

# Python プログラミング基礎

九州大学 数理・データサイエンス教育研究センター

# この教材の説明

- 「プログラムって何？」「なんか怖い」「Pythonって美味しいもの？」と思っている**超初心者向け**です
- なので、内容は「最初のハードルを越える」ことを目指しており、**基本中の基本のみ**です
  - とはいえ、この教材の内容さえ押さえておけば、結構いろいろできます
- 「いままでコンピュータって（人が作った）アプリ使うばかりだったけど、**自分のやりたいことをコンピュータにやらせるのも面白いかも**」と思ってくれる人を少しでも増やしたい

# 全体の構成

- 教材 1
  - インTRODクション～プログラムってなんだ？
  - プログラミング言語 Python ～無料で遊べるのに凄い
  - 【準備】Gogole Colaboratoryへ行こう！
  - 文字を表示してみよう ～ 記念すべき最初のプログラム
- 教材 2
  - 計算させてみよう ～ たす, ひく, かける, わる
  - 変数 ～ 電卓を越える！
  - If文 ～ ますますプログラムっぽく
  - 便利なコメントアウト ～ ちょっと休憩
  - 数当てゲームを作ってみよう ～ ゲームの基本中の基本
- 教材 3
  - For文 ～ 人間には面倒な繰り返しを簡単に
  - 【寄り道】コンピュータの計算精度 ～ 意外と正確じゃない
  - 再びゲームを作ってみよう ～ 工夫次第でどんどん楽しく
- 教材 4
  - List ～ 複数の変数を固めたもの
  - 関数とライブラリ ～ ひみつ道具で遊ぼう
- 付録
  - if文のちょっと進んだ使い方
  - AIを用いたプログラミング
  - Google Driveとの連携

## 2タイプのスライド

- 青いタイトルのスライド
  - 読んで学ぶスライド
  - なるべく楽しい感じにしています

74

実は先ほどのゲームの例に出てくる  
「条件に応じてやることを変える」

- 森をふらふら歩く
- モンスターに会う
  - 逃げる?
    - Yesなら「1. 森をふらふら歩く」に戻る
    - Noなら次の「3. 戦う」に進む
- 戦う
  - モンスターを攻撃する
  - モンスターのHP(ヒットポイント)が減る
  - モンスターのHPがゼロになった?
    - Yesならモンスターを倒したので、「1. 森をふらふら歩く」に戻る
    - Noなら、「3. 戦う」に戻って攻撃を繰り返す

逃げる? 条件

Yes No

1. 1に戻る 3. 3に進む やること

---

HP=0? 条件

Yes No

1. 1に戻る 3. 3に進む やること

- 緑のタイトルのスライド
  - 実際にプログラムを作って学ぶスライド
  - なるべく楽しい題材を選んでます

111

いままで学習した内容だけでできる「進んだ練習」:  
ちょっとしたロールプレイングゲーム!

```

import random

print("あなたは勇者です。")
print("モンスターがあらわれた!")

print("0: たたかう")
print("1: にげる")
action = int(input("どうする? 0か1を入力してください: "))

if action == 0:
    player = random.randint(1, 6)
    monster = random.randint(1, 6)

    print(f"あなたの攻撃: {player}")
    print(f"モンスターの攻撃: {monster}")

    if player > monster:
        print("モンスターをたおした!")
    else:
        print("やられてしまった...")
else:
    print("あなたはにげだした!")

```

**実行例**

```

あなたは勇者です。
モンスターがあらわれた!
0: たたかう
1: にげる
どうする? 0か1を入力してください: 0
あなたの攻撃: 6
モンスターの攻撃: 2
モンスターをたおした!

```

ifの中に、  
またifが!

# Python プログラミング基礎 教材 1

九州大学 数理・データサイエンス教育研究センター

# 全体の構成

## ● 教材 1

- インTRODクション～プログラムってなんだ？
- プログラミング言語 Python ～無料で遊べるのに凄い
- 【準備】Gogole Colaboratoryへ行こう！
- 文字を表示してみよう ～ 記念すべき最初のプログラム

## ● 教材 2

- 計算させてみよう ～ たす, ひく, かける, わる
- 変数 ～ 電卓を越える！
- If文 ～ ますますプログラムっぽく
- 便利なコメントアウト ～ ちょっと休憩
- 数当てゲームを作ってみよう ～ ゲームの基本中の基本

## ● 教材 3

- For文 ～ 人間には面倒な繰り返しを簡単に
- 【寄り道】コンピュータの計算精度 ～ 意外と正確じゃない
- 再びゲームを作ってみよう ～ 工夫次第でどんどん楽しく

## ● 教材 4

- List ～複数の変数を固めたもの
- 関数とライブラリ ～ ひみつ道具で遊ぼう

## ● 付録

- if文のちょっと進んだ使い方
- AIを用いたプログラミング
- Google Driveとの連携

# イントロダクション

プログラムってなんだ？

# (いきなりですが) オムレツを作る

## 1. 材料の準備

- 卵 (2個)
- 塩 (ひとつまみ)
- コショウ (ひとつまみ)
- バター (5g)



## 2. 下準備

- ボウルに卵を割り入れる
- 塩とコショウを加える
- 卵がなめらかになるまでよくかき混ぜる



## 3. メイン処理

- 中火でフライパンを加熱する
- バターを入れて溶かす
- 卵液を流し込む
- 半熟になったらオムレツを半分に折りたたむ



## 4. 完了処理

- お皿にオムレツをのせる
- 完成したオムレツを盛り付けて提供する





# RPGにおけるモンスターとの<sub>一方的</sub>戦闘

1. 森をふらふら歩く
2. モンスターに出会う

## ●逃げる？

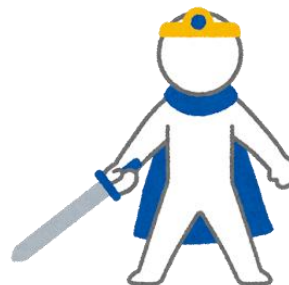
- Yesなら「1. 森をふらふら歩く」に戻る
- Noなら次の「3. 戦う」に進む



## 3. 戦う

- モンスターを攻撃する
- モンスターのHP(ヒットポイント)が減る
- モンスターのHPがゼロになった？

- Yesならモンスターを倒したので、「1. 森をふらふら歩く」に戻る
- Noなら、「3. 戦う」に戻って攻撃を繰り返す





# 家全体を大掃除する

## 1. 掃除道具を準備する

- 掃除機、雑巾、洗剤、ゴミ袋を用意

## 2. 各部屋を順番に掃除する

- リビングをきれいにする →
- キッチンをきれいにする →
- 寝室をきれいにする →
- 子供部屋をきれいにする →

どうやって？ →

## 3. 家全体のゴミをまとめて処分




- 部屋ごとに出たゴミ袋を回収
- ゴミ置き場や処理施設へ運ぶ



## 4. 最終チェック

- 床や棚の拭き残しを確認
- 必要な備品の補充漏れを確認

## ● きれいにする

- 部屋にある不要物を片付ける 
- 散乱している小物を所定の場所に戻す
- 使わないものをゴミ箱に入れる
- 掃除機がけをする
  - 角や狭い場所も含め、隅々までホコリやゴミを吸い取る
- 拭き掃除をする
  - 家具の汚れを雑巾や洗剤で拭き取る
  - 床の汚れを雑巾や洗剤で拭き取る
- ゴミをまとめる
  - 部屋のゴミ箱の中身を回収して新しい袋に交換する


きれいにする手順はどの部屋も同じ  
→ まとめてみた

# 実は以上が、プログラミングの基本でした！

- これら3つがわかったなら、もう大丈夫！

3 (いきなりですが) オムレツを作る

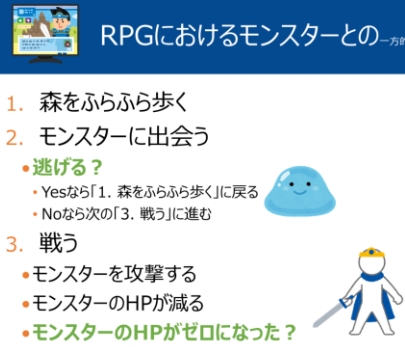
1. 材料の準備
  - 卵 (2個)
  - 塩 (ひとつまみ)
  - コショウ (ひとつまみ)
  - バター (5g)
2. 下準備
  - ボウルに卵を割り入れる
  - 塩とコショウを加える
  - 卵がなめらかになるまでよくかき混ぜる
3. メイン処理
  - 中火でフライパンを加熱する
  - バターを入れて溶かす
  - 卵液を流し込む
  - 半熟になったらオムレツを半分折りたたむ
4. 完了処理
  - お皿にオムレツをのせる
  - 完成したオムレツを盛り付けて提供する



4 RPGにおけるモンスターとの一方的戦闘

1. 森をふらふら歩く
2. モンスターに出会う
  - 逃げる？
    - Yesなら「1. 森をふらふら歩く」に戻る
    - Noなら次の「3. 戦う」に進む
3. 戦う
  - モンスターを攻撃する
  - モンスターのHPが減る
  - モンスターのHPがゼロになった？
    - Yesならモンスターを倒したので、「1. 森をふらふら歩く」に戻る
    - Noなら、「3. 戦う」に戻って攻撃を繰り返す

