

講義・演習環境アプリケーションハンズオン — 2. nbgraderを用いた課題の自動採点 —

2025年4月23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。

自動採点の演習内容

- 教師としてログインして課題を作成・公開。
- いったんログアウト。
- 学生としてログインして課題を取得、解答を記入、提出。
- いったんログアウト。
- 教師としてログインして提出物を回収・採点、フィードバックを公開。
- いったんログアウト。
- 学生としてログインしてフィードバックを取得。

※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。

課題の作成・公開（教師）

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。
 - 最初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**

user102t

.....

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](#)






demo moodle Home ダッシュボード **マイコース**   教1  編集モード 

マイコースに移動

ダッシュボード

タイムライン

次の7日  日付で並べ替える 



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。

The image shows a two-step process for accessing MCJ-CloudHub. On the left, the 'demo moodle' 'マイコース' (My Courses) page is shown. Under 'コース概要' (Course Overview), there are filters and a list of courses. One course, 'コース102' (Course 102), is highlighted with a red dashed box. A red arrow points from this course to the right-hand screenshot. The right-hand screenshot shows the 'コース102' (Course 102) page. Under the '一般' (General) section, there is a list of tools. The 'MCJ-CloudHub' tool is highlighted with a red dashed box. Red text next to it says 'クリックしてMCJ-CloudHubにアクセス' (Click to access MCJ-CloudHub).

demo moodle 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 ▾

コース102
カテゴリ1

クリックして
コースに移動

コース102

コース 設定 参加者 評価 レポート さらに ▾

▽ 一般 すべてを折りたたむ

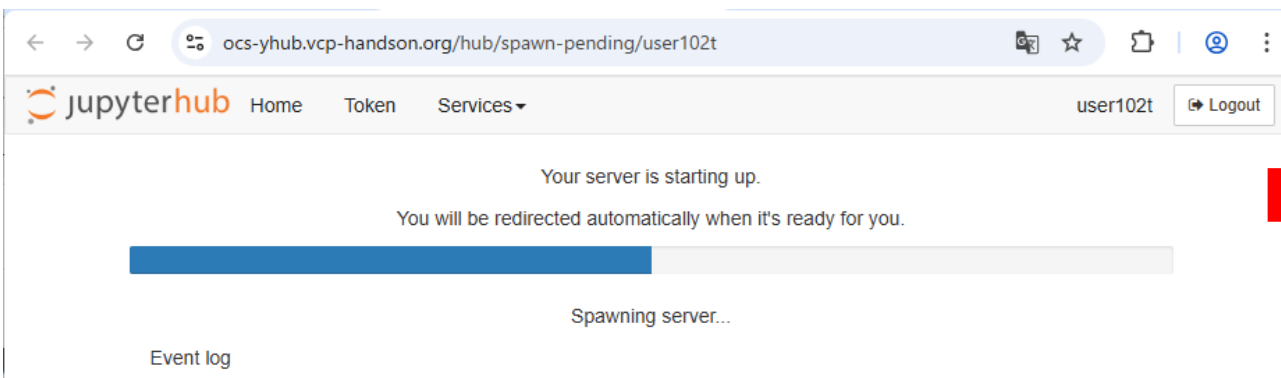
アナウンスメント

MCJ-CloudHub

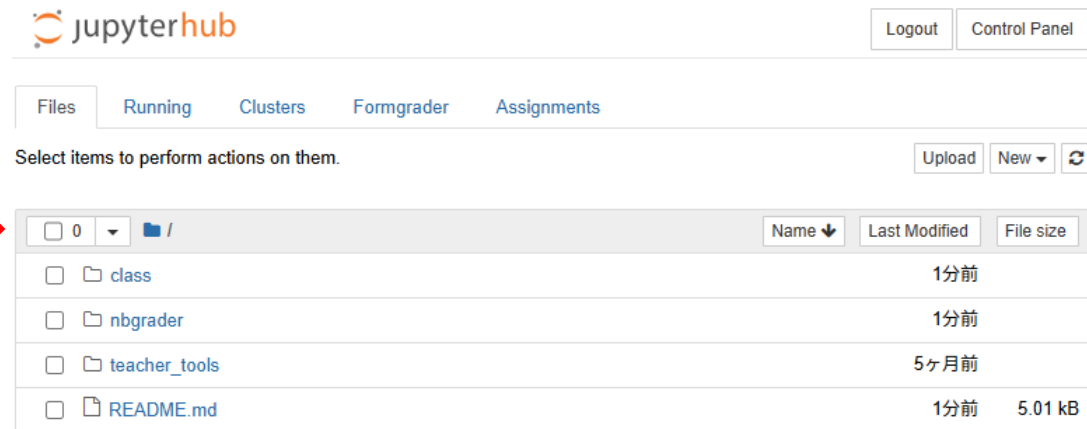
クリックしてMCJ-CloudHubに
アクセス

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

課題の作成 (1/24)

- 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

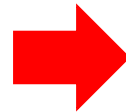
クリックして課題一覧画面に移動

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
assign01	None	released						1
+ Add new assignment...								

