

# 講義・演習環境アプリケーションハンズオン

## — 1. nbgraderを用いた課題の自動採点 —

2025年9月2日

# ご注意ください

- この資料はテスト環境にて作成しています。
- 実際のハンズオンセミナーの環境とはコース名、コンテンツ名、表示内容が多少異なります。

# 最初にMoodleに関する簡単な説明 (1/2)

- MoodleのWebサイトはコースの集まり。
- コースは各々の科目・授業・行事のためのWebページ（または一連のWebページ）。

Moodle (Webサイトに相当)

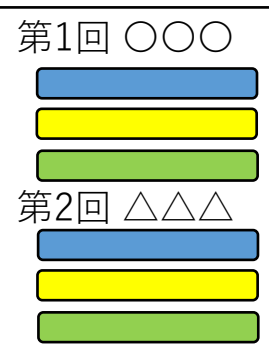


多数のWebページ（コース）の集まり

授業科目のコース  
(Webページに相当)

授業のコースを訪問して資料を閲覧

受験結果や評点等のページでは自分の情報のみが表示



- コースのデザイン
- 講義資料の掲載
- 小テストの作成
- 課題の作成と採点
- 活動履歴の把握

# 最初にMoodleに関する簡単な説明 (2/2)

- Moodleのコース内にはさまざまなコンテンツを貼り付け可能。
- 活動
  - 評点が付与されるコンテンツ。
  - 小テスト、課題（レポート提出）、アンケート（評点を付与可能）など。
  - 今回のセミナーでは「外部ツール」という活動を使用。
- リソース
  - 評点が付与されないコンテンツ。
  - ファイル、フォルダ、ページ、リンクなど。

# 自動採点の演習内容

- 学生としての活動。
  - 学生としてログインして、共通のMoodleコースにアクセス。
  - 課題の取得、プログラムの記入・実行、課題の提出。
  - 多人数に対する一括の採点を見学。
  - 採点結果の取得・閲覧。
  - いったんログアウト。
- 教師としての活動。
  - 教師としてログインして、各自のMoodleコースにアクセス。
  - 課題の作成、配布。
  - セミナーのスタッフによる課題の提出を待機。
  - 提出物の回収・採点、採点結果の公開。

※プライベートウィンドウや複数のブラウザを使用して、教師と学生の両方のアカウントでずっとログインしていても構いませんが、セミナーでは1つのブラウザとウィンドウで操作する手順を説明します。

# 課題の提出（学生）

みなさん同じ科目の受講生として課題ファイルを取得・提出してもらいます。

# 学生としてMoodleにログイン

- 以下のURLにアクセス。
  - <https://mcj-handson-cwh.vcp-handson.org/moodle/login/index.php>
- 割り当てられたユーザ名とパスワードでログイン。
  - 今回は学生用のアカウントを使用してログイン。

割り当てられた学生用  
アカウントでログイン

user110s

.....

ログイン

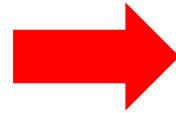
[パスワードを忘れましたか?](#)

いくつかのコースにはゲストアクセスで  
できます

ゲストとしてアクセスする

日本語 (ja) ▾

クッキー通知



New Site

Home

ダッシュボード

マイコース



SU ▾

編集モード



クリックしてマイコースに移動

こんにちは student さん! 🙌

タイムライン

次の7日 ▾

日付で並べ替える ▾

活動タイプまたは活動名で検索する



アクションを必要とする活動はありません。

# コースに移動してMCJ-CloudHubを起動

- マイコースに自分のユーザ番号に対応したコース) があるので、そのコースアクセス (画面例とは異なるかも)。
- コース内に設置された外部ツールをクリック。

New Site Home ダッシュボード マイコース

## マイコース

コース概要

すべて 検索 コース名で並べ替える カード

course\_110  
カテゴリ1

tutorial\_course  
カテゴリ1

クリックしてコースに移動

一般

アナウンスメント

MCJ-CloudHub

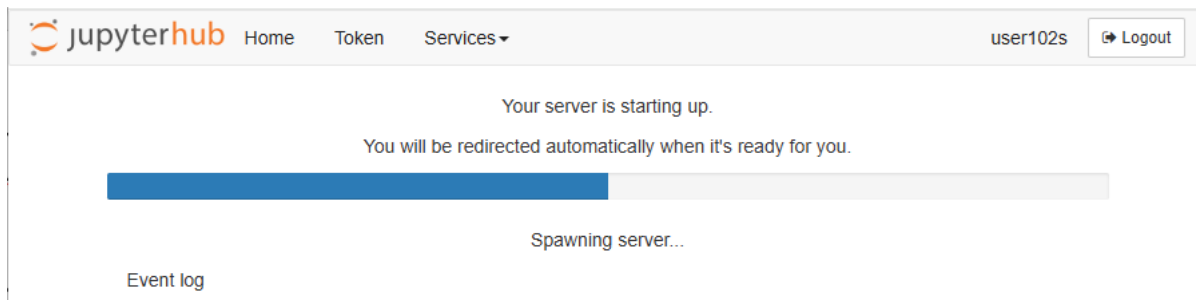
クリックしてMCJ-CloudHubにアクセス

新しいセクション

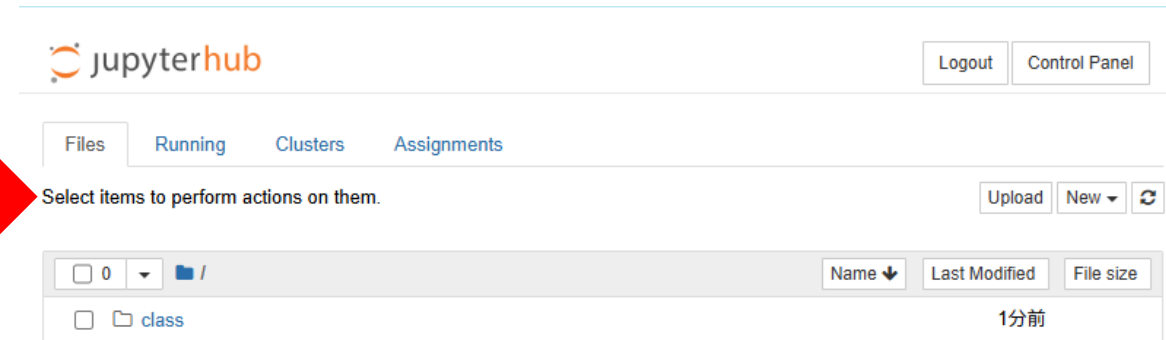


# Jupyter環境の起動

- MCJ-CloudHubにアクセスすると、各自のJupyter環境（プログラムの開発・実行環境）が起動。
  - 起動が完了するまで待機。



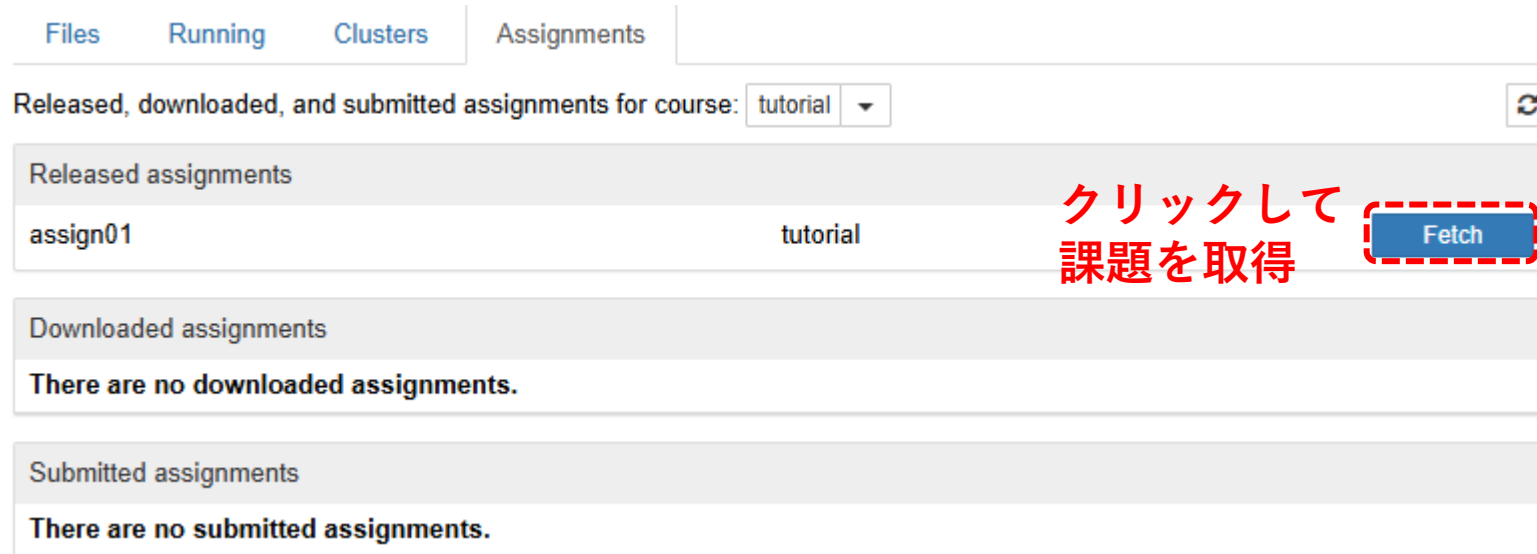
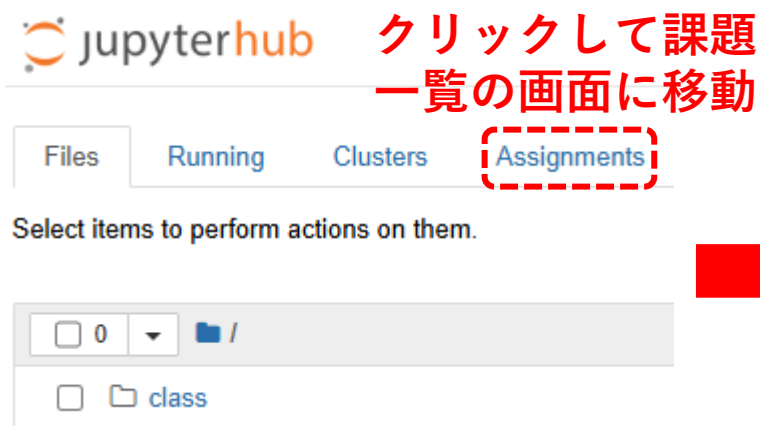
**Jupyter環境が起動するまで待機**



**起動が完了するとファイルやフォルダの一覧が表示**

# 課題の取得 (1/2)

- 画面上部の「Assignments」をクリックして課題一覧画面に移動。
- 「assign01」が公開されているので、「Fetch」をクリック。



## 課題の取得 (2/2)

- 取得済み課題の中に「assign01」が表示されるので、課題名をクリック。

Files Running Clusters Assignments

Released, downloaded, and submitted assignments for course: tutorial

Released assignments

There are no assignments to fetch.

Downloaded assignments

assign01 ▶	tutorial	Submit
------------	----------	--------

課題名をクリックして、課題に含まれるファイルを表示

- 課題に含まれるファイルが表示されるので、「01-auto」をクリック。

Released assignments

There are no assignments to fetch.

Downloaded assignments

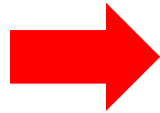
assign01 ▼	tutorial	Submit
01-auto		Validate

ファイル名をクリックして編集を開始

# 解答の作成 (1/4)

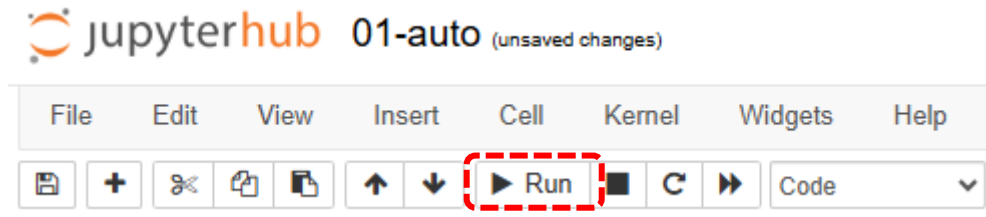
- プログラムを記入する課題の場合、「raise NotImplementedError()」を削除した後に解答を記入。

```
In [ ]: def square(n):  
        # YOUR CODE HERE  
        raise NotImplementedError()
```



```
In [ ]: def square(n):  
        # YOUR CODE HERE  
        return n*n
```