状況によって次のアクションが変わる



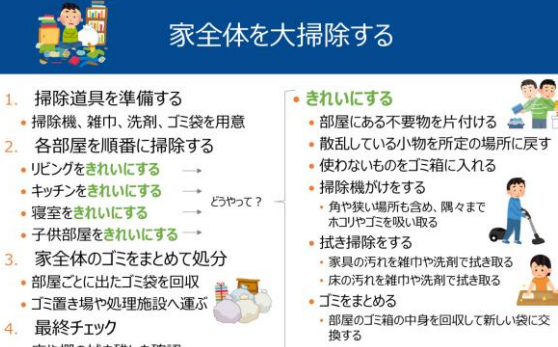
5 家全体を大掃除する

1. 掃除道具を準備する
  - 掃除機、雑巾、洗剤、ゴミ袋を用意
2. 各部屋を順番に掃除する
  - リビングをきれいにする →
  - キッチンをきれいにする → どうやって？
  - 寝室をきれいにする →
  - 子供部屋をきれいにする →
3. 家全体のゴミをまとめて処分
  - 部屋ごとにたまったゴミ袋を回収
  - ゴミ置き場や処理施設へ運ぶ
4. 最終チェック
  - 床や棚の拭き残しを確認
  - 必要な備品の補充漏れを確認

きれいにする

- 部屋にある不要物を片付ける
- 散乱している小物を所定の場所に戻す
- 使わないものをゴミ箱に入れる
- 掃除機がけをする
- 角や狭い場所も含め、隅々までホコリやゴミを吸い取る
- 拭き掃除をする
- 家具の汚れを雑巾や洗剤で拭き取る
- 床の汚れを雑巾や洗剤で拭き取る
- ゴミをまとめる
- 部屋のゴミ箱の中身を回収して新しい袋に交換する

きれいにする手順はどの部屋も同じ → まとめてみた



初期化から終了までの  
一連の流れ

変数と条件分岐と  
繰り返し

関数・メソッドによる  
構造化

「初期化？」「分岐？」「関数？」…難しそうな専門用語ですが、皆さんはこれらを理解できる能力を持っているんです

- 「無駄な苦手意識」を持たずに、自信をもって行きましょう！

# まだなんだかもやもやする…

## 結局「プログラム」や「プログラミング」って何？

- 日常用語での「プログラム」
  - 何かを**実施する手順**・順番を記したもの
  - 例 1 : 運動会のプログラム (= 競技の順番)
  - 例 2 : コンサートのプログラム (= 演奏する曲の順番)
- コンピュータの「プログラム」
  - コンピュータに何かを**実施させる手順**を記したもの
  - 例 1 : コンピュータに「『こんにちは』と画面に表示させる」プログラム (手順)
  - 例 2 : コンピュータに「 $2+3$ を計算させる」プログラム (手順)
  - 例 3 : コンピュータに「面白いゲームを実行させる」プログラム (手順)
- では「プログラミング」とは？
  - プログラムを作ること, です
  - コンピュータは自分では何もできないので, プログラムを作って手順を指示してやる必要があるのです

どちらも  
「手順」!



# プログラミング言語 Python

無料で遊べるのに凄い

# プログラミング言語とは？

- コンピュータに**何かをさせる手順**を記すための**専用の言語**
- コンピュータが正しく読み取り理解する必要がある
  - → なので、コンピュータにとってわかりやすい「言語」となっている
- コンピュータにとって「わかりやすい」とは？
  - **曖昧性がない**
  - **記述ルールが厳密に決まっている**
  - **無駄な記述がない**
  - **「（人間のよう）に 空気を読む」必要がない**

面倒そう



# 普通の言語（自然言語）とプログラミング言語

|        | 普通の言語                      | プログラミング言語                               |
|--------|----------------------------|---|
| 使う相手   | 人間                         | コンピュータ                                  |
| 曖昧さ    | 曖昧でも通じる（例：「アレ、ちょっとやっておいて」） | 曖昧さはゆるされない                              |
| 表現の自由度 | 非常に高い                      | ある程度決まった書き方が必要                          |
| 解釈の仕方  | 文脈や空気で補える                  | 書かれた通りにしか動かない                           |
| 間違えたとき | 相手が補ってくれることもある             | エラーになり止まるか、期待していない（＝間違いを真に受けた）<br>挙動をする |



融通が全然利かないのか…

# 「融通が利かない」ことにも**大きなメリット**が！

- 普通の言葉（自然言語）は**誤解やズレ**が起きやすい

- 「塩を少々」→人によって量が違う
- 「ちょっと片付けて」→どこまでやればいいのかわからない



- カチツとした言葉は**正確で安心**！

- 「塩を5g」→誰が作っても同じ味になる
- 「テーブル上の黄色い本を本棚の一番上の棚に立てて置いて」  
→やるべきことが明確で無駄や失敗がない



- プログラミング言語もカチツとしているおかげで**正確で安心**

- 一度きちっと書いてしまえば、いつも正確なので安心して使える



# プログラミング言語にはたくさんの種類がある

- 英語
- 中国語
- スペイン語
- ドイツ語
- 韓国語
- 日本語
- ...

人間が使う  
言語もいろいろ  
(性質や文法が違う)

- SQL
- C++
- HTML
- Java
- R
- Python

コンピュータが使う  
言語もいろいろ  
(性質や文法が違う)



<https://www.freepik.com>



<https://www.freepik.com>

# 色々ある中で なぜ今回pythonを選ぶのか？



## ● シンプル！

- 日常言語でも，冠詞とか，時制変化とか，丁寧語とか，名詞の性別とか，ないほうが，覚えるのは楽ですよ？
- Pythonも（昔のプログラミング言語に比べればずいぶん）単純化されている

## ● ライブラリが豊富！

- ライブラリ：すでに出来上がったプログラム．自分のプログラムでも利用可能
  - インターネット上には世界中の人が作ったPythonプログラムがたくさんあって自由に使えるものも多い！
- 日常言語でも，「例文がたくさんあれば，コピペで楽に作文できる」という感じ

## ● 用途もいろいろ！

- Webページ作るのも，AI作るのも，数値計算するのも，アプリ開発するのも，データ分析するのも，グラフィックスをつくるのも，なんでもできる

日常言語と違って，プログラミング言語には「流行」があります．  
いまはpythonが大流行中．ただし，10年後はわからない．  
ただpythonが使えるようになれば他の言語への移行も楽！

# 【準備】 Gogole Colaboratoryへ行こう！

Googleアカウントを持っていると便利

# Google Colaboratory

- 以下の web site ^Go!

<https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja>



# Google Colaboratory

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

共有 ログイン

接続

目次

- はじめに
- データサイエンス
- 機械学習
- その他のリソース
- 使用例
- + セクション

Colab へようこそ

(新規) Gemini API をお試しください

- [Generate a Gemini API key](#)
- [Talk to Gemini with the Speech-to-Text API](#)
- [Gemini API: Quickstart with Python](#)
- [Gemini API code sample](#)
- [Compare Gemini with ChatGPT](#)
- [More notebooks](#)

すでに Colab をよくご存じの場合は、この動画でインタラクティブなテーブル、実行されたコードの履歴表示、コマンドパレットについてご覧ください。

3 Cool Google Colab Features

[ ]

Colab とは

自分のGoogle  
アカウントで  
ログインすると  
作ったプログラムを  
保存できるように

この辺の画面  
デザインは  
頻繁に変わるので  
あまり気にしない

# 練習：ノートブック（=プログラムを書き込むところ）をつくろう！

①「ファイル」をクリック

②「新しいノートブック」をクリック



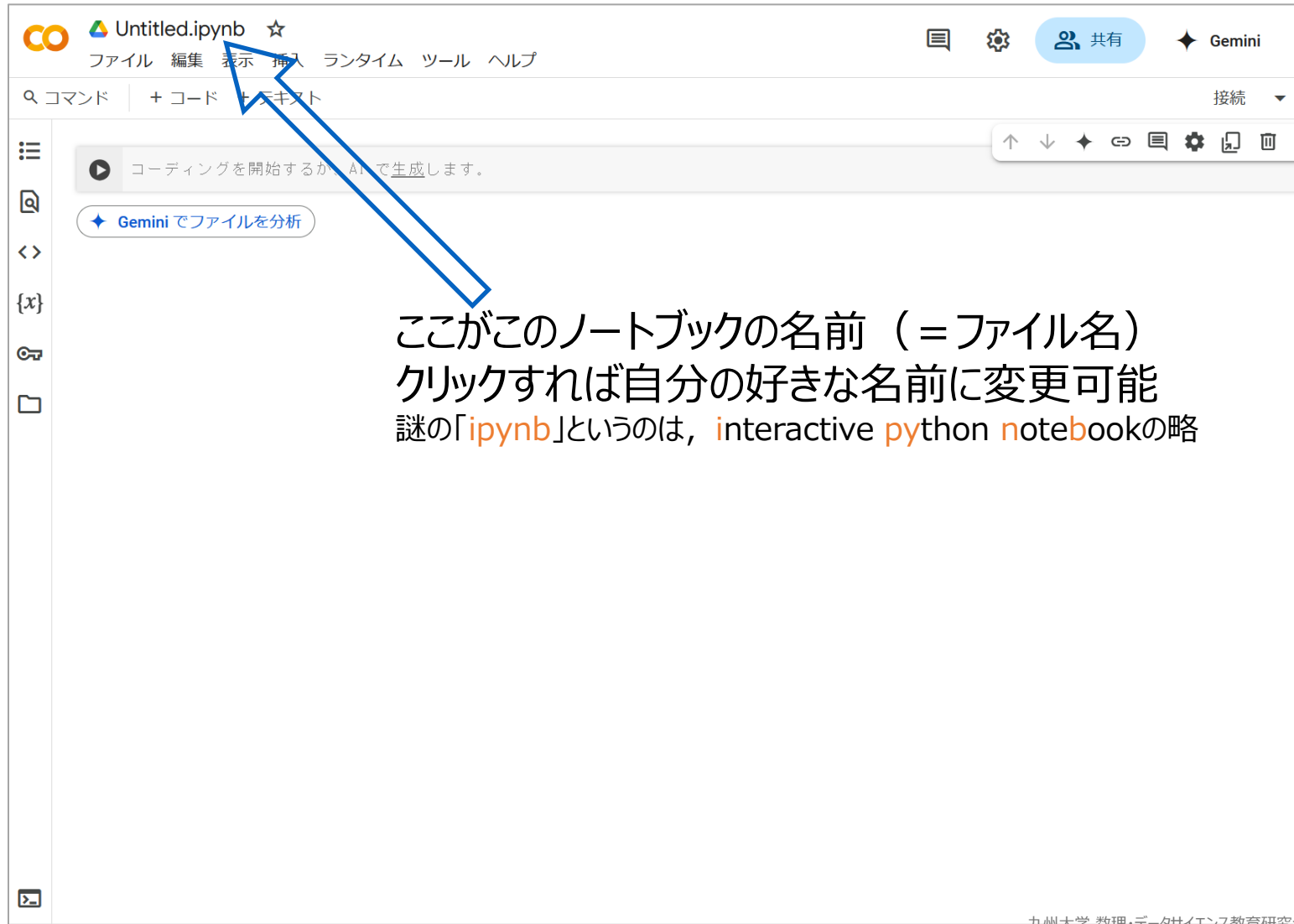
The screenshot shows the Google Colaboratory web interface. The 'File' menu is open, and the option 'ドライブの新しいノートブック' (New notebook in Drive) is highlighted with a blue box. A hand cursor is pointing at this option. The interface includes a top navigation bar with 'Colaboratory へようこそ', '共有' (Share), and 'Gemini' options. The main content area displays a welcome message and links to various Gemini API resources. A video player is visible at the bottom, titled '3 Cool Google Colab Features'.

