし, Google Collaboratoryを起動して各設問に回答すること
  - <https://bit.ly/3zc3vZR>
- 演習課題を提出する際は, ノートブックのURLを共有し, そのURLをNUCTに提出すること
  - 課題の回答とURL共有の方法は次で紹介(動画)



# 演習課題の進め方



課題4 ☆

共有

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

+ コード + テキスト | ドライブにコピー

接続 編集

## Python演習課題 1

※コードを実行する際は、▶ ボタンをクリックするか、ShiftキーとEnterキーを同時に押してください。

### 課題①

print関数を用いて、"Hello, world!"を表示してみましょう。

```
[ ] 1 10
```



### 課題②

input関数を用いてキーボードから名前を入力し、print関数を用いて入力された名前を表示するプログラムを作ってください。

```
[ ] 1
```

### 課題③

input関数を用いてキーボードから2つの文字列を受け取り、その2つを連結した文字列を表示するプログラムを作ってください。

```
[ ] 1
```

### 課題④





# 演習課題の提出方法



課題4 のコピー ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 最終保存: 9:31



コメント

共有



+ コード + テキスト

接続 ▾

編集



## Python演習課題 1

※コードを実行する際は、▶ ボタンをクリックするか、ShiftキーとEnterキーを同時に押してください。

### 課題①

print関数を用いて,      \_2024秋\_末尾スライド追加前

[ ] 1

### 課題②

input関数を用いてキーボードから名前を入力し, print関数を用いて入力された名前を表示するプログラムを作ってください。

[ ] 1

### 課題③

input関数を用いてキーボードから2つの文字列を受け取り, その2つを連結した文字列を表示するプログラムを作ってください。

[ ] 1

### 課題④



# 課題提出時の注意点

- 課題提出の際には提出前に必ず以下2点を確認すること
  - Google Collaboratoryの共有設定の際には『リンクを知っている全員』にチェックを入れる
  - 課題提出時に貼り付けたURLの末尾が「?usp=sharing」となっている（共有設定を開き、「リンクのコピー」ボタンを押して共有用のURLをコピー＆ペーストする）