- プログラムを記入できたらセルを実行。



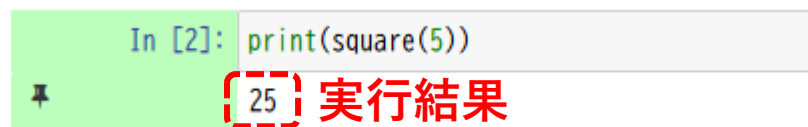
クリックしてセルを実行

## 解答の作成 (2/4)

- プログラムを実行してエラーが表示されなければ、確認用のセルをクリックした後、そのセルを実行。

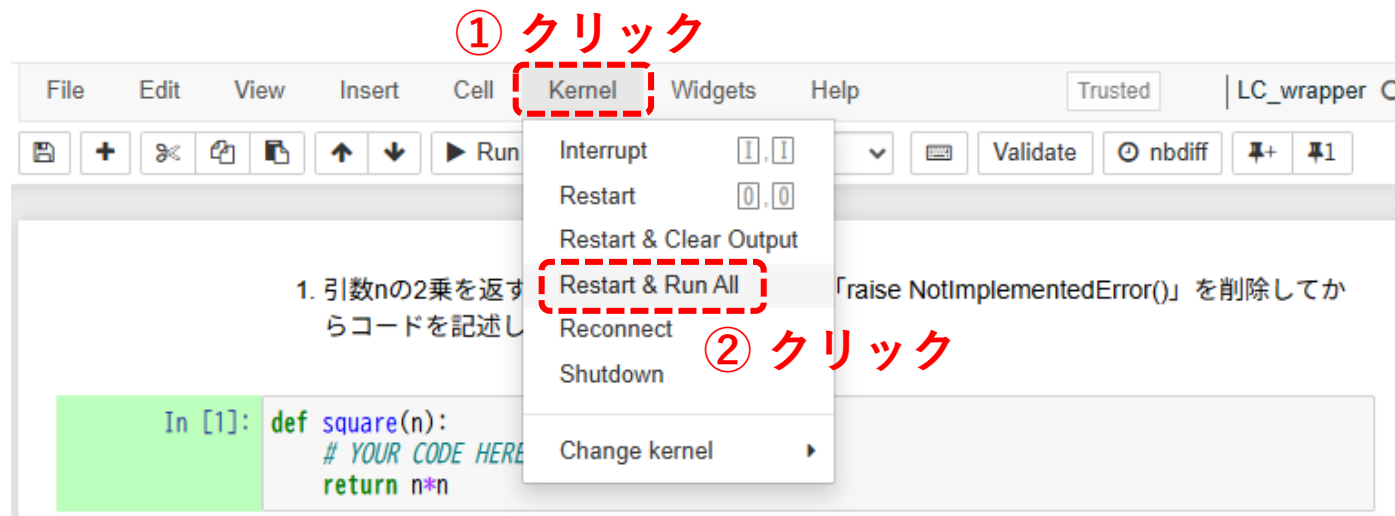


- 実行結果の表示に基づいて、解答の正誤を判断。
  - 教師側はそうように確認用のセルを作成すること。

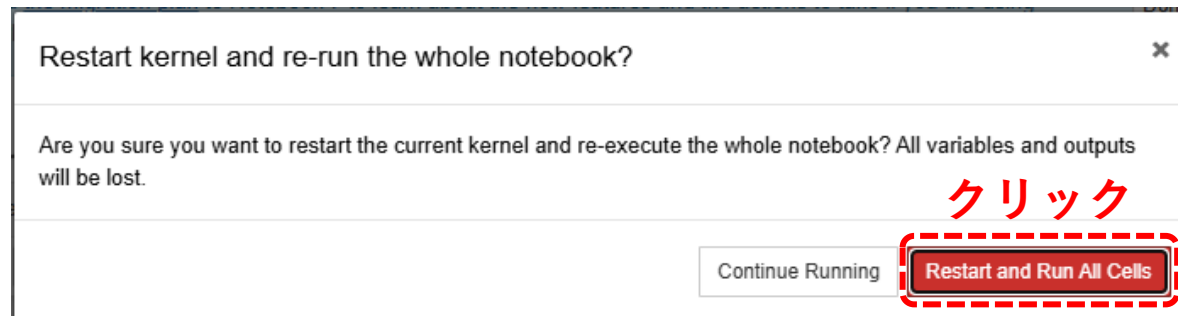


# 解答の作成 (3/4)

- 画面上部のメニューバーの中から「Kernel」をクリックし、続いて、「Restart & Run All」をクリック。



- 「Restart and Run All Cells」をクリック。エラーが発生しなければOK。



# 解答の作成 (4/4)

- 画面左上のディスクのアイコンをクリックして、ファイルを保存。



クリックしてファイルを保存

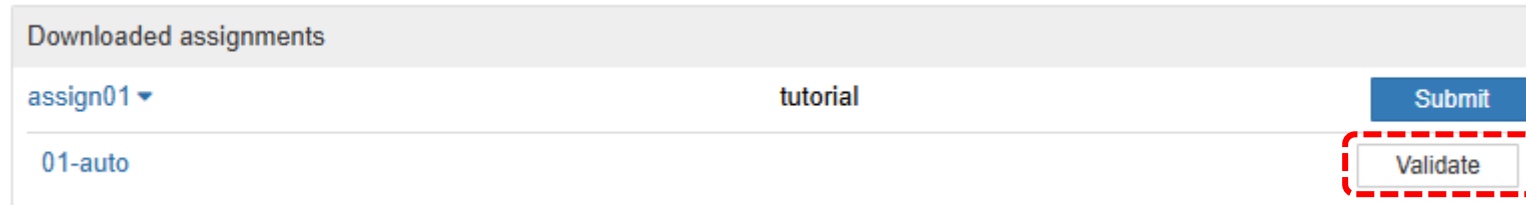
- タブを削除してホームに戻る。

クリックしてタブを削除



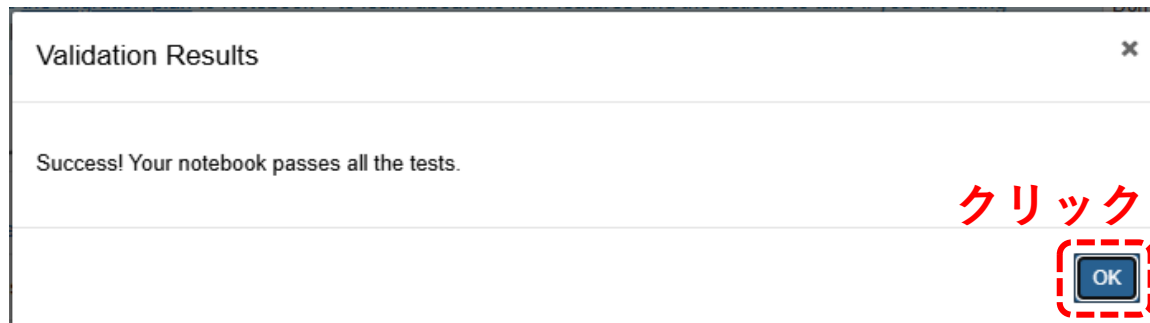
# 解答の提出 (1/4)

- 課題ファイルについて「Validate」をクリック。



クリックして  
ファイルを検証

- エラーが表示されず、以下の画面が表示されたらOK。
  - 「OK」をクリックして元の画面に戻る



クリック



# 解答の提出 (2/4)

- 課題「assign02」について、「Submit」をクリック。

Downloaded assignments		
assign01 ▼	tutorial	<a href="#">Submit</a>
01-auto		<a href="#">Validate</a>

クリックして  
解答を提出

- 提出が完了すると、フィードバックを取得するためのボタンが表示。

Submitted assignments		
assign01	tutorial	<a href="#">Fetch Feedback</a>
2025-08-21 16:19:39 JST		

# 解答の提出 (3/3)

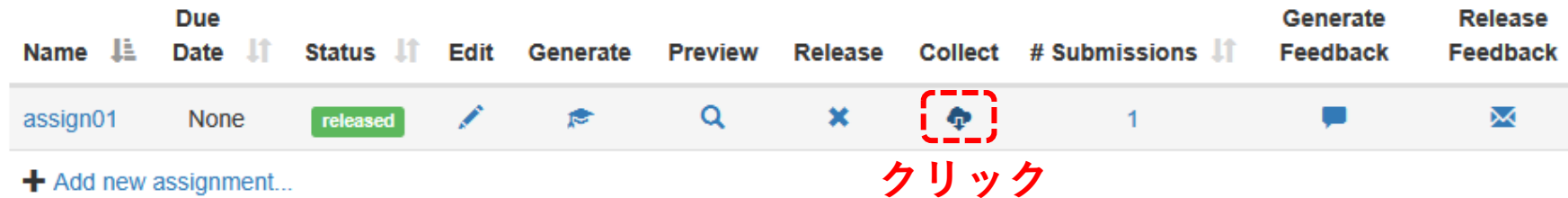
- みなさんからの提出を待って、セミナーの講師が提出物を採点し、採点結果を受講生向けに公開しますので、しばらくお待ちください。
- 採点の操作をご覧いただけますと幸いです。

# 課題の回収・採点

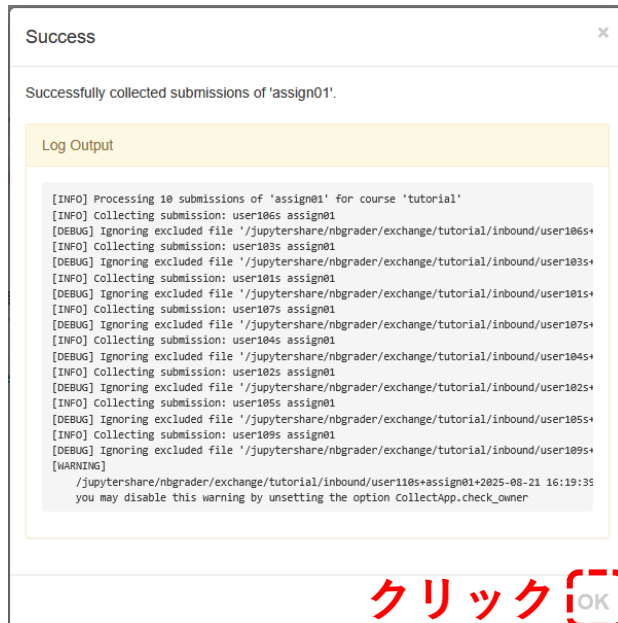
セミナーの講師が、教師役として提出物を回収・採点します。

# 課題の回収 (1/2)

- 課題の一覧の画面で、対象の課題のCollectアイコンをクリック。



- 「OK」をクリック。



# 課題の回収 (2/2)

- 新規の提出があった場合、提出数 (# Submissions) の値が増加。

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
assign01	None	released						10		
+ Add new assignment...										

値が増加

# 課題の採点（1/5）

- 対象の課題の提出数をクリック。

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
assign01	None	released						10		
+ Add new assignment...										

クリック

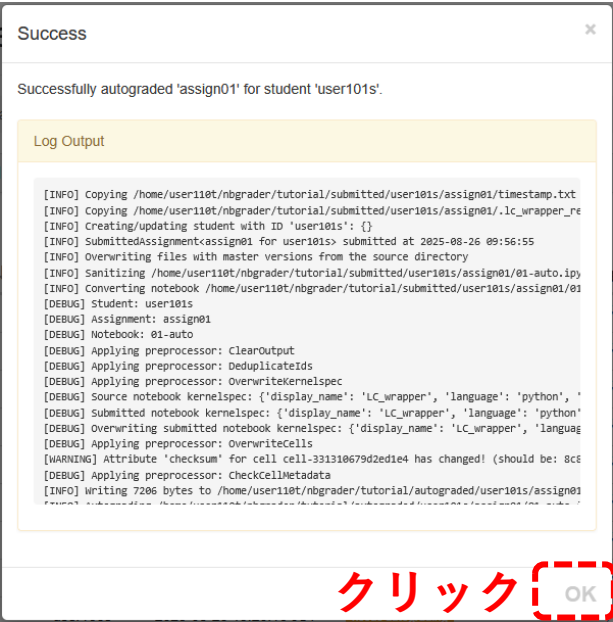
- 1名ずつ採点する場合、対象学生の自動採点アイコンをクリック。

Student Name	Student ID	Timestamp	Status	Score	Autograde	Generate Feedback	Release Feedback
student, user101	user101s	2025-08-26 18:56:55 JST	needs autograding		⚡		
student, user102	user102s	2025-08-26 18:58:35 JST	needs autograding		⚡		
student, user103	user103s	2025-08-26 19:00:22 JST	needs autograding		⚡		
student, user104	user104s	2025-08-26 19:06:21 JST	needs autograding		⚡		

クリック

# 課題の採点 (2/5)

- 「OK」をクリック。



- 手動で採点すべき課題が含まれていなければ、採点終了。

採点終了（graded）に変化

Student Name	Student ID	Timestamp	Status	Score	Autograde	Generate Feedback	Release Feedback
student, user101	user101s	2025-08-26 18:56:55 JST	graded	1 / 1	⚡	💬	✉
student, user102	user102s	2025-08-26 18:58:35 JST	needs autograding		⚡	💬	✉

# 課題の採点 (3/5)

- 一括で採点する場合、ホーム画面からターミナルを起動。



- 「nbgrader/{コース名}」のディレクトリに移動。
  - この例では「nbgrader/tutorial」ディレクトリに移動。

```
user110t@b7972d6e49ef:~$ cd nbgrader/tutorial
user110t@b7972d6e49ef:~/nbgrader/tutorial$
```



## 課題の採点 (4/5)

- 以下のコマンドを実行して一括採点を実行（assign01は課題名）。

```
nbgrader autograde "assign01" --force
```

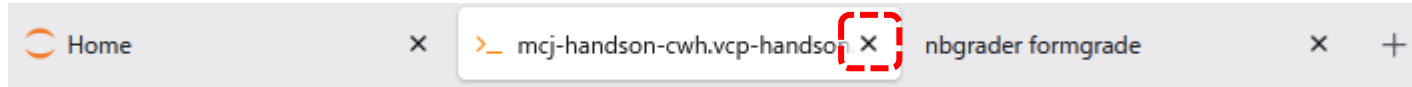
- 採点対象の提出物（ノートブック）を順番にすべて実行するだけの時間を要するため、しばらく待機。
- 採点が完了するとプロンプトに戻る。

```
[AutogradeApp | INFO] Writing 8072 bytes to /home/user110t/nbgrader/tutorial/autograded/user110s/assign01/01-auto.ipynb
[AutogradeApp | INFO] Setting destination file permissions to 444
user110t@b7972d6e49ef:~/nbgrader/tutorial$
user110t@b7972d6e49ef:~/nbgrader/tutorial$ █
```