# 続き：こんなのが出てきたら準備完了

The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** "Untitled.ipynb" with a star icon, and a menu bar with "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", and "ヘルプ".
- Right Header:** "共有" (Share) button and "Gemini" logo.
- Command Bar:** "コマンド" search, "+ コード", "+ テキスト", and "接続" dropdown.
- Left Sidebar:** Navigation icons for home, search, expand/collapse, math symbols, and file explorer.
- Main Area:** A prompt box with a play button icon and the text "コーディングを開始するか、AI で生成します。" (Start coding or generate with AI?). Below it is a button labeled "Gemini でファイルを分析" (Analyze file with Gemini).
- Bottom Right:** A terminal icon.

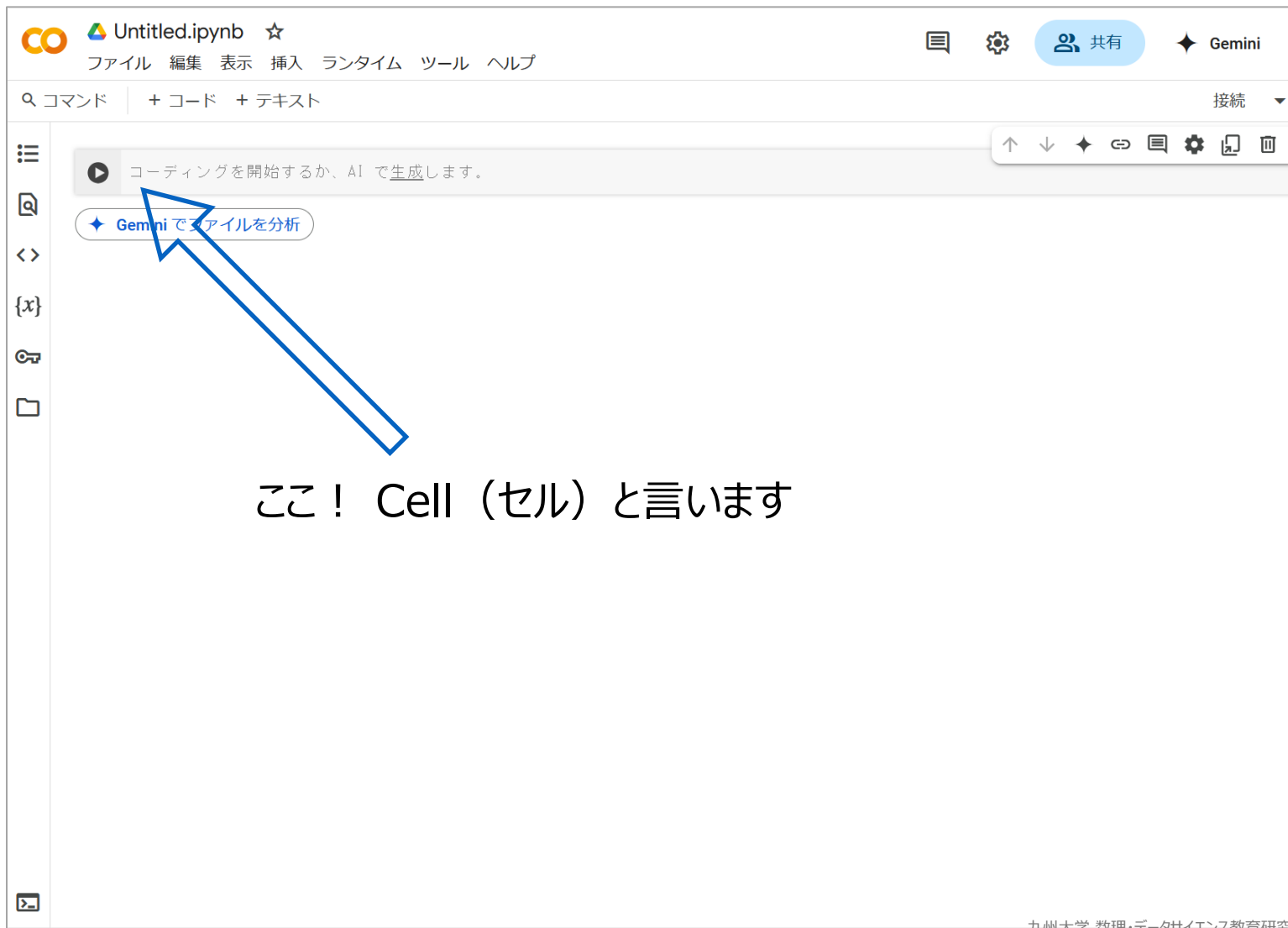
# 続き：こんなのが出てきたら準備完了



ここがこのノートブックの名前（=ファイル名）  
クリックすれば自分の好きな名前に変更可能  
謎の「ipynb」というのは、interactive python notebookの略

九州大学 数理・データサイエンス教育研究センター 2025年4月版

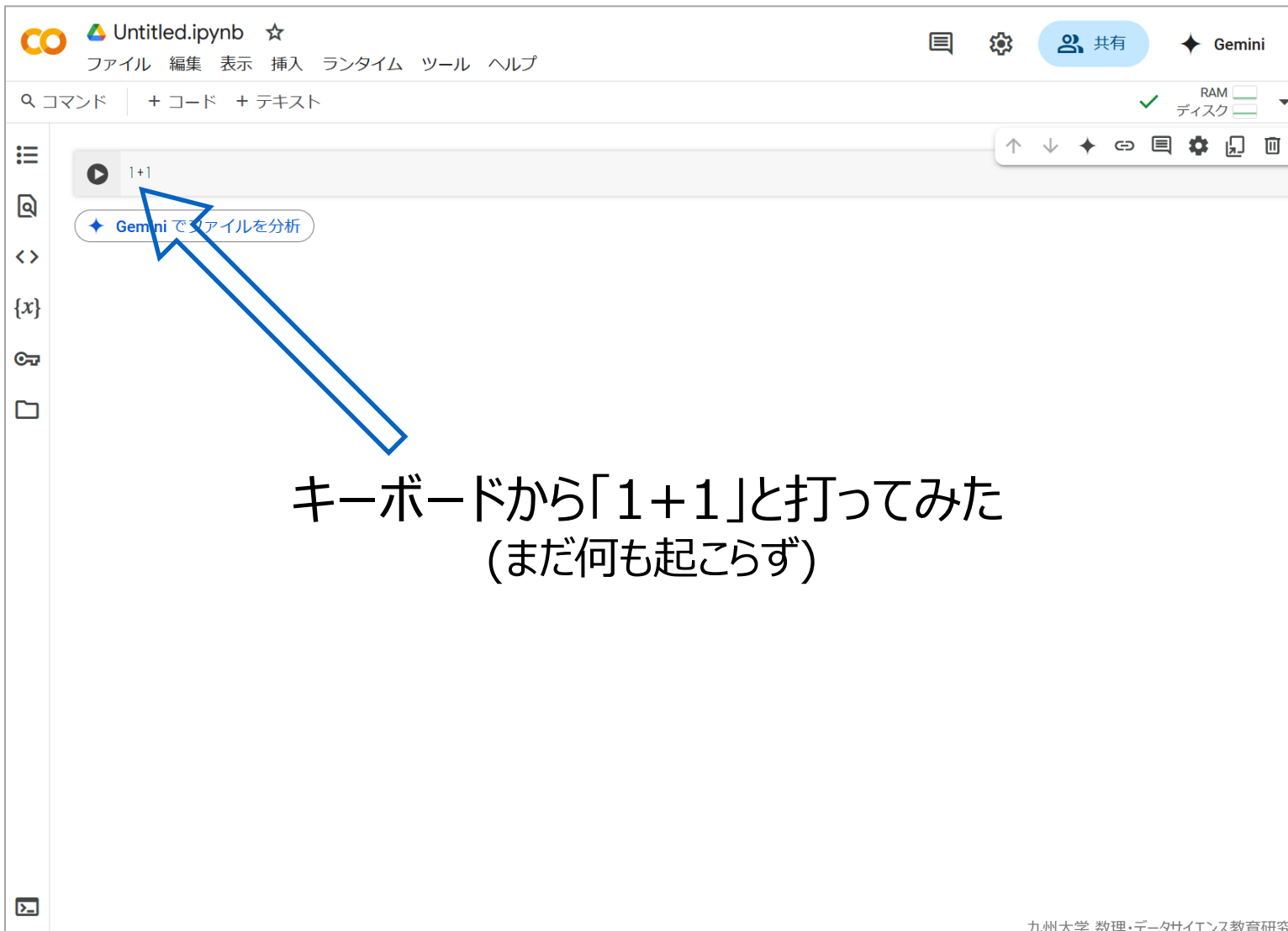
# 続き：どこにプログラムを書くのか？



The screenshot shows a Jupyter Notebook interface. At the top, the title bar reads "Untitled.ipynb" with a star icon. Below it are menu items: "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", "ヘルプ". On the right side of the title bar, there are icons for "共有" (Share) and "Gemini". Below the title bar, there is a search bar with "コマンド" and buttons for "+ コード" and "+ テキスト". On the right side of this bar is a "接続" (Connect) dropdown menu. The main area of the notebook shows a cell with a play button icon and the text "コーディングを開始するか、AI で生成します。". Below this text is a button that says "Gemini でファイルを分析". A blue arrow points from the text "ここ！ Cell (セル) といいます" below the screenshot to the button.

ここ！ Cell (セル) といいます

# 続き：そこに「1+1」と打ってみよう



Untitled.ipynb ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

RAM  ディスク

1+1

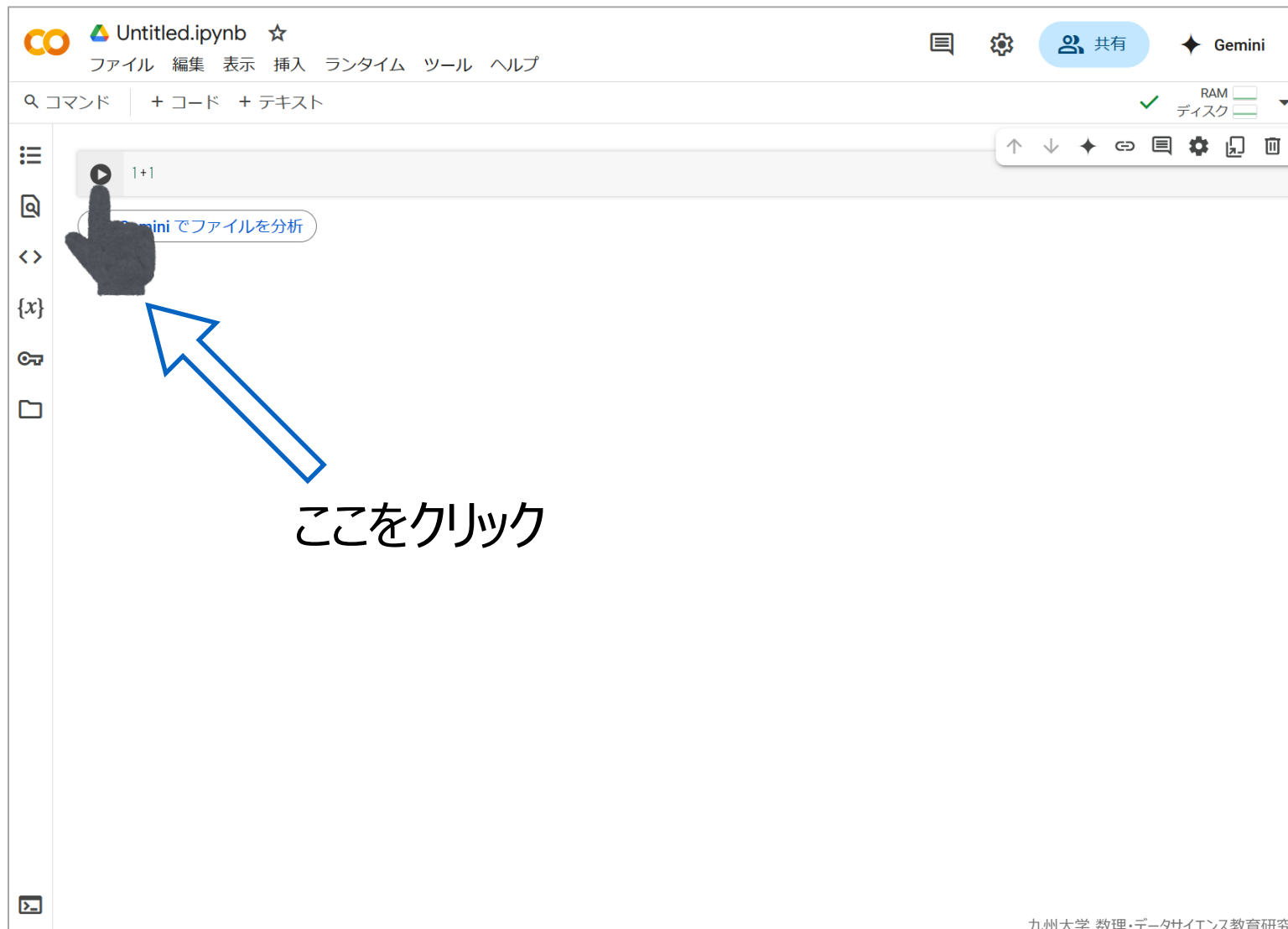
◆ Gemini でファイルを分析

↑ ↓ ◆ 🔗 🗨 ⚙ 📄 🗑

キーボードから「1+1」と打ってみた  
(まだ何も起こらず)

九州大学 数理・データサイエンス教育研究センター 2025年4月版

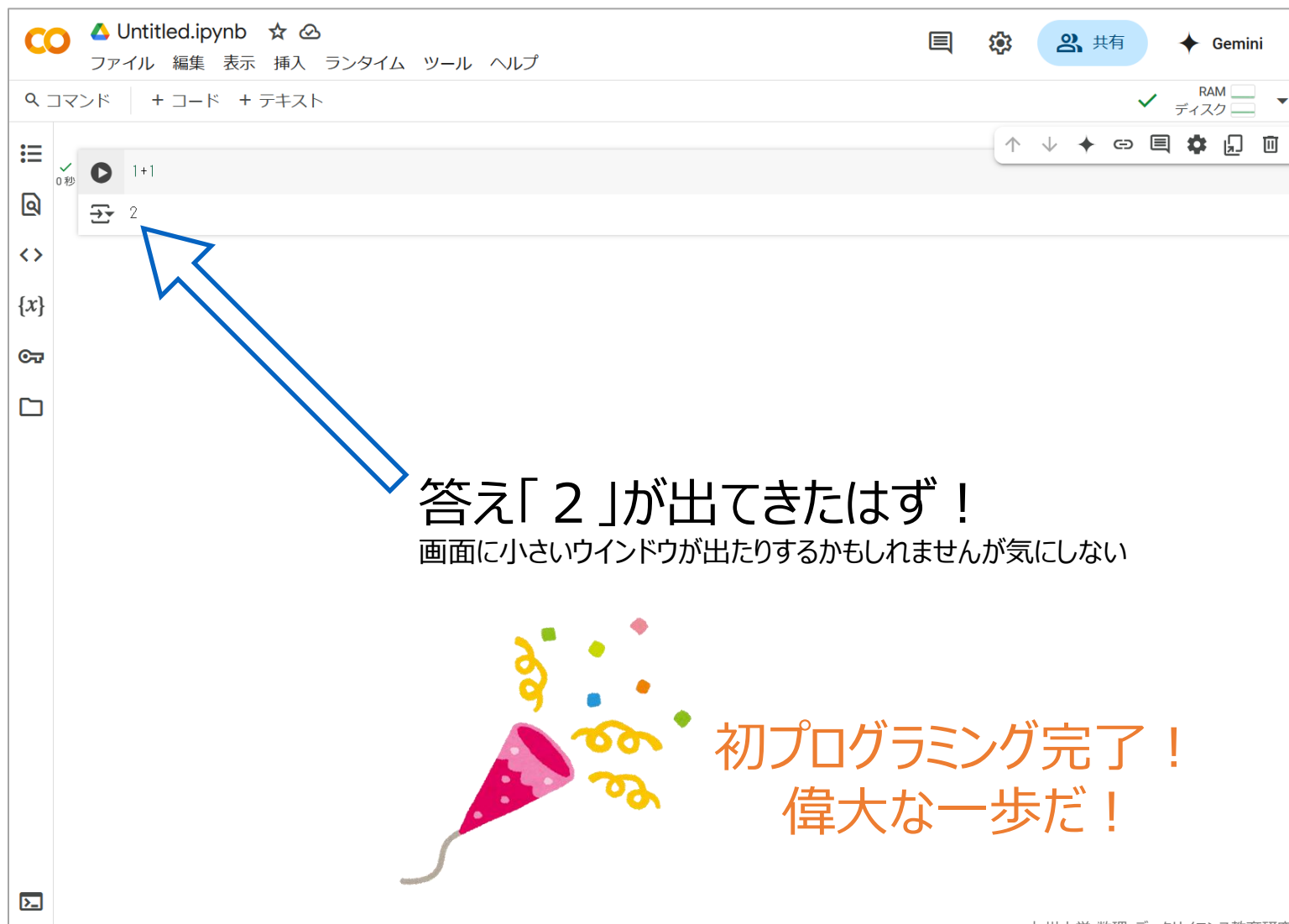
# 続き：横にある「矢印 」をクリックする



The screenshot shows a Jupyter Notebook interface. At the top, the title bar reads "Untitled.ipynb" with a star icon. Below it are menu items: "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", "ヘルプ". On the right side of the title bar, there are icons for "共有" (Share) and "Gemini". Below the title bar, there is a search bar with "コマンド" and buttons for "+ コード" and "+ テキスト". On the right side of the search bar, there are status indicators for "RAM" and "ディスク". The main area of the notebook shows a code cell with a play button icon and the text "1+1". A blue arrow points to the play button icon, and the text "ここをクリック" (Click here) is written below the arrow. The left sidebar contains icons for "ファイル", "変数", "セル", "ノートブック", and "ヘルプ".