# 課題の採点 (5/5)

- ターミナルのタブを閉じる。

クリックしてタブを閉じる

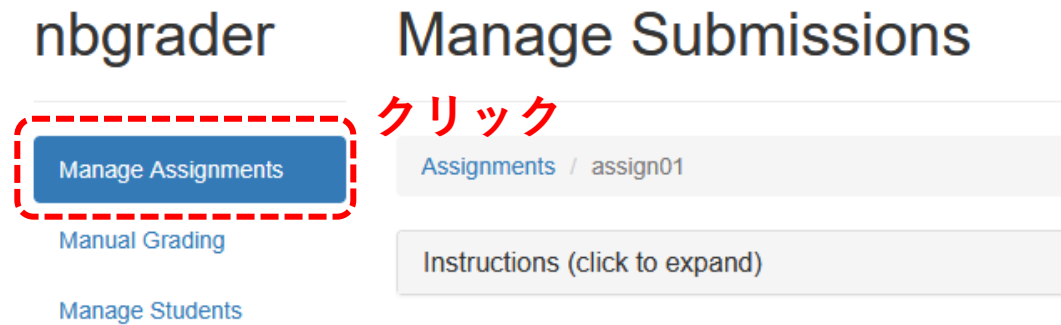


- 提出物一覧の画面を再読み込みすると、採点が完了している。

Student Name	Student ID	Timestamp	Status	Score	Autograde	Generate Feedback	Release Feedback
student, user101	user101s	2025-08-26 18:56:55 JST	graded	1 / 1	⚡	💬	✉
student, user102	user102s	2025-08-26 18:58:35 JST	graded	1 / 1	⚡	💬	✉
student, user103	user103s	2025-08-26 19:00:22 JST	graded	0 / 1	⚡	💬	✉
student, user104	user104s	2025-08-26 19:06:21 JST	graded	1 / 1	⚡	💬	✉
student, user105	user105s	2025-08-26 19:09:20 JST	graded	1 / 1	⚡	💬	✉
student, user106	user106s	2025-08-26 19:12:11 JST	graded	0 / 1	⚡	💬	✉
student, user107	user107s	2025-08-26 19:17:33 JST	graded	0 / 1	⚡	💬	✉
student, user108	user108s	2025-08-26 19:18:51 JST	graded	1 / 1	⚡	💬	✉
student, user109	user109s	2025-08-26 19:20:10 JST	graded	1 / 1	⚡	💬	✉
student, user110	user110s	2025-08-21 16:19:39 JST	graded	1 / 1	⚡	💬	✉

# フィードバックの作成・公開 (1/3)

- 「Manage Assignments」をクリック。

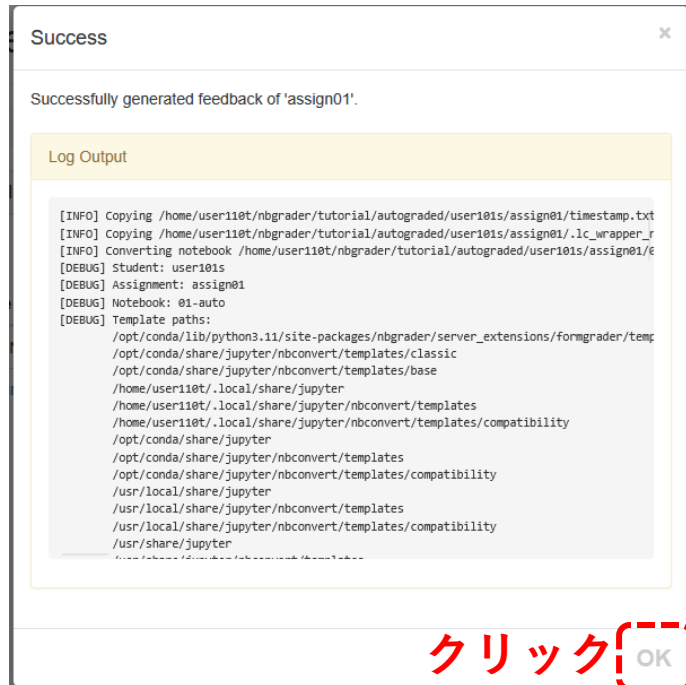


- 対象の課題の「Generate Feedback」アイコンをクリック。



# フィードバックの作成・公開 (2/3)

- 「OK」をクリック。



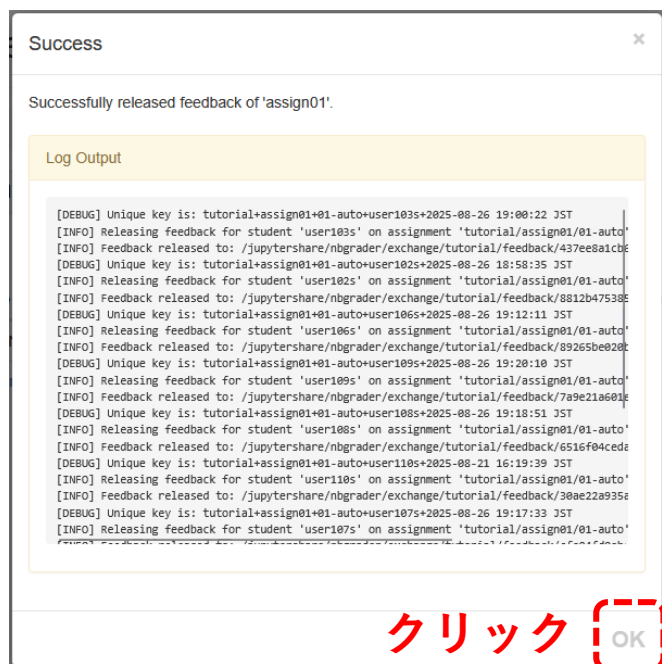
# フィードバックの作成・公開 (3/3)

セミナーの講師

- 対象の課題の「Release Feedback」をクリック。



- 「OK」をクリック。



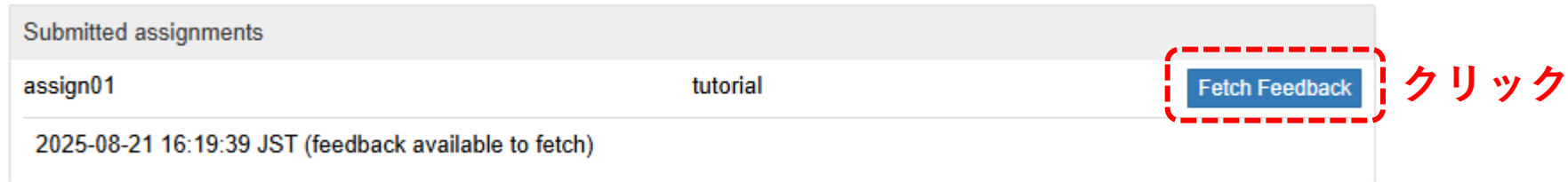
受講生はフィードバックを取得可能になります。

# フィードバックの取得（学生）

学生として、提出した課題についての採点結果を取得してもらいます。

# フィードバックの取得 (1/3)

- 「assign01」について「Fetch feedback」をクリック。



- フィードバックが公開されていたらリンクが作成されるので、リンクをクリック。



# フィードバックの取得 (2/3)

- ファイル名をクリックしてフィードバックの内容を確認。

<input type="checkbox"/> 0	/ tutorial / assign01 / feedback / 2025-08-21 16:19:39 JST			Name ▼	Last Modified	File size
	..				数秒前	
<input type="checkbox"/>	<a href="#">01-auto.html</a>	クリックしてフィードバックを表示			3分前	129 kB



# フィードバックの取得 (3/3)

- フィードバックの内容を確認。

01-auto (Score: 1.0 / 1.0)

1. Test cell (Score: 1.0 / 1.0)

引数nの2乗を返す関数squareを定義します。「raise NotImplementedError()」を削除してからコードを記述してください。

In [1]:

Student's answer

(Top)

```
def square(n):  
    # YOUR CODE HERE  
    return n*n
```

このセルでプログラムの計算結果を確認しましょう。

In [2]:

```
print(square(5))
```

25

In [3]:

Grade cell: cell-331310679d2ed1e4

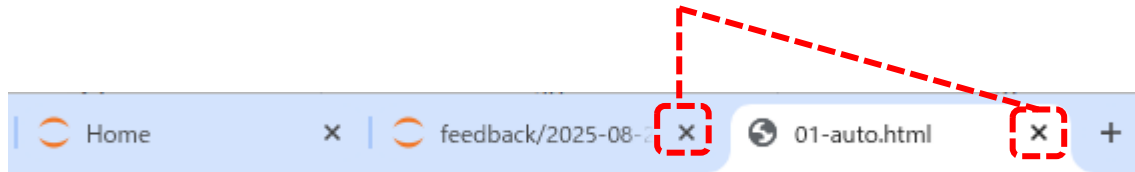
Score: 1.0 / 1.0 (Top)

```
### BEGIN HIDDEN TESTS  
assert square(1) == 1  
assert square(2) == 4  
assert square(3) == 9  
assert square(4) == 16  
### END HIDDEN TESTS
```

# ログアウト (1/3)

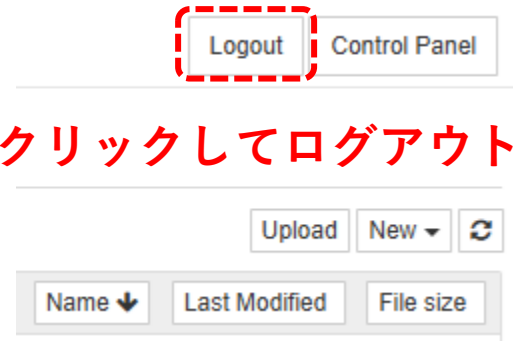
- タブを削除していき、ホームに戻る。

クリックしてタブを削除



## ログアウト (2/3)

- 画面右上の「Logout」をクリックしてMCJ-CloudHubからログアウト。



- MCJ-CloudHubのタブを削除。



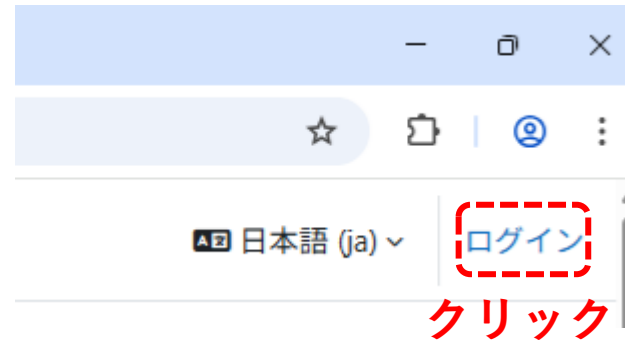
# ログアウト (3/3)

- Moodleからもログアウト。
  - 次の演習では皆さんに教師役になっていただくため、Moodleからもログアウトします。

① 画面右上のユーザのアイコンをクリック



② クリックしてログアウト



# Moodleコンテンツの準備（教師）

教師としてMoodleにログインして、外部ツールを貼り付けてもらいます。

# 課題ファイルのダウンロード

- このセミナーでは既存の課題ファイルをWeb上からダウンロードしておき、演習環境にアップロードして使用します。
- 以下のWebページから「01-auto.ipynb」と「01-manual.ipynb」をダウンロードしてください。
  - <https://github.com/nii-gakunin-cloud/handson>

## 2025年度ハンズオン

- 2025年9月2日(火) 学認クラウドオンデマンド構築サービスハンズオンセミナー
  - MCJ-CloudHubの概要 [教材](#)
  - 講義・演習環境アプリケーションハンズオン (演習) [auto 教材](#)
  - 講義・演習環境アプリケーションハンズオン (演習) [Notebook1](#)
  - 講義・演習環境アプリケーションハンズオン (演習) [mamual 教材](#)
  - 講義・演習環境アプリケーションハンズオン (演習) [Notebook2](#)
  - オンデマンド構築サービスの仕組みと環境構築概要 [教材](#)

# 教師としてMoodleにログイン

- 以下のURLにアクセス。
  - <https://mcj-handson-cwh.vcp-handson.org/moodle/login/index.php>
- 割り当てられたユーザ名とパスワードでログイン。
  - 今回は教師用のアカウントを使用してログイン。

割り当てられた教師用  
アカウントでログイン



A screenshot of the Moodle login page. A red dashed box highlights the login fields: a text input containing 'user110t', a password input with dots, and a blue 'ログイン' (Login) button. Below the box is a link 'パスワードを忘れましたか?' (Forgot your password?). Further down, there is a section titled 'いくつかのコースにはゲストアクセスできます' (Some courses allow guest access) with a button 'ゲストとしてアクセスする' (Access as guest). At the bottom, there are links for '日本語 (ja)' and a 'クッキー通知' (Cookie notice) button.



A screenshot of the Moodle dashboard. At the top, there is a navigation bar with links: 'New Site', 'Home', 'ダッシュボード', and 'マイコース' (My Course), which is highlighted with a red dashed box. To the right of the navigation bar are icons for notifications, messages, and a user profile 'tu'. A red text label 'クリックしてマイコースに移動' (Click to move to My Course) points to the 'マイコース' link. Below the navigation bar, the main content area starts with a greeting 'こんにちは teacher さん!' (Hello teacher!). This is followed by a 'タイムライン' (Timeline) section with filters for '次の7日' (Next 7 days) and '日付で並べ替える' (Sort by date), and a search bar for '活動タイプまたは活動名で検索する' (Search by activity type or name). At the bottom, there is a message 'アクションを必要とする活動はありません。' (There are no activities requiring action).

# コースに移動してMCJ-CloudHubを起動

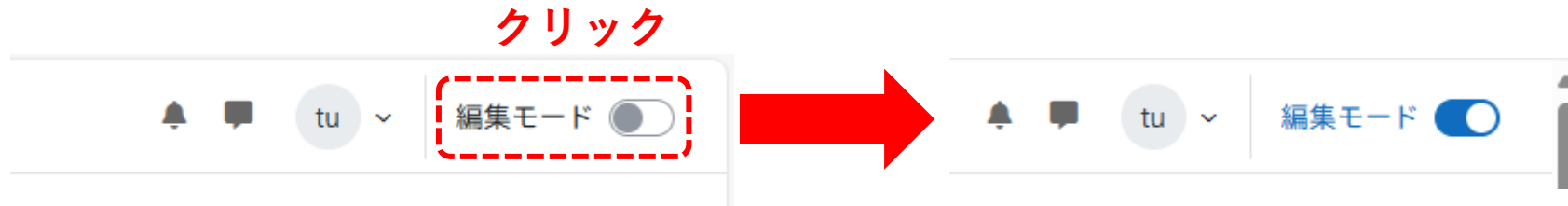
- マイコースに自分のユーザ番号に対応したコースがあるので、そのコースアクセス（画面例とは異なるかも）。
- 最初はコース内のコンテンツが無い（アナウンスメントだけ）。





# コースを編集モードに変更

- 画面右上のスイッチのアイコンをクリック。



- 連動してコースの見た目に変更される。



# 外部ツールの貼り付け (1/3)

- どのセクションでも構わないので「活動またはリソースを追加する」をクリック。
- 「mcjcloudhub」をクリック。
  - ハンズオン環境ではMoodleの「外部ツール」に対して、MCJ-CloudHubへの接続設定を含め、別名（mcjcloudhub）を付与したものを準備しています。



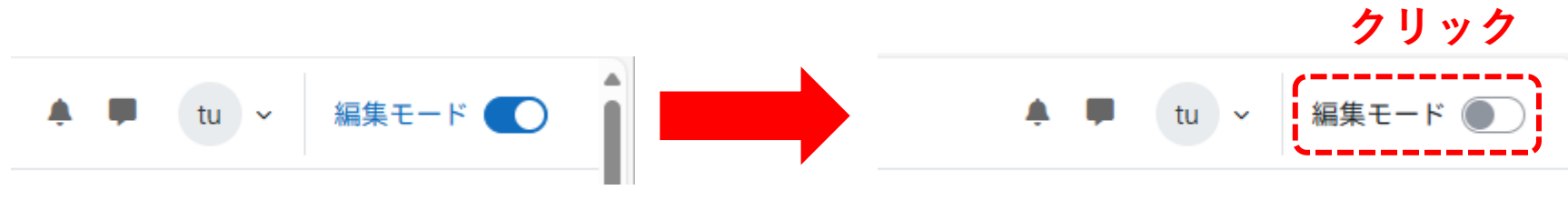
# 外部ツールの貼り付け (2/3)

- 設定画面が表示されたら、画面での表示名を設定。
- 下方向にスクロールした後、「保存してコースに戻る」をクリック。
- 設定した表示名でリンクが作成されている。



# 外部ツールの貼り付け (3/3)

- 画面右上のスイッチのアイコンをクリックして、コースを編集モードから通常モードに戻す。



- 連動してコースの見た目が変更される。

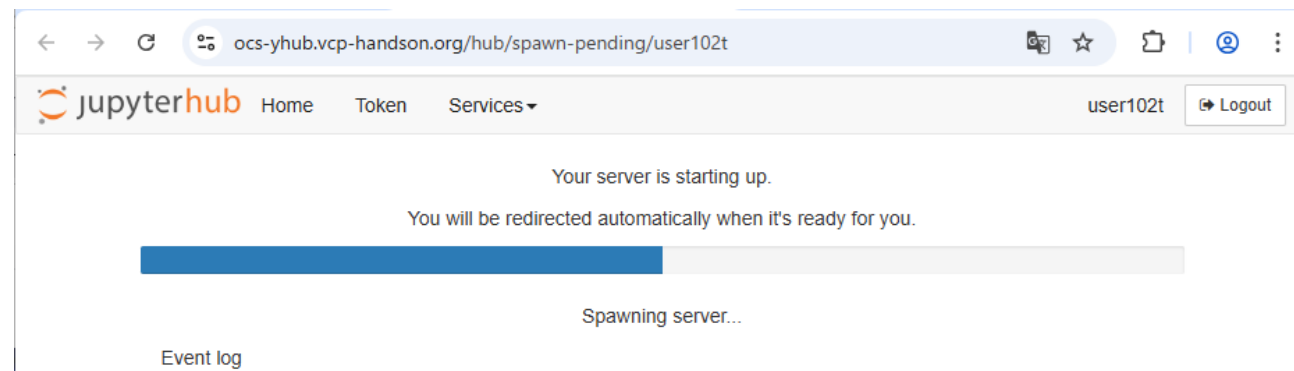


# 課題の作成・公開（教師）

教師役として課題を作成・公開していただきます。

# Jupyter環境の起動 (1/3)

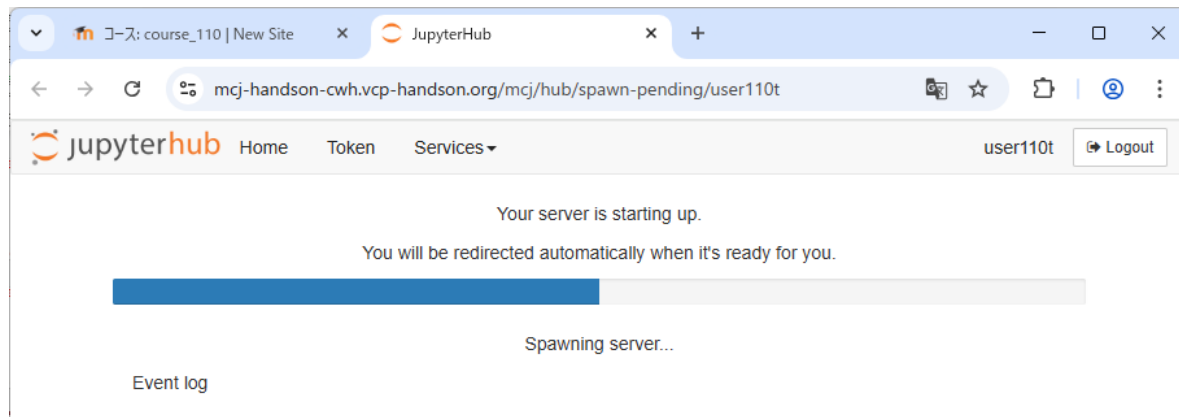
- 先ほど作成したMCJ-CloudHubへのリンクをクリック。
- Jupyter環境が起動するまで待機。



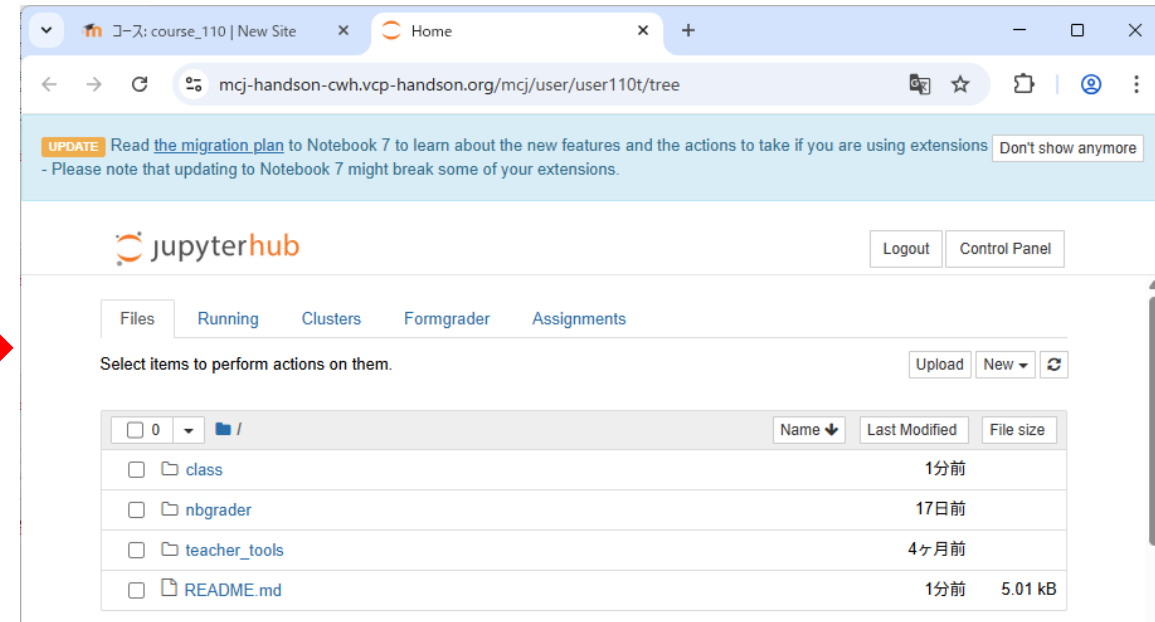
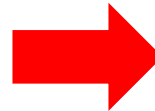
Jupyter環境が起動するまで待機

# Jupyter環境の起動 (2/3)

- 画面上部の「Formgrader」をクリックして課題一覧画面に移動。
- 「Add new assignment...」をクリック。



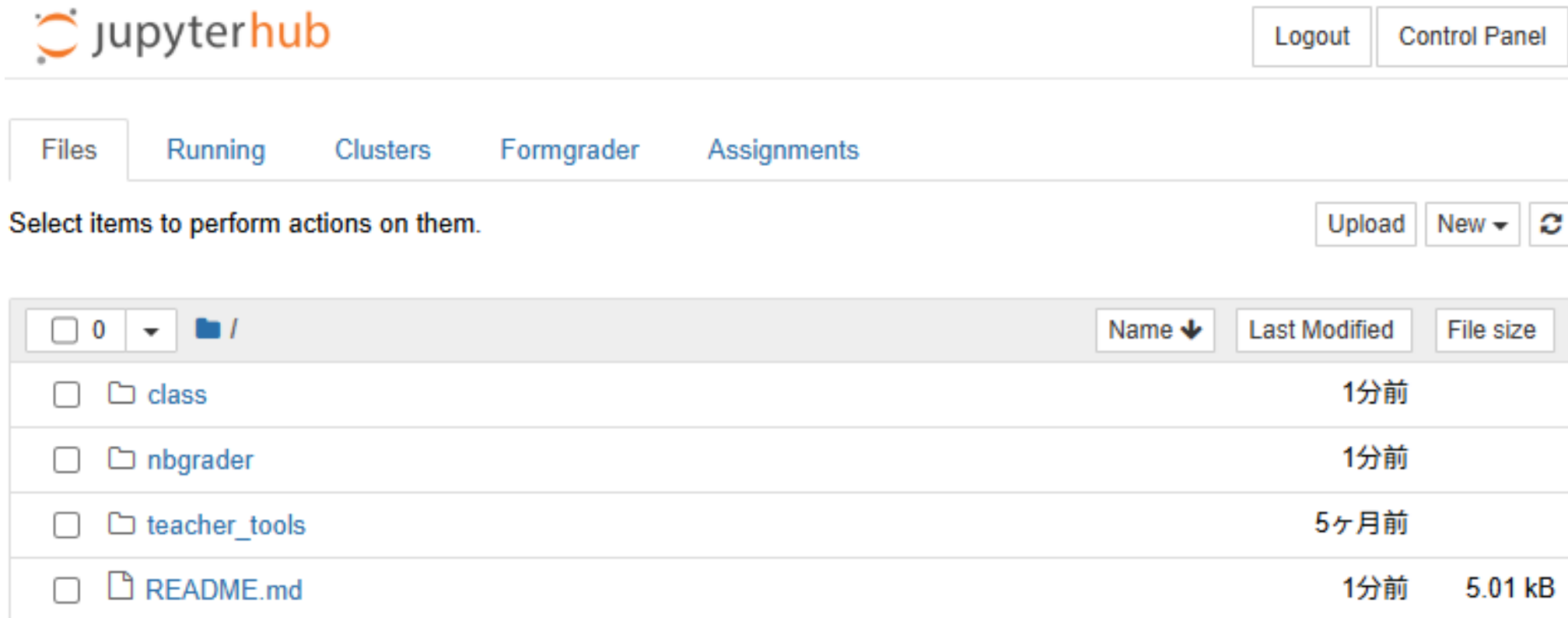
**Jupyter環境が起動するまで待機**



**起動が完了するとファイルやフォルダの一覧が表示**

# Jupyter環境の起動 (3/3)

- 起動が完了するとファイルやフォルダの一覧が表示。



The screenshot displays the JupyterHub web interface. At the top left is the JupyterHub logo. To the right are buttons for 'Logout' and 'Control Panel'. Below the logo is a navigation bar with tabs for 'Files', 'Running', 'Clusters', 'Formgrader', and 'Assignments'. The 'Files' tab is active. Below the navigation bar, there is a prompt 'Select items to perform actions on them.' and buttons for 'Upload', 'New', and a refresh icon. The main content area shows a file browser view for the root directory '/'. It includes a table with columns for 'Name', 'Last Modified', and 'File size'. The table lists four items: 'class' (folder, 1分前), 'nbgrader' (folder, 1分前), 'teacher\_tools' (folder, 5ヶ月前), and 'README.md' (file, 1分前, 5.01 kB).

	Name ↓	Last Modified	File size
<input type="checkbox"/>	0	↓	📁 /
<input type="checkbox"/>	📁 class	1分前	
<input type="checkbox"/>	📁 nbgrader	1分前	
<input type="checkbox"/>	📁 teacher_tools	5ヶ月前	
<input type="checkbox"/>	📄 README.md	1分前	5.01 kB



# 課題の作成 (1/7)

- 画面上部の「Formgrader」をクリックして課題一覧画面に移動。
- 「Add new assignment...」をクリック。

jupyterhub

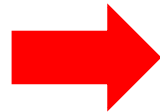
クリックして課題一覧画面に移動

Files Running Clusters **Formgrader** Assignments

Select items to perform actions on them.