ここをクリック

# 続き：すると答えが出てくる！



Untitled.ipynb ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

RAM ディスク

1+1

2

答え「2」が出てきたはず！  
画面に小さいウィンドウが出たりするかもしれませんが気にしない

初プログラミング完了！  
偉大な一歩だ！

九州大学 数理・データサイエンス教育研究センター 2025年4月版

# エラーを怖がらないように！



- プログラミングにはエラーがつきもの！
- 例えば, "1+1"の後に" ="まで付けてしまうと…

マークが赤くなる

赤い波線でエラーを起こしている箇所を指示

どんなエラーかを説明

ここではsyntax errorと表示.  
これは,「文法 (プログラムの書き方ルール) に関して間違っている!」ということ

次のステップ: [エラーの修正](#)

- どんなエラーがでてでも「爆発」はしません！
  - エラーを怖がらずにいろいろ試してみましょう！それがプログラムの自由さだ！



# 日本語入力モードにも 気を付けよう！



- “1+1”を日本語入力モードで” 1 + 1 ”としてしまうと…

マークが赤くなる

赤い波線でエラーを起こしている箇所を指示

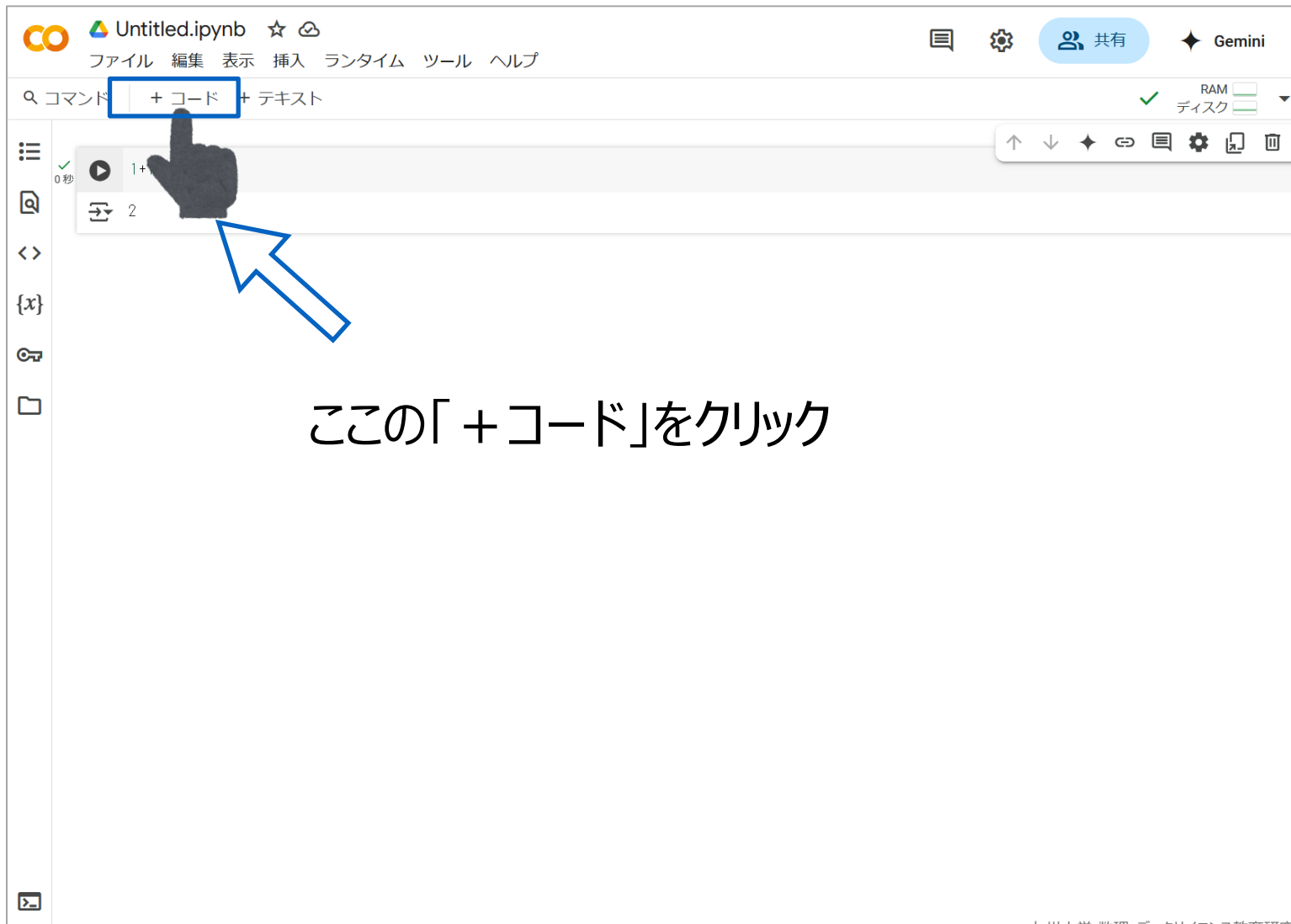
どんなエラーかを説明

ここではinvalid characterと表示。これは、「使えない文字を使っている」ということ

次のステップ: [エラーの修正](#)

- 日本語入力は（ほぼ）使わない，ぐらいと覚えておこう
  - 後述するprintやコメントでは使ってもよい

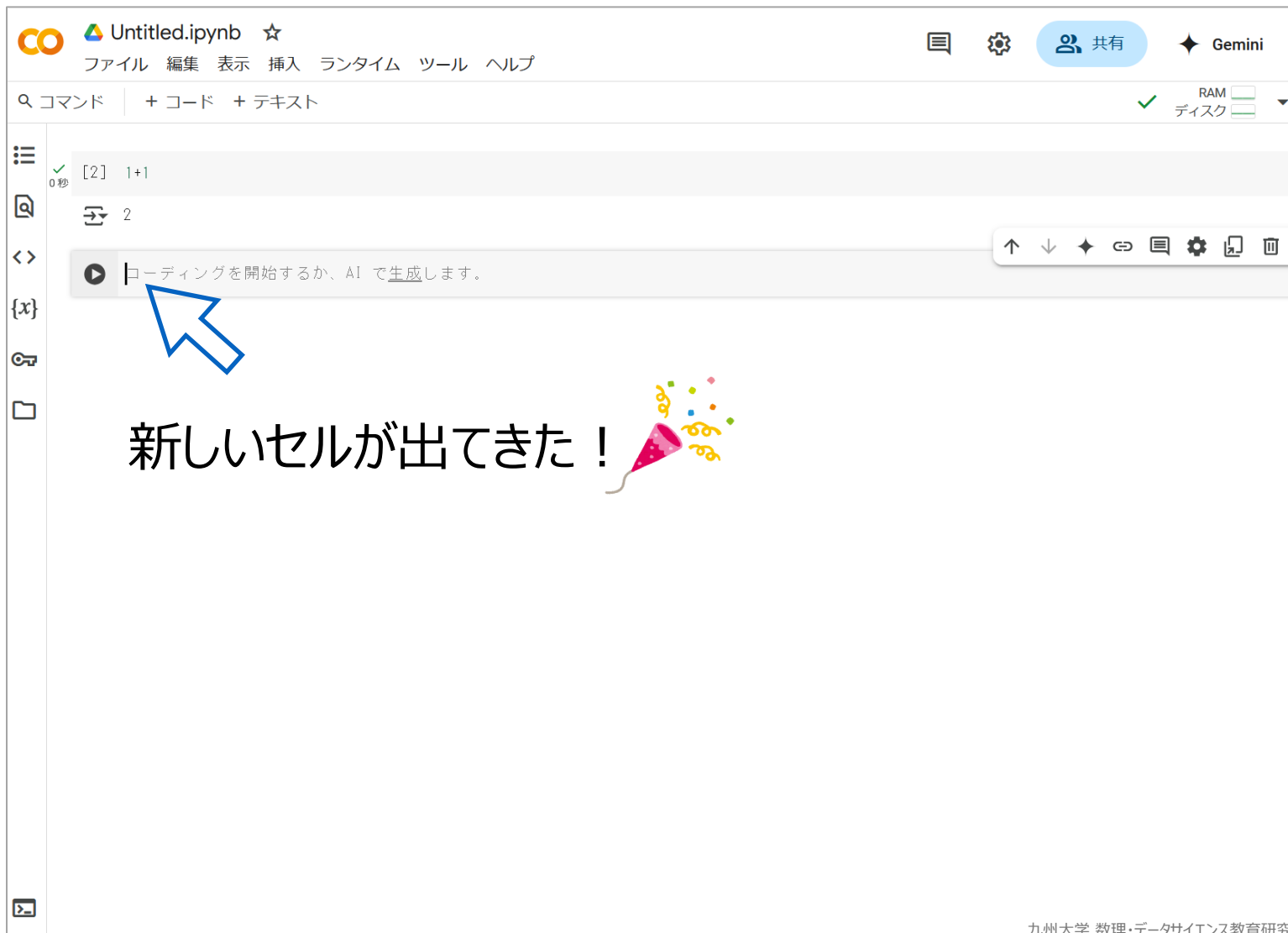
# 練習：もっとやりたい… そんなときは？



The screenshot shows the Jupyter Notebook interface for an 'Untitled.ipynb' file. The top navigation bar includes 'ファイル 編集 表示 挿入 ランタイム ツール ヘルプ' and '共有 Gemini'. The command palette is open, showing 'コマンド + コード + テキスト'. A blue box highlights the '+ コード' button, with a hand cursor pointing to it and a blue arrow pointing to the text below. The notebook content area shows two code cells, the first with a play icon and '1+' and the second with a refresh icon and '2'. The right sidebar shows RAM and Disk usage indicators.