0 /

- ☐ class
- ☐ nbgrader
- ☐ teacher\_tools
- ☐ README.md



Instructions (click to expand)

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
No data available in table								

**+ Add new assignment...** クリックして課題の登録を開始

# 課題の作成 (2/7)

- 課題の名称（今回はassign01）を入力して、「Save」をクリック。
- 課題一覧画面に戻ったら課題の名称（assgin01）をクリック。

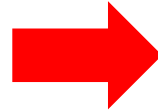
Add New Assignment ×

① 課題の名称を入力

Name

Due date (optional)

② クリックして課題を登録



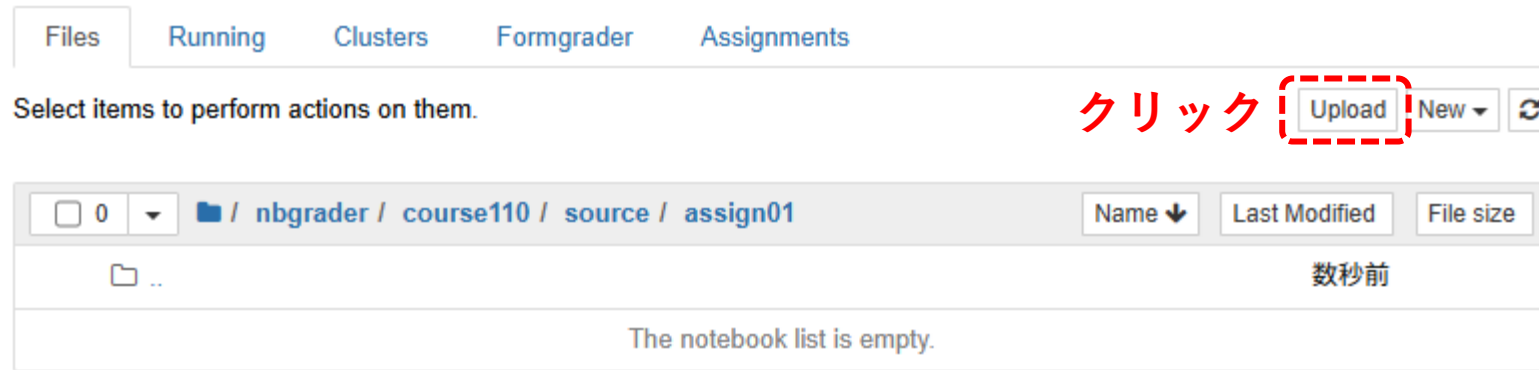
Instructions (click to expand)

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
<a href="#">assign01</a>	None	<span>draft</span>						0

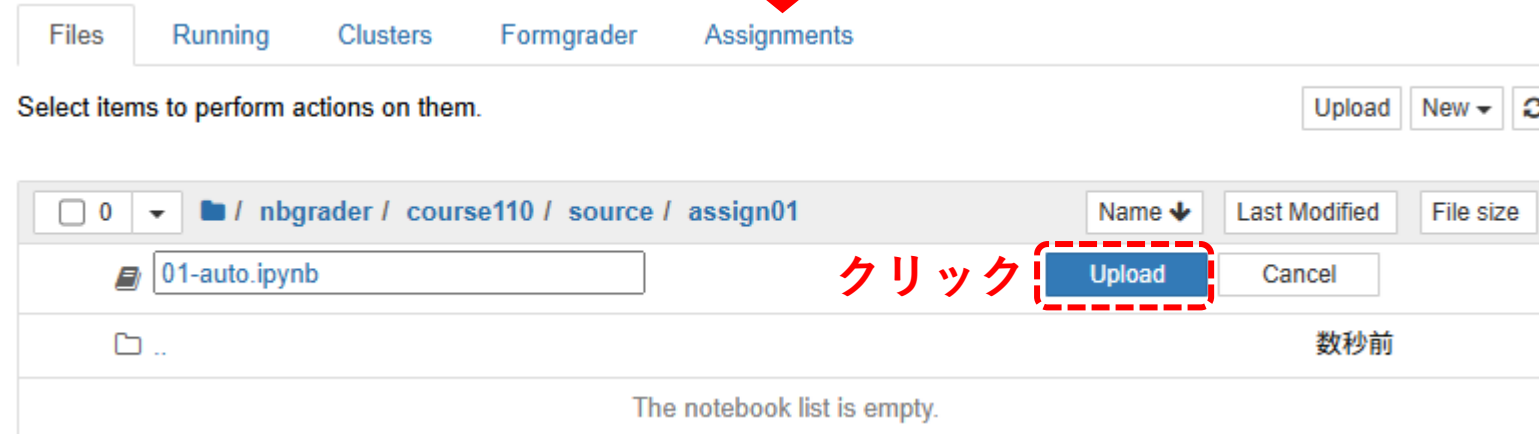
課題名をクリック

# 課題の作成 (3/7)

- 画面右上の「Upload」をクリック。
- PC上で課題ファイル (01-auto.ipynb) を選択。
- 該当するファイル名の横にある「Upload」をクリック。



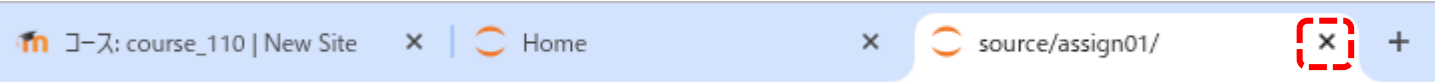
PC上で課題ファイルを選択して・・・



# 課題の作成（4/7）

- タブを削除してホームに戻る。

クリックしてタブを削除



クリックして課題一覧画面に移動

jupyterhub

Files

Running

Clusters

Formgrader

Assignments

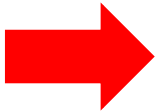
Select items to perform actions on them.

☐ 0
 

▼

📁 /

☐ 📁 class
 ☐ 📁 nbgrader
 ☐ 📁 teacher\_tools
 ☐ 📄 README.md

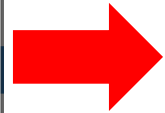
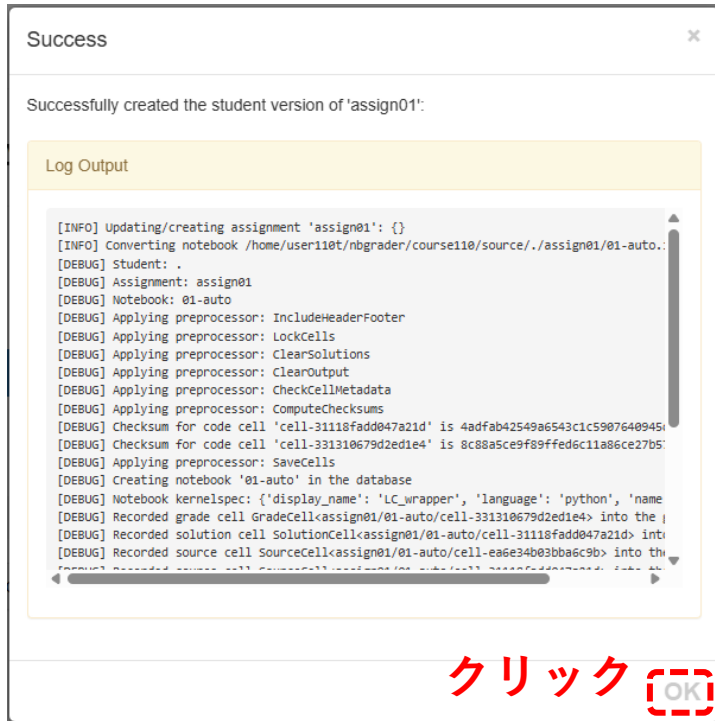


	Due									
Name	Date	Status	Edit	Generate	Preview	Release	Collect	Submissions	#	
assign01	None	draft							0	
+ Add new assignment...										

クリック

# 課題の作成 (5/7)

- 「Success」の画面が表示されたら画面下部の「OK」をクリック。
- 続いて、「Preview」のアイコンをクリック。



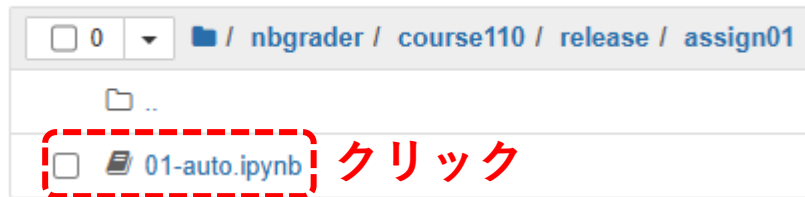
Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
assign01	None	draft						0
+ Add new assignment...								

クリックして配布用ファイルを表示

# 課題の作成 (6/7)

- ファイルの一覧が表示されたら、配布用ファイルをクリック。

Select items to perform actions on them.



- ファイルの内容を確認。

引数nの2乗を返す関数squareを定義します。「raise NotImplementedError()」を削除してからコードを記述してください。

```
In [ ]: def square(n):  
        # YOUR CODE HERE  
        raise NotImplementedError()
```

このセルでプログラムの計算結果を確認してみましょう。

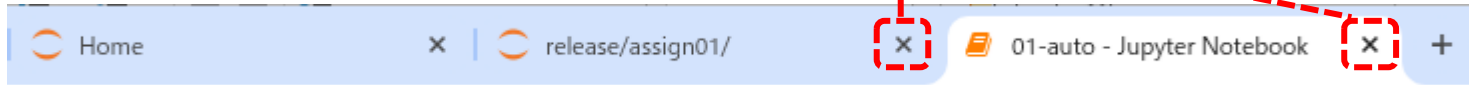
```
In [ ]: print(square(5))
```

```
In [ ]:
```

# 課題の作成 (7/7)

- タブを削除していき、ホームに戻る。

クリックしてタブを削除



# 課題の公開 (1/3)

- 「assign01」の行の「Release」のアイコンをクリック。

jupyterhub

クリックして課題一覧画面に移動

Files Running Clusters **Formgrader** Assignments

Select items to perform actions on them.

☐ 0 /

- ☐ class
- ☐ nbgrader
- ☐ teacher\_tools

➔

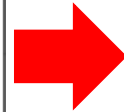
Name ↓↑	Due Date ↓↑	Status ↓↑	Edit	Generate	Preview	Release	Collect	# Submissions
assign01	None	draft						0
+ Add new assignment...								

クリックして課題の公開を開始



## 課題の公開 (2/3)

- 「Success」の画面が表示されたら画面下部の「OK」をクリック。



Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
assign01	None	released						0
+ Add new assignment...								

回収アイコンや提出数が表示される

- タブを削除して、ホームに戻る。

クリックしてタブを削除



# 課題の公開 (3/3)

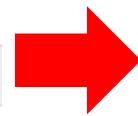
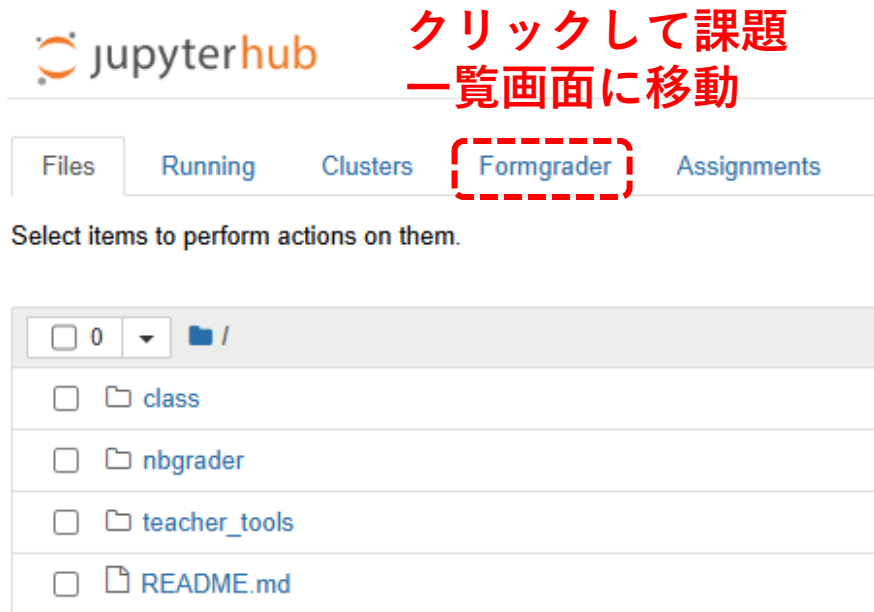
- セミナーのスタッフが学生役になって課題を提出しますので、しばらくお待ちください。

# 課題の回収・採点（教師）

教師役として課題を回収・採点していただきます。

# 課題の回収 (1/2)

- 画面上部の「Formgrader」をクリックして課題一覧画面に移動。
- 課題「assign01」の行にある回収アイコンをクリック。

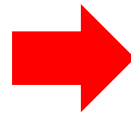
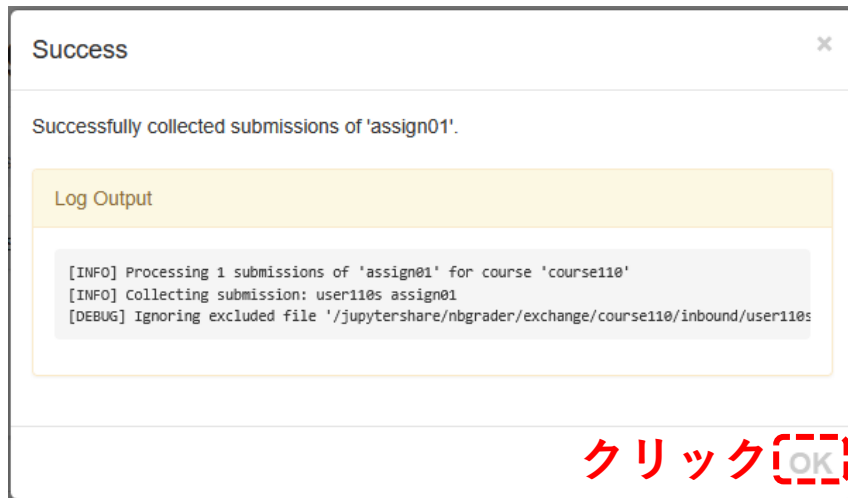


Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
assign01	None	released						0
+ Add new assignment...								

クリックして提出物を回収

## 課題の回収 (2/2)

- 以下のような画面が表示されたら「OK」をクリック。
- 新規の提出があると、提出数が増える。



Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
assign01	None	released						1
+ Add new assignment...								

新規の提出があると提出数が増える

# 課題の自動採点 (1/2)

- 課題の提出数をクリック。

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
assign01	None	released						1
+ Add new assignment...								

クリックして提出一覧に移動

- 自動採点を実行。

Student Name	Student ID	Timestamp	Status	Score	Autograde	Generate Feedback	Release Feedback
student, user110	user110s	2025-08-26 21:32:44 JST	needs autograding				

クリックして自動採点を開始

# 課題の自動採点 (2/2)

- 以下のような画面が表示されたら「OK」をクリック。
- 採点された提出物の状態が「graded」に変化。
  - 手動採点も含まれるファイルの場合には「needs manual grading」に変化。

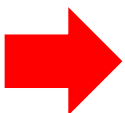
Success

Successfully autograded 'assign01' for student 'user110s'.

Log Output

```
[INFO] Copying /home/user110t/nbgrader/course110/submitted/user110s/assign01/timestamp.txt
[INFO] Copying /home/user110t/nbgrader/course110/submitted/user110s/assign01/.lc_wrapper_r
[INFO] Creating/updating student with ID 'user110s': {}
[INFO] SubmittedAssignment<assign01 for user110s> submitted at 2025-08-26 12:32:44
[INFO] Overwriting files with master versions from the source directory
[INFO] Sanitizing /home/user110t/nbgrader/course110/submitted/user110s/assign01/01-auto.ip
[INFO] converting notebook /home/user110t/nbgrader/course110/submitted/user110s/assign01/0
[DEBUG] Student: user110s
[DEBUG] Assignment: assign01
[DEBUG] Notebook: 01-auto
[DEBUG] Applying preprocessor: ClearOutput
[DEBUG] Applying preprocessor: DeduplicateIds
[DEBUG] Applying preprocessor: OverwriteKernelspec
[DEBUG] Source notebook kernelspec: {'display_name': 'LC_wrapper', 'language': 'python', '
[DEBUG] Submitted notebook kernelspec: {'display_name': 'LC_wrapper', 'language': 'python'
[DEBUG] Overwriting submitted notebook kernelspec: {'display_name': 'LC_wrapper', 'langua
[DEBUG] Applying preprocessor: OverwriteCells
[WARNING] Attribute 'checksum' for cell cell-331310679d2ed1e4 has changed! (should be: 8c8
[DEBUG] Applying preprocessor: CheckCellMetadata
[INFO] Writing 7211 bytes to /home/user110t/nbgrader/course110/autograded/user110s/assigne
[INFO] Autograded /home/user110t/nbgrader/course110/autograded/user110s/assigne
```

クリック[OK]



Student Name	Student ID	Timestamp	Status	Score	Autograde
student, user110	user110s	2025-08-26 21:32:44 JST	graded	1 / 1	⚡

提出物の状態が「graded」に変化

# 補足

- GUI上の操作では多数の受講生に対して、順番に自動採点のアイコンをクリックする必要があります。
- JupyterHub上のターミナル上では、すべての（未評定の）提出物に対して順番に自動採点を進めるコマンドを使用可能です。



# フィードバックの作成・公開 (1/3)

- 画面左側の「Manage Assignments」をクリックして、課題一覧画面に移動。

nbgrader

Manage Assignments

クリック

Manual Grading

Manage Students

# フィードバックの作成・公開（2/3）

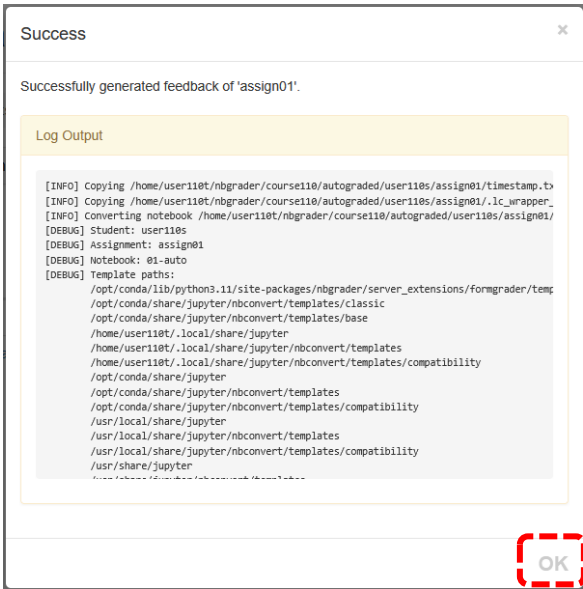
- 課題「assign01」のフィードバック作成のアイコンをクリック。

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
assign01	None	released						1		

+ Add new assignment...

クリックしてフィードバックを作成

- 以下のような画面が表示されたら「OK」をクリック。



OK クリック

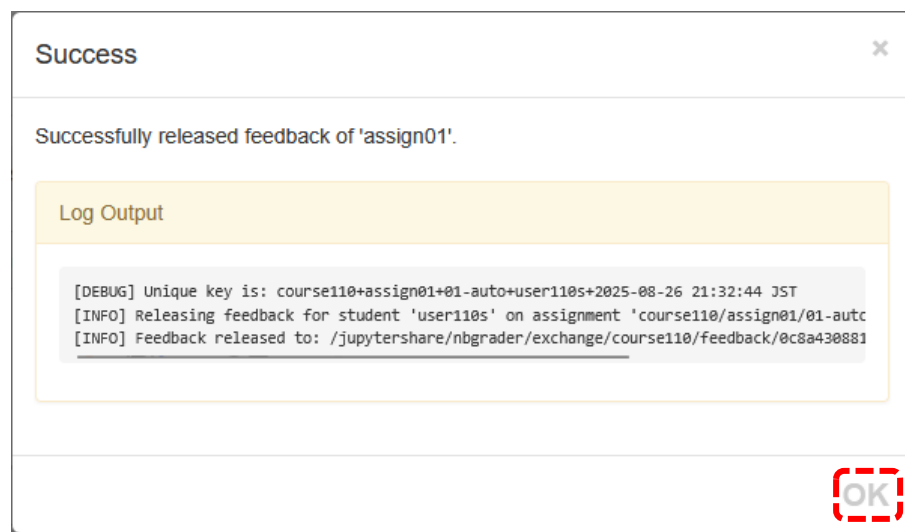
# フィードバックの作成・公開 (3/3)

- 課題「assign01」のフィードバック公開のアイコンをクリック。

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
assign01	None	released						1		
+ Add new assignment...										

クリックしてフィードバックを公開

- 以下のような画面が表示されたら「OK」をクリック。



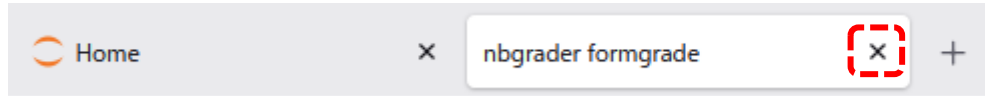
受講生はフィードバックを取得可能になります。

OK クリック

# ログアウト (1/3)

- タブを削除して、ホームに戻る。

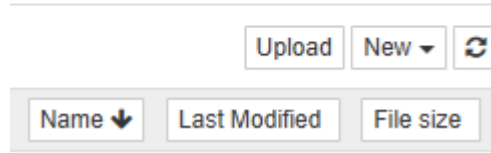
クリックしてタブを削除



- 画面右上の「Logout」をクリックしてMCJ-CloudHubからログアウト。



クリックしてログアウト



# ログアウト (2/3)

- MCJ-CloudHubのタブを削除。

クリックしてタブを削除



# ログアウト (3/3)

- Moodleからもログアウト。

① 画面右上のユーザのアイコンをクリック



② クリックしてログアウト



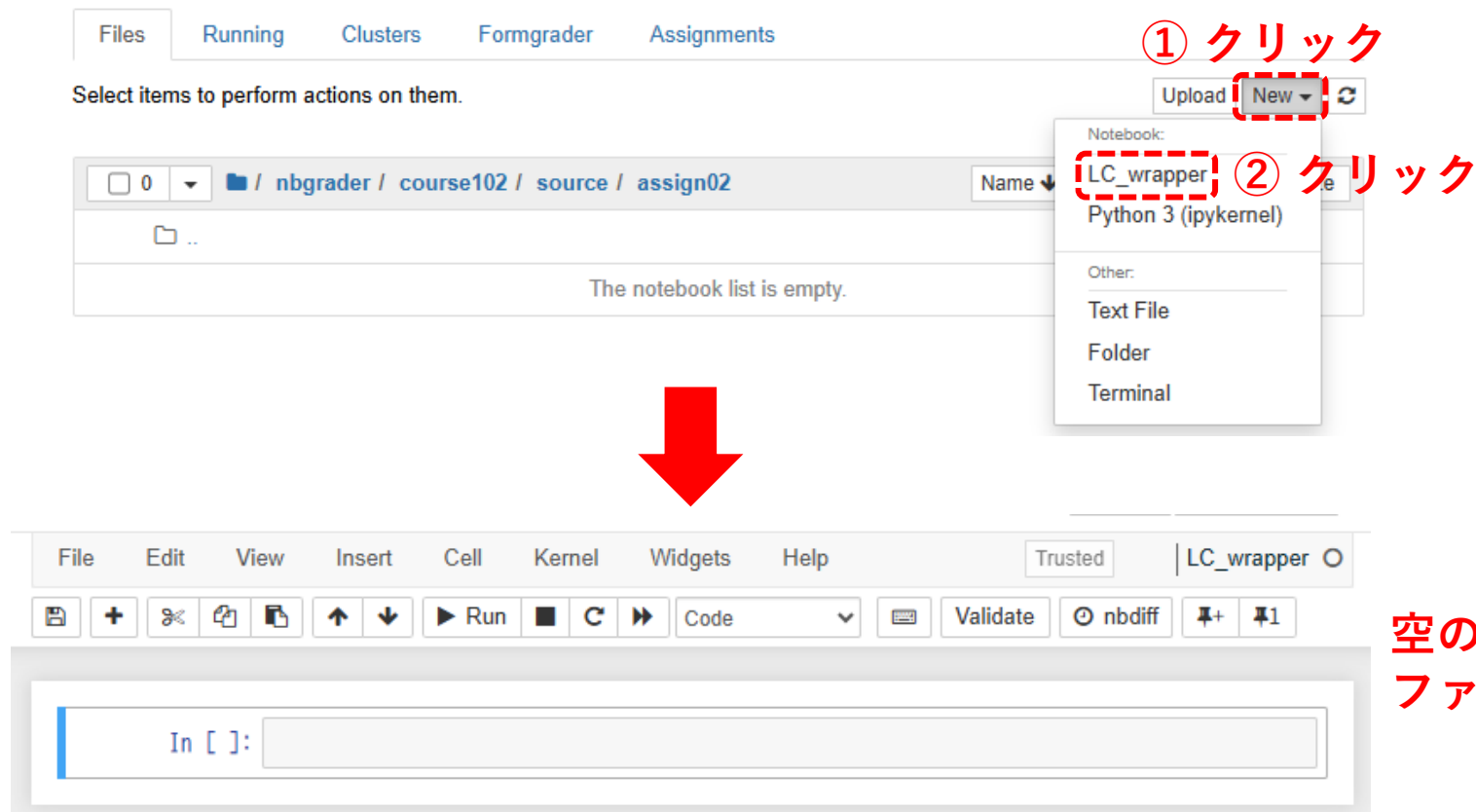
# 付録：課題ファイルの作成

セミナーで用いた課題ファイルの作成手順について説明します。



# 課題の作成 (1/16)

- 画面右上にある「new」をクリックして、「LC\_wrapper」を選択。
- 空のJupyter Notebookファイルが作成される。



空のJupyter Notebook  
ファイルが作成される

# 課題の作成 (2/16)

- 画面上部のファイル名（この時点ではUntitled）をクリック。
- ファイル名（01-auto.ipynb）を入力して「Rename」をクリック。

The image shows a sequence of three screenshots from the JupyterHub interface, illustrating the steps to rename a notebook. Red dashed boxes and arrows highlight the specific actions.