この「+コード」をクリック

# 練習：もっとやりたい… そんなときは？

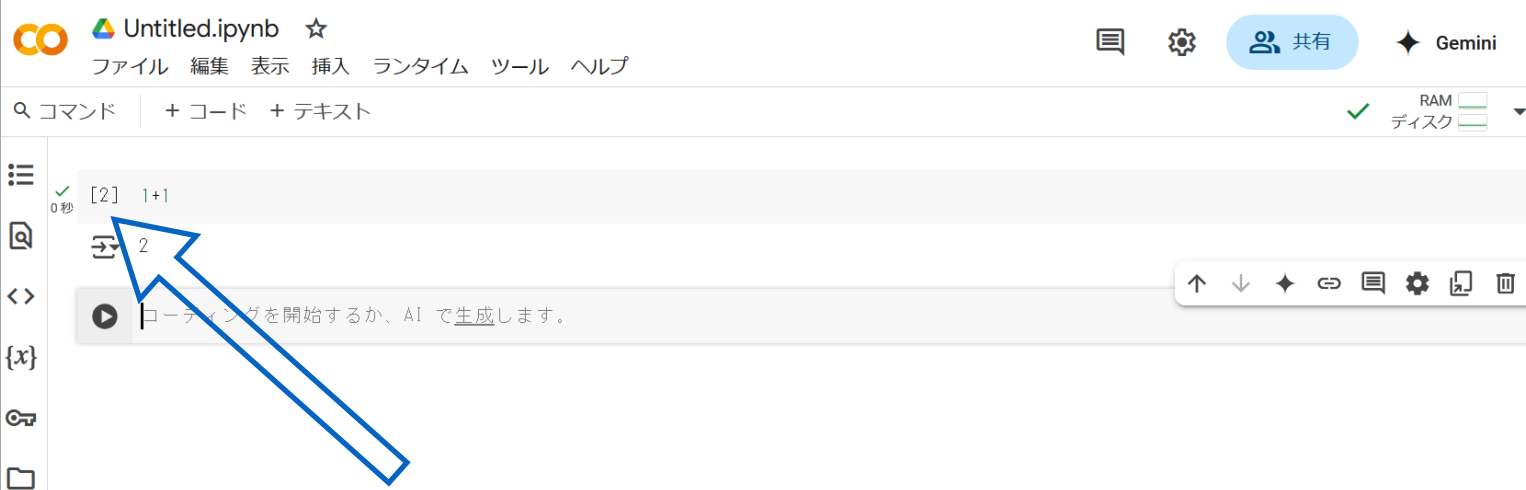


The screenshot shows a Jupyter Notebook interface with the following elements:

- Top bar: "Untitled.ipynb" with a star icon, and navigation menus for "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", and "ヘルプ".
- Right side of top bar: "共有" (Share) button and "Gemini" logo.
- Search bar: "コマンド" with options for "+ コード" and "+ テキスト".
- RAM and Disk usage indicators: "RAM" and "ディスク" with progress bars.
- Code cells:
  - Cell 1: "[2] 1+1" with a green checkmark and "0秒" (0 seconds).
  - Cell 2: "2" with a refresh icon.
  - Cell 3: A prompt "ローディングを開始するか、AI で生成します。" (Start loading or generate with AI?). A blue arrow points to the play button on the left of this cell.
- Bottom left sidebar: Navigation icons for home, search, expand/collapse, code, text, and file explorer.

新しいセルが出てきた！ 

# 練習：もっとやりたい… そんなときは？

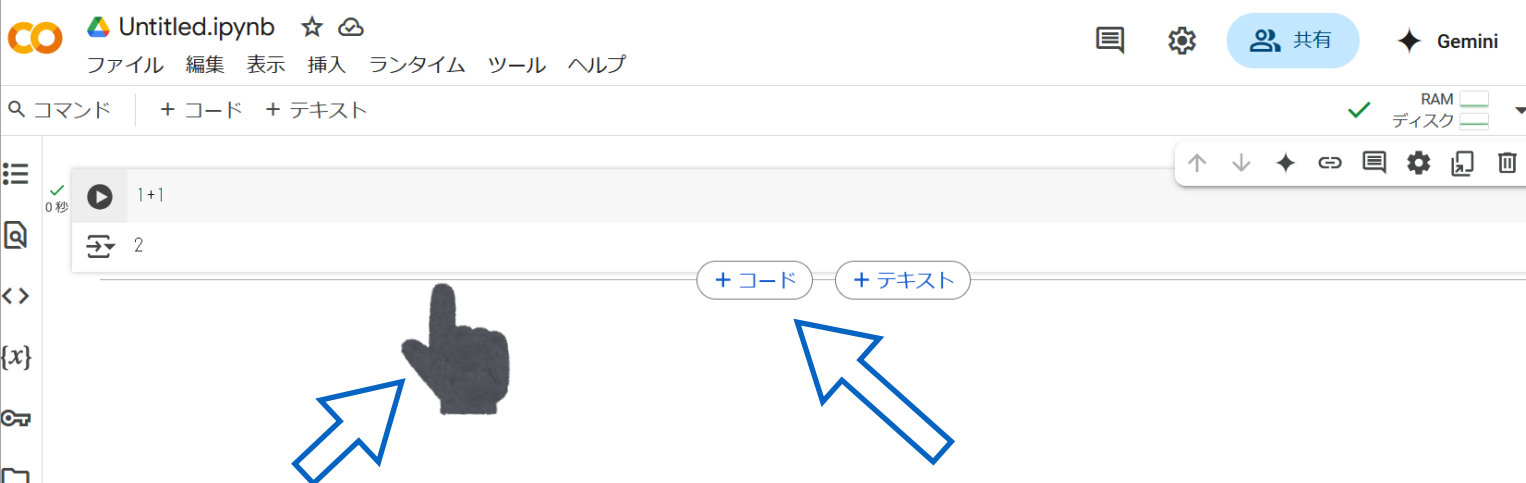


The screenshot shows a Jupyter Notebook interface. At the top, it says "Untitled.ipynb" and "Gemini". Below that, there are menu options: "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", "ヘルプ". A search bar contains "コマンド" and "コード", "テキスト". On the right, there are icons for "共有" and "RAM", "ディスク". The main area shows a code cell with the code "1+1" and an output cell with the value "2". A blue arrow points to the execution time "[2] 0秒" in the output cell. Below the screenshot, there is a text box with the following text:

この[ ]内の数字はとりあえず気にしない  
(ノートブックを開いてから何回目に実行されたセルかを表します)