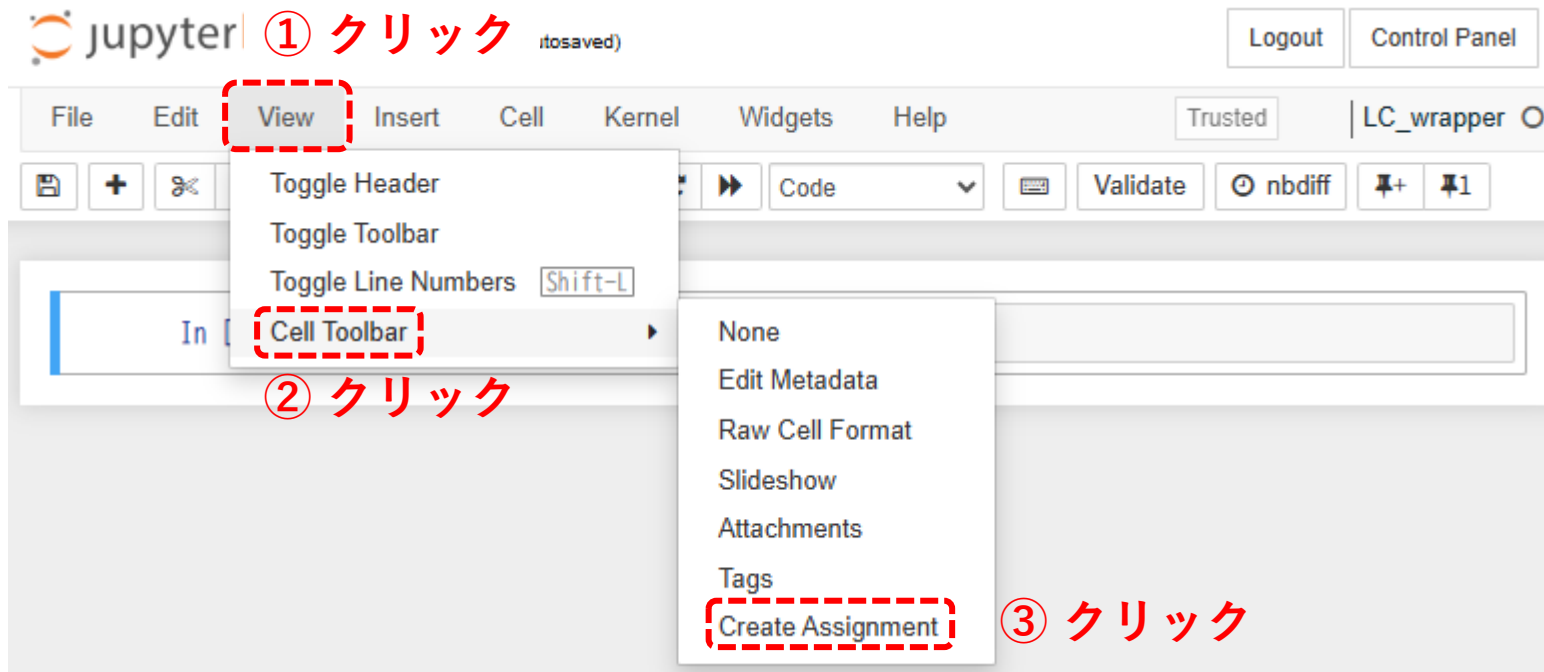
**Top Screenshot:** The JupyterHub interface shows the notebook titled "Untitled (unsaved changes)". The "File" menu is open, and the "Rename" option is highlighted with a red dashed box and the label "クリック" (Click).

**Middle Screenshot:** A "Rename Notebook" dialog box is displayed. The input field contains "01-auto.ipynb", which is highlighted with a red dashed box and the label "① ファイル名を入力" (1. Enter filename). The "Rename" button is also highlighted with a red dashed box and the label "② クリック" (2. Click).

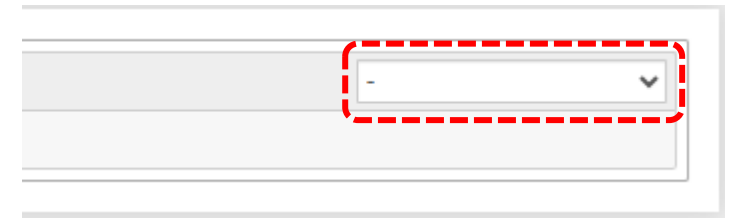
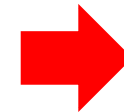
**Bottom Screenshot:** The JupyterHub interface shows the notebook titled "01-auto (autosaved)". The title is highlighted with a red dashed box and the label "ファイル名が変更されている" (Filename is being changed).

# 課題の作成 (3/16)

- メニューバーの中から「View」をクリックした後、「Cell Toolbar」、「Create Assignment」を順番に選択。
- セルの右上にリストボックスが表示。

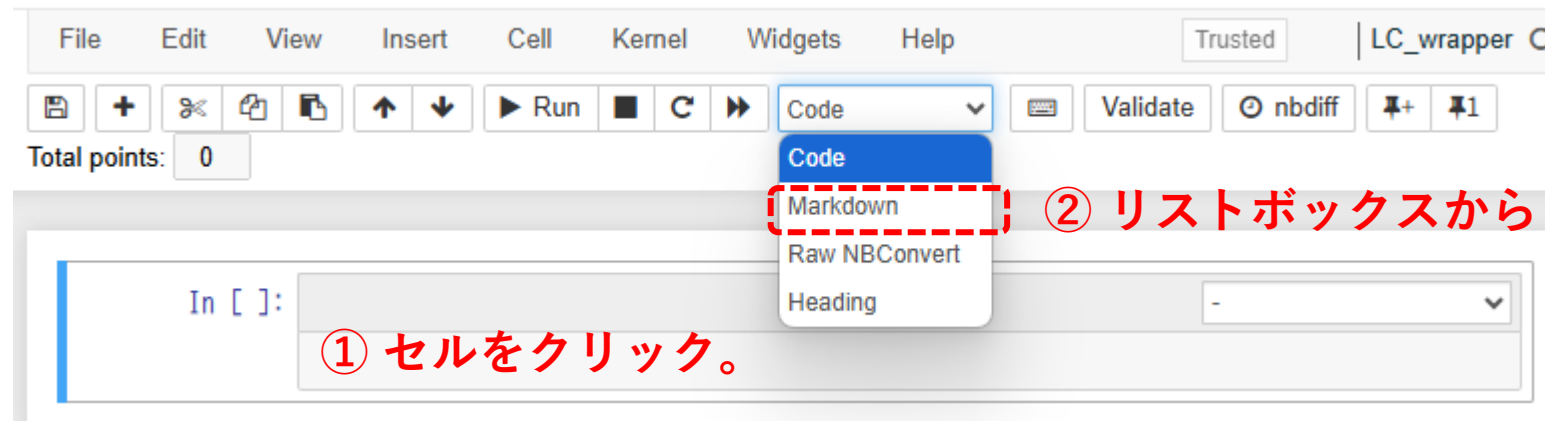


セルの右上にリスト  
ボックスが表示



# 課題の作成 (4/16)

- 最初のセルをクリックしてそこにフォーカスを移動。
- 画面上部のリストボックスをクリックした後、「Markdown」を選択。

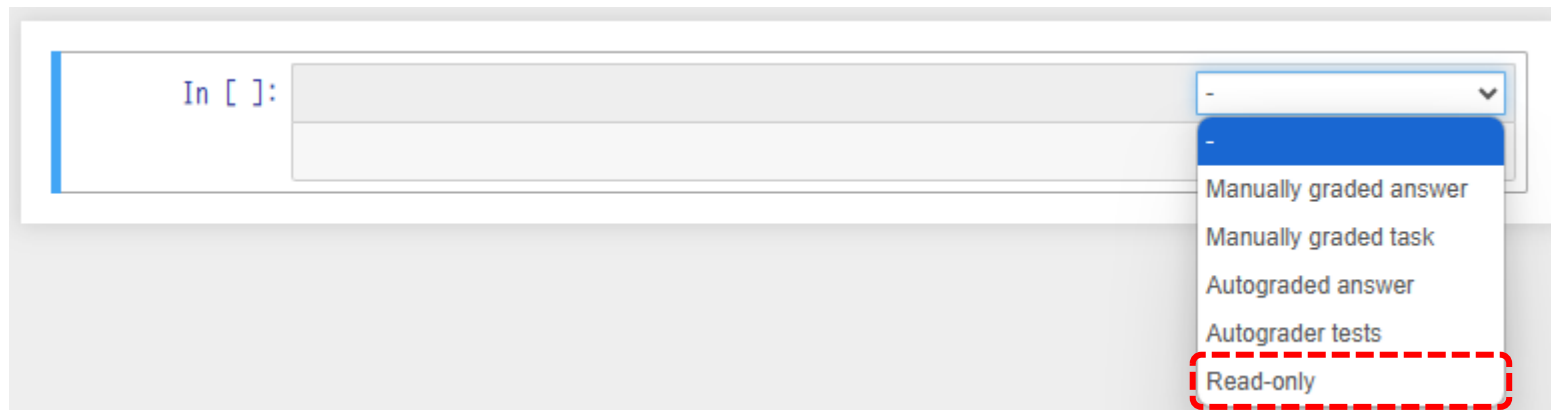


① セルをクリック。

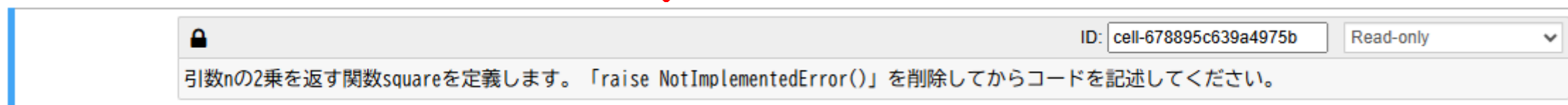
② リストボックスから「Markdown」を選択。

# 課題の作成 (5/16)

- セルの右上にあるリストボックスの中から「Read-only」を選択。
- セルの中に、以下のような問題文を記述。



① リストボックスから「Read-only」を選択



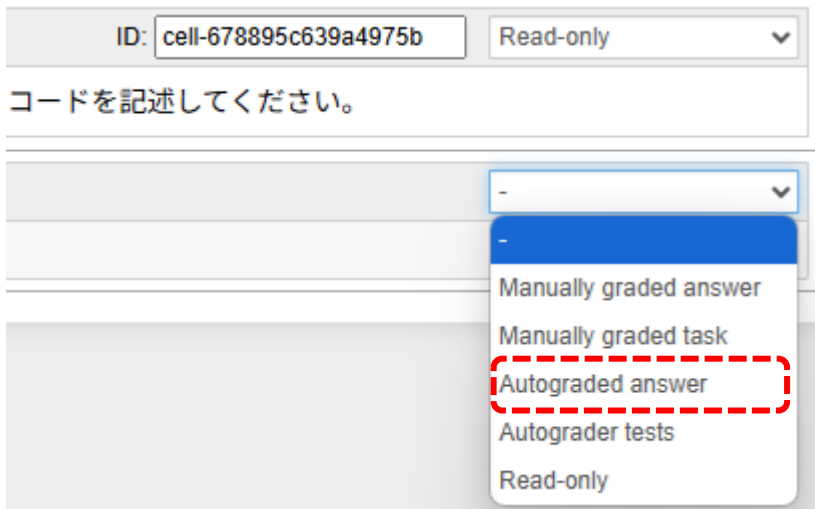
# 課題の作成 (6/16)

- 先程のセルを実行（次のセルが作成される）。

The screenshot shows a Jupyter Notebook interface. At the top, there is a menu bar with 'File', 'Edit', and 'View'. Below it is a toolbar with various icons. The 'Run' button, represented by a right-pointing triangle, is highlighted with a red dashed box. Above this button, the text '① クリックしてセルを実行' (Click to execute the cell) is written in red. To the right of the toolbar, there is a 'Trusted' status indicator and a 'LC\_wrapper' button. Below the toolbar, there is a text area containing the instruction: '引数nの2乗を返す関数squareを定義します。「raise NotImplementedError()」を削除してからコードを記述してください。' (Define a function square that returns the square of argument n. Delete 'raise NotImplementedError()' and then write the code). Below the text area, there is a code input field with the prompt 'In [ ]:' and a dropdown menu. Below the code input field, the text '② 次のセルが作成される' (The next cell is created) is written in red.

# 課題の作成 (7/16)

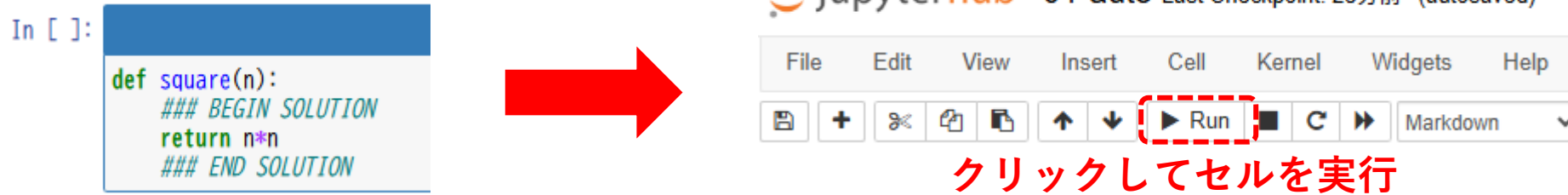
- セルの右上のリストボックスをクリックした後、「Autograded answer」を選択。



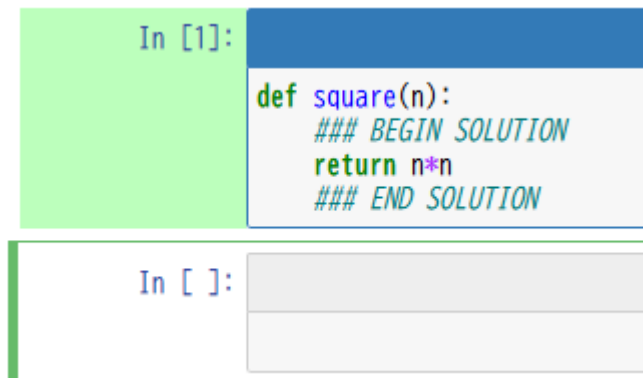
① リストボックスから「Autograded answer」を選択

# 課題の作成 (8/16)

- セルの中に以下のようなコードを記述した後、セルを実行。



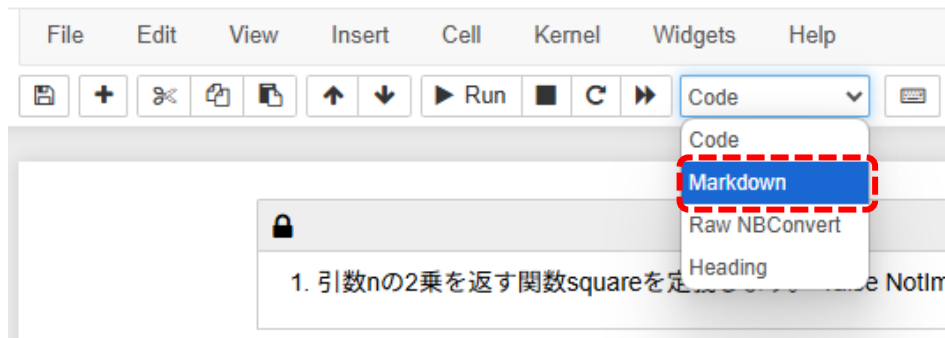
- エラーが発生しなければOK (次のセルは自動的に作成される)。





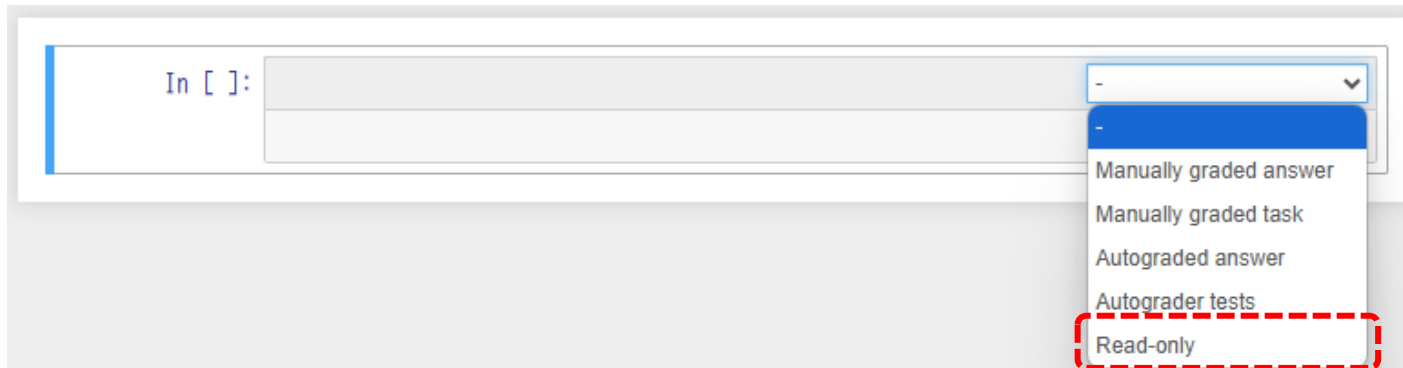
# 課題の作成 (9/16)

- 作成されたセルをクリックした後、画面上部のリストボックスの中から「Markdown」を選択。



リストボックスから  
「Markdown」を選択

- セルの右上にあるリストボックスの中から「Read-only」を選択。

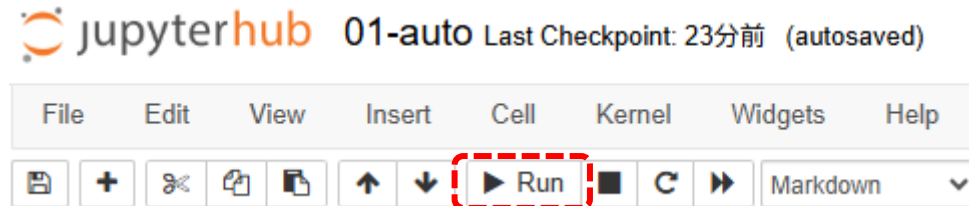


リストボックスから  
「Read-only」を選択

# 課題の作成 (10/16)

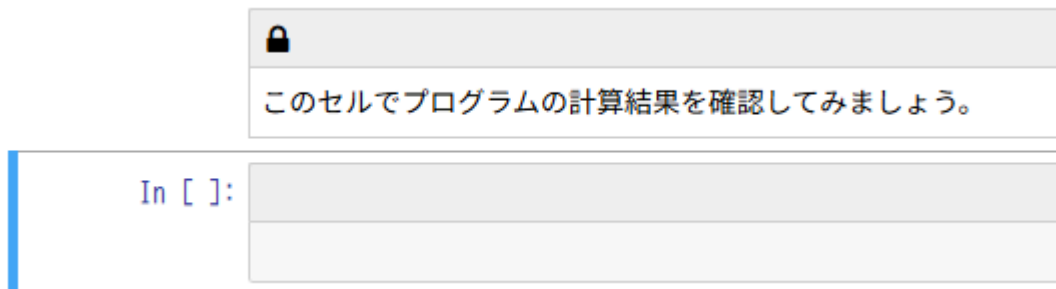
- 先程のセルの中に以下のような文章を入力した後、セルを実行。

このセルでプログラムの計算結果を確認してみましょう。



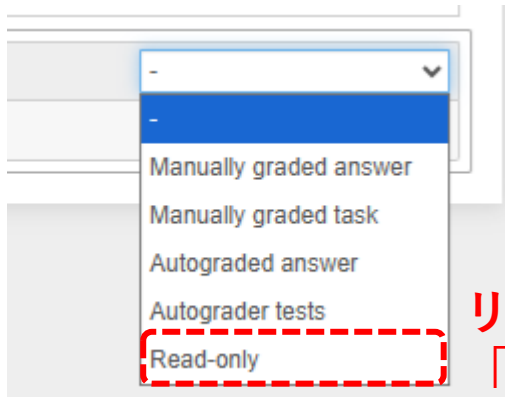
クリックしてセルを実行

- 次のセルが自動的に作成される。



# 課題の作成 (11/16)

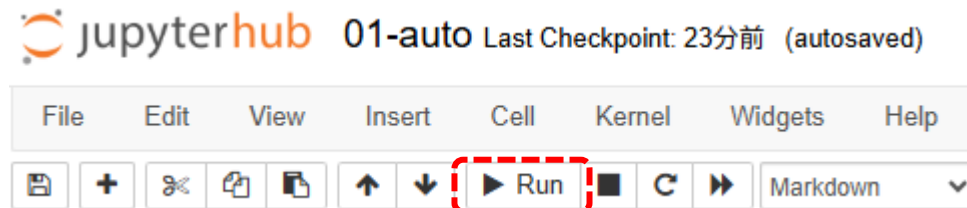
- セルの右上にあるリストボックスの中から「Read-only」を選択。



リストボックスから  
「Read-only」を選択

- 作成されたセルの中に以下のようなコードを入力した後、セルを実行。

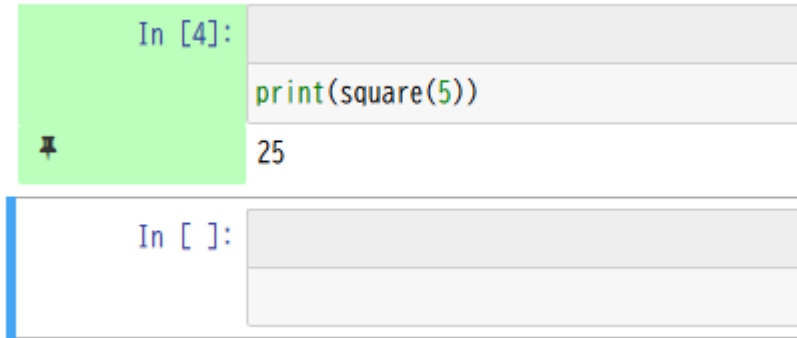
```
print(square(5))
```



クリックしてセルを実行

# 課題の作成 (12/16)

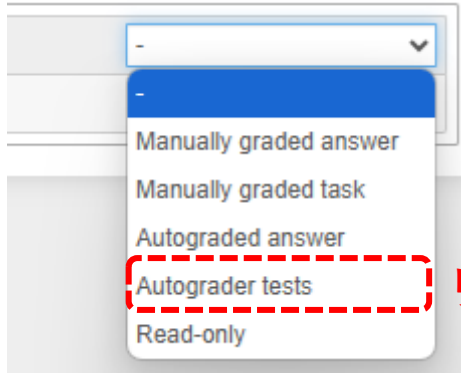
- 実行結果が表示されたらOK（次のセルが自動的に作成される）。



The image shows a Jupyter Notebook interface. The top cell, labeled 'In [4]:', contains the code `print(square(5))`. Below the code, the output '25' is displayed. The cell has a green background and a small icon on the left. Below this cell, a new empty cell is visible, labeled 'In [ ]:', with a blue background and a small icon on the left.

# 課題の作成 (13/16)

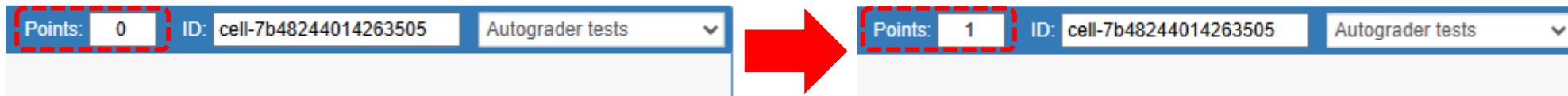
- 作成されたセルの右上のリストボックスをクリックした後、「Autograder tests」を選択。



リストボックスから「Autograder tests」を選択

- セルに評点の項目が追加されるので、評点を1点に設定。

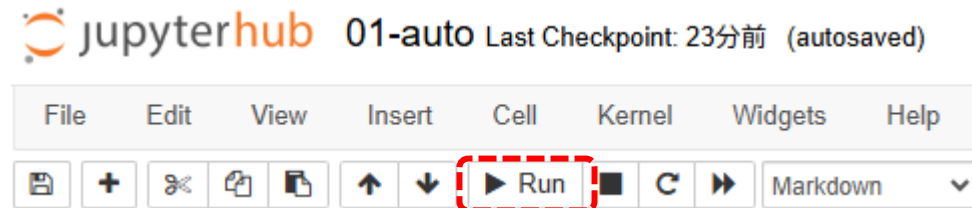
評点を1点に設定



# 課題の作成 (14/16)

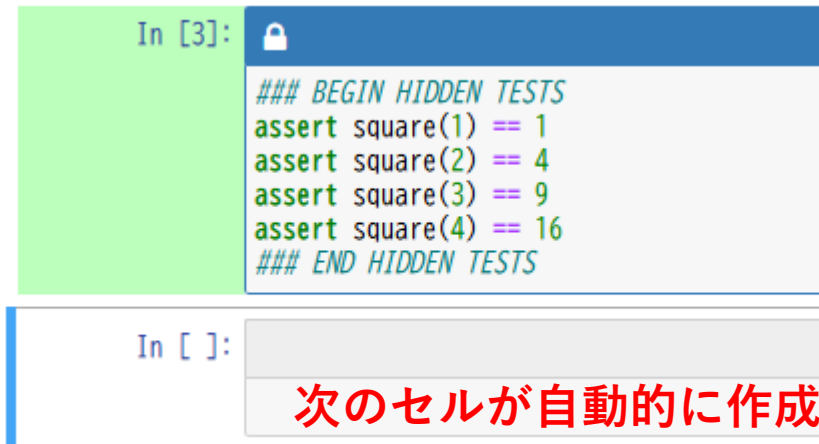
- セルの中に以下のようなコードを記述した後、セルを実行。

```
### BEGIN HIDDEN TESTS
assert square(1) == 1
assert square(2) == 4
assert square(3) == 9
assert square(4) == 16
### END HIDDEN TESTS
```



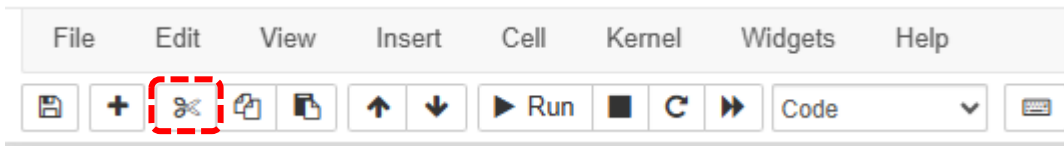
クリックしてセルを実行

- 次のセルが自動的に作成される。



# 課題の作成 (15/16)

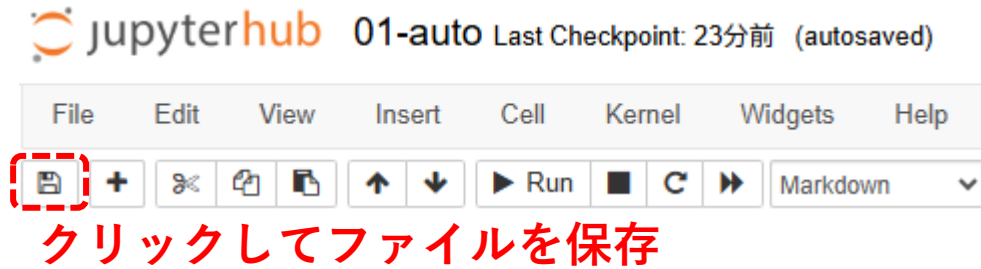
- 末尾のセルは不要なので削除。
  - 末尾のセルが選択されているときに、画面上部のハサミのアイコンをクリック。



クリックしてセルを削除

# 課題の作成 (16/16)

- 画面左上のディスクのアイコンをクリックして、ファイルを保存。



- タブを削除していき、ホームに戻る。