# 練習：もっとやりたい… そんなときは？

## 別の方法



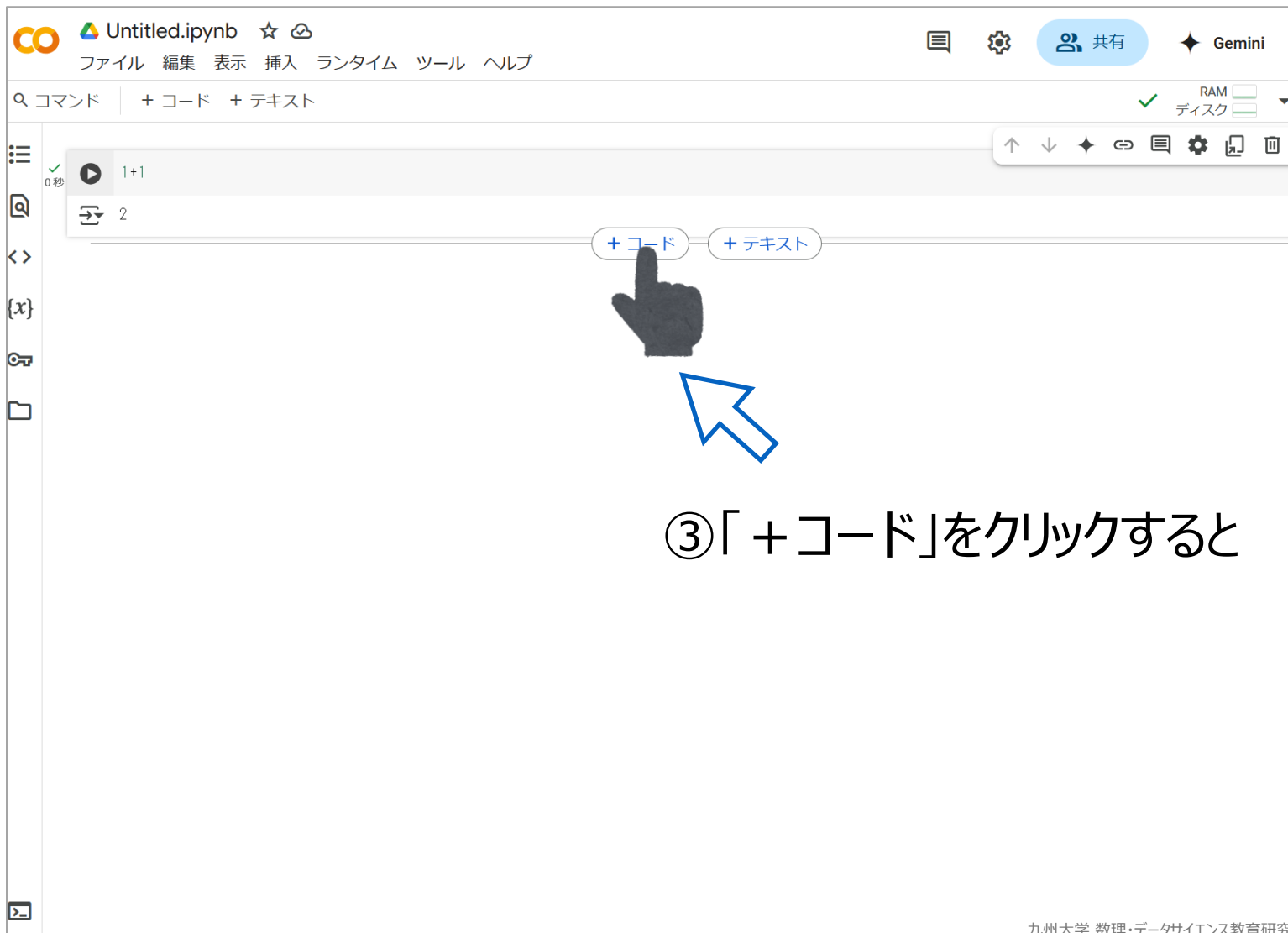
①最後のセルの下ぐらいに  
マウ斯卡ーソルを持ってくると…

②「+コード」というのが  
出てくる

九洲大学 数理・データサイエンス教育研究センター 2025年4月版

# 練習：もっとやりたい… そんなときは？

## 別の方法

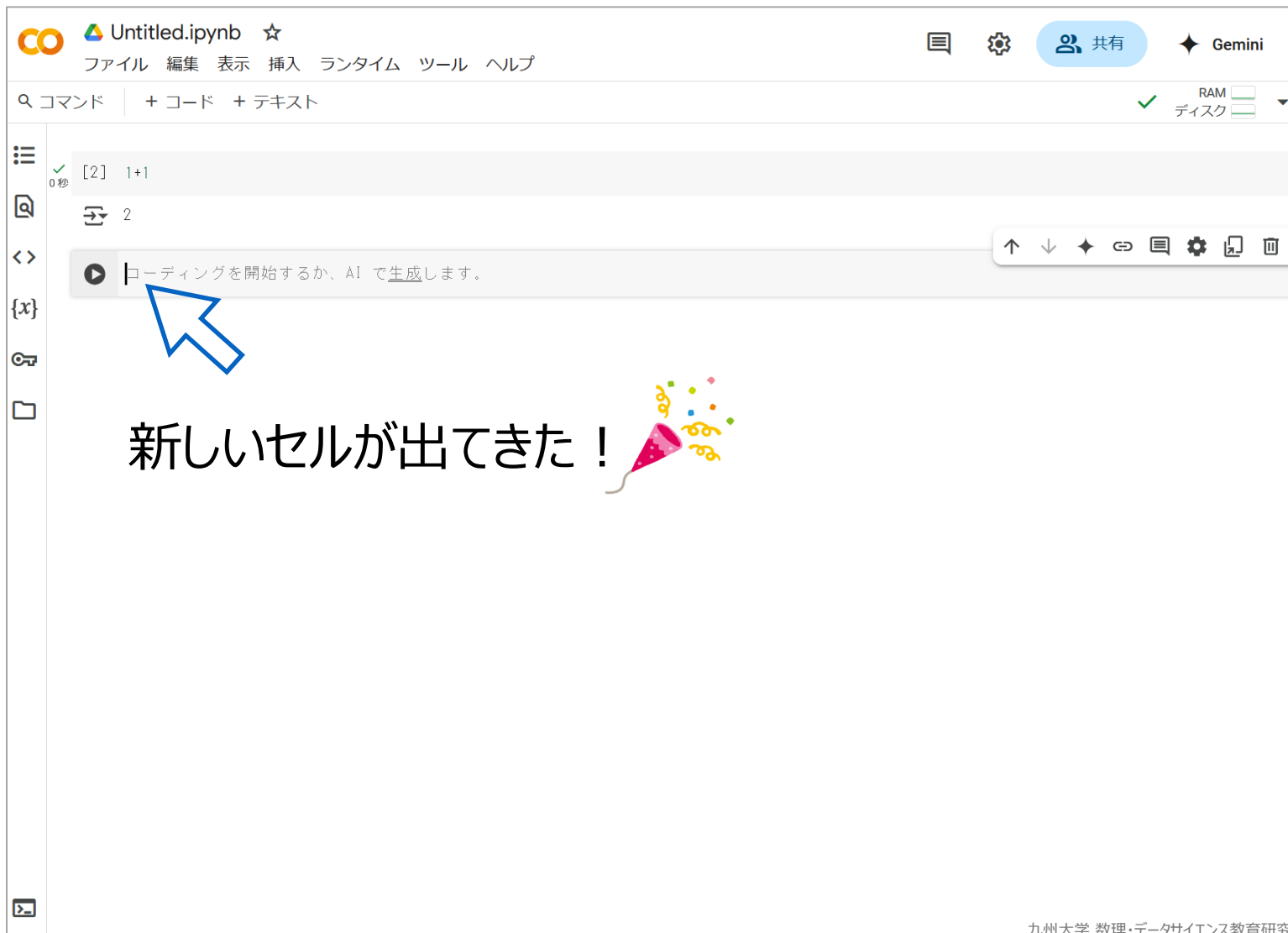


The screenshot shows a Jupyter Notebook interface with the title 'Untitled.ipynb'. The top menu bar includes 'ファイル', '編集', '表示', '挿入', 'ランタイム', 'ツール', and 'ヘルプ'. The top right corner shows '共有' and 'Gemini'. The left sidebar contains icons for home, back, forward, search, and file management. The main area displays a code cell with the output '1+1' and '2'. Below the code cell, there are two buttons: '+コード' and '+テキスト'. A hand icon is pointing to the '+コード' button, and a blue arrow points to it from below. The text '③「+コード」をクリックすると' is positioned below the buttons.

③「+コード」をクリックすると

# 練習：もっとやりたい… そんなときは？

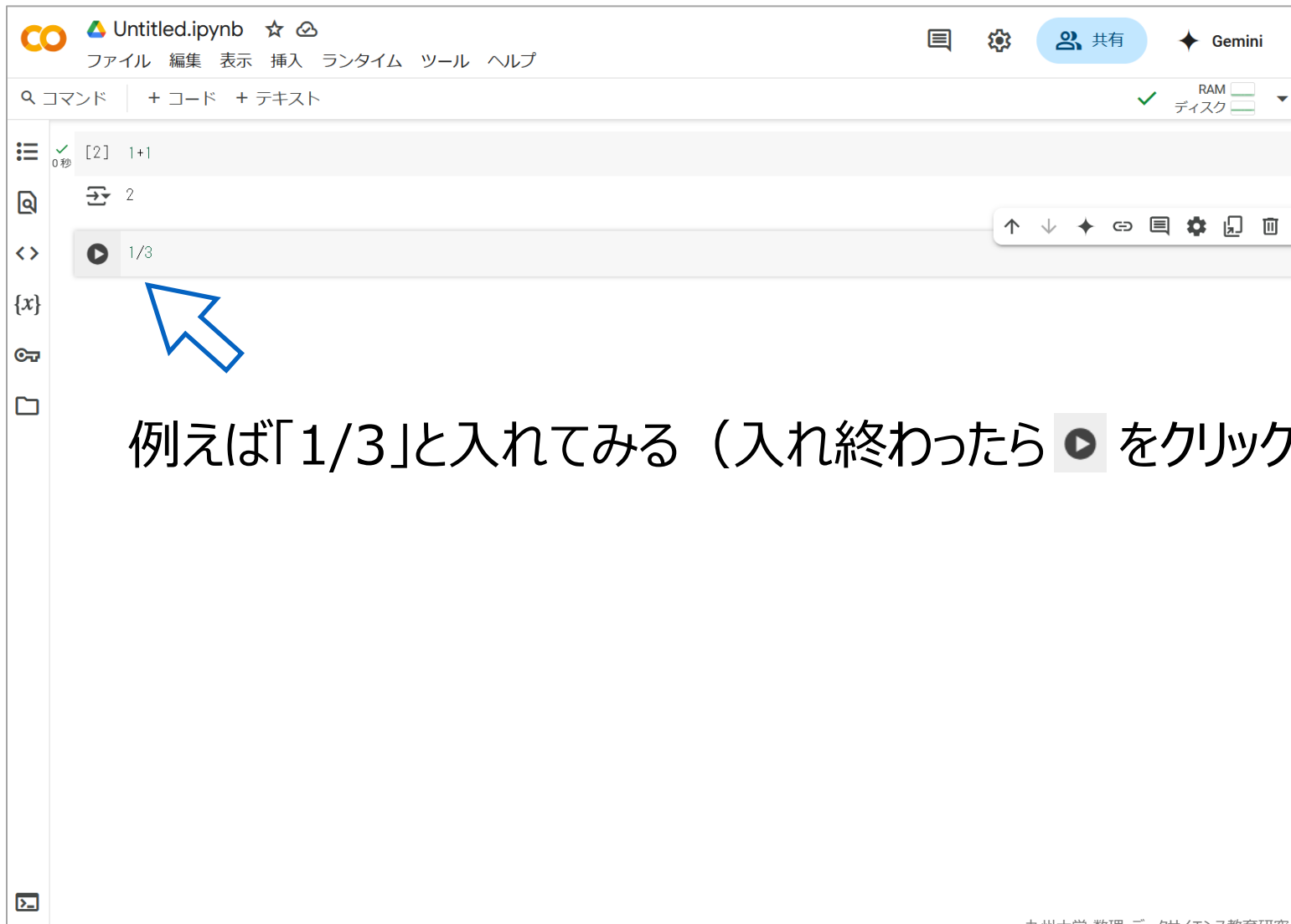
## 別の方法




The screenshot shows a Jupyter Notebook interface with the following elements:

- Top bar: "Untitled.ipynb" with a star icon, and navigation links: "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", "ヘルプ".
- Right side of top bar: "共有" (Share) button and "Gemini" logo.
- Search bar: "コマンド" with "+ コード" and "+ テキスト" options.
- RAM and Disk usage indicators: "RAM" and "ディスク" with green bars and a checkmark.
- Cell execution history: A cell with code "[2] 1+1" and output "0秒", and another cell with code "2".
- Cell toolbar: "↑", "↓", "◆", "↶", "🗨️", "⚙️", "📄", "🗑️".
- Cell content: A play button icon followed by the text "ローディングを開始するか、AI で生成します。". A blue arrow points to the play button.
- Text overlay: "新しいセルが出てきた！" (A new cell has appeared!) with a party popper icon.

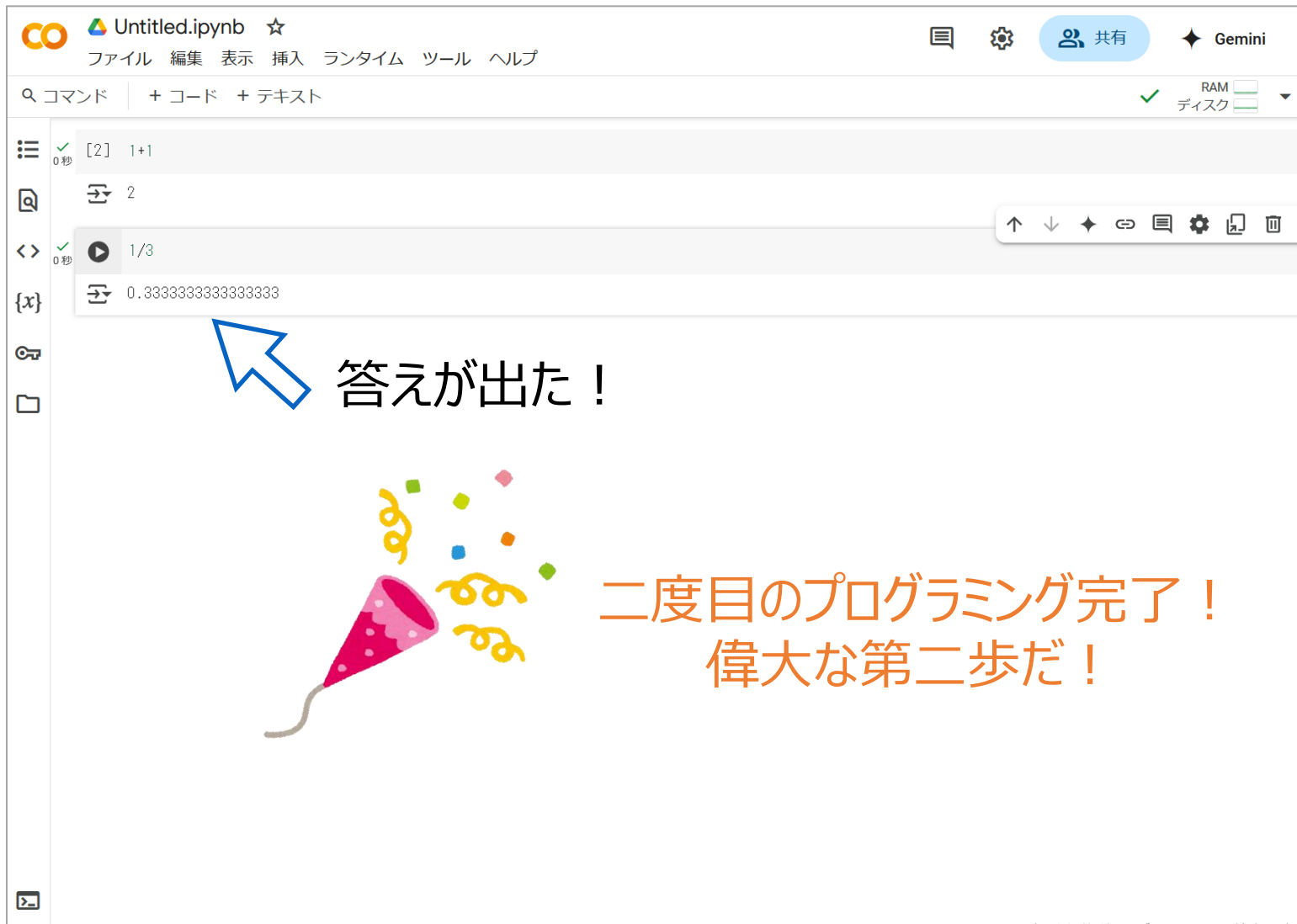
# 練習：新しい計算式を入れてみる



The screenshot shows a Jupyter Notebook interface. At the top, the title bar reads "Untitled.ipynb" with a star icon and a share icon. Below the title bar, there are menu options: "ファイル", "編集", "表示", "挿入", "ランタイム", "ツール", and "ヘルプ". On the right side of the title bar, there are icons for chat, settings, a "共有" (Share) button, and "Gemini". Below the title bar, there is a search bar labeled "コマンド" and buttons for "+ コード" and "+ テキスト". On the right side of this bar, there are status indicators for "RAM" and "ディスク" (Disk) with green checkmarks. The main area of the notebook shows a list of cells. The first cell is a code cell containing "1+1" with a green checkmark and "0秒" (0 seconds) next to it. The second cell is a code cell containing "2" with a refresh icon. The third cell is a code cell containing "1/3" with a play button icon. A blue arrow points to the play button icon. To the right of the cells, there are navigation icons: up, down, search, refresh, settings, and trash. At the bottom left, there is a small icon of a terminal window.

例えば「1/3」と入れてみる（入れ終わったら  をクリック）

# 練習：新しい計算式を入れてみる



Untitled.ipynb ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

共有 Gemini

RAM ディスク

コマンド + コード + テキスト

[2] 1+1

2

1/3

0.3333333333333333

答えが出た！

二度目のプログラミング完了！  
偉大な第二歩だ！

九州大学 数理・データサイエンス教育研究センター 2025年4月版

# んん？ プログラム = 電卓なの？



The screenshot shows a Jupyter Notebook titled 'Untitled.ipynb'. The interface includes a top bar with '共有' (Share) and 'Gemini' buttons, and a left sidebar with icons for file operations. The main area contains a code cell with the following content:

```
[2] 1+1  
2  
1/3  
0.3333333333333333
```

Handwritten annotations in Japanese are present:

- A callout box on the left points to the code cell, containing the text:   
1+1とか1/3とかなら  
電卓でも計算できるぞ
- A callout box on the right contains the text:   
まあ確かに、  
「コンピューター(computer)=  
計算(compute)する機械」  
だからなあ...

# んん？ プログラム = 電卓なの？



人工知能

画像作成

検索

ゲーム

## そんなことはない！

## めちゃめちゃいろんなことができます！

あなたのスマホの上で動いているものは全部プログラム

メッセージ  
送受信

自動運転

## 「無限に遊べる」

## プログラムの世界へようこそ！


天気予報

ヘルスケア

# 文字を表示してみよう

記念すべき最初のプログラム


# 文字を表示してみる： print

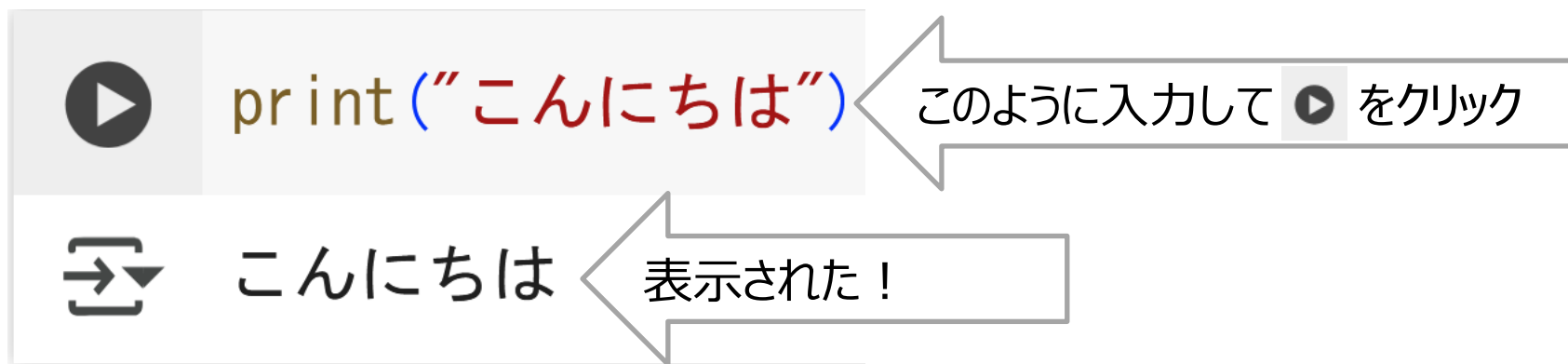
- `print("Hello")` と入力
  - 引用符("...")が必要！
-  をクリック



- そのまま表示されるだけなので何が嬉しいのかわかりにくいですが、あなたの「Helloと表示せよ！」という命令にコンピュータが従ったんです！

# 文字を表示してみる： print

- `print("こんにちは")` と入力
-  をクリック



- といふわけで、日本語も表示できる

## 練習：色々な文字列を表示させてみよう



```
print("ピザと10回言ってみて")
```

こう入力して  をクリック



```
ピザと10回言ってみて
```

表示された！

- 上の例で「ピザピザピザ…」と出るわけじゃないです
- あくまで引用符("...")の中をそのまま表示するだけ