クリックして課題の登録を開始

課題の作成 (2/24)

- 課題の名称（今回はassign02）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（assgin02）をクリック。

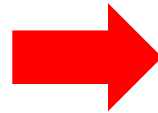
Add New Assignment ×

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

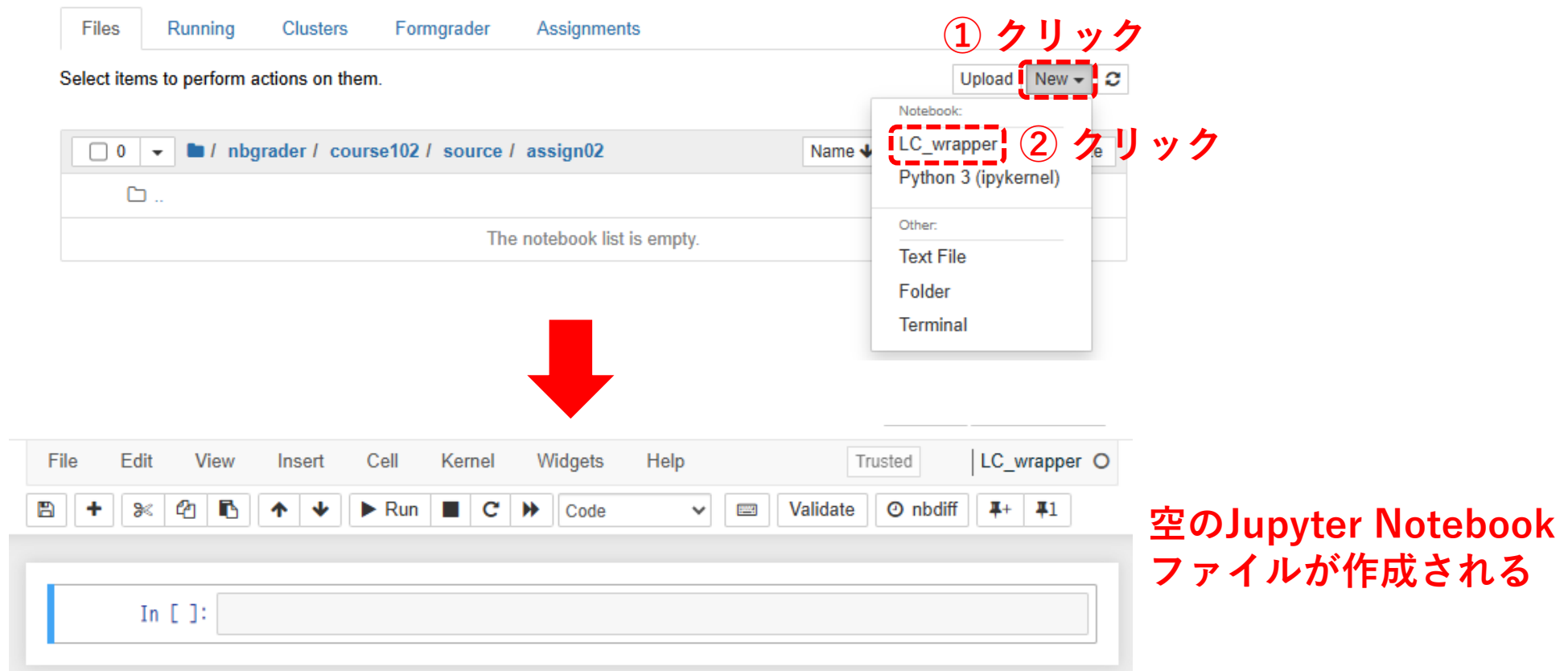


Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
assign01	None	released						1
assign02	None	draft						0

課題名をクリック

課題の作成 (3/24)

- 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。



The screenshot illustrates the process of creating a new Jupyter Notebook in the nbgrader interface. The top navigation bar includes tabs for 'Files', 'Running', 'Clusters', 'Formgrader', and 'Assignments'. Below this, a message states 'Select items to perform actions on them.' The main area shows a file browser for the path '/ nbgrader / course102 / source / assign02'. A red arrow points from the 'new' button in the top right to the 'LC_wrapper' notebook in the dropdown menu. The notebook is created with the name 'LC_wrapper' and the kernel 'Python 3 (ipykernel)'. The bottom part of the image shows the Jupyter Notebook interface with the title 'LC_wrapper' and a code cell containing 'In []:'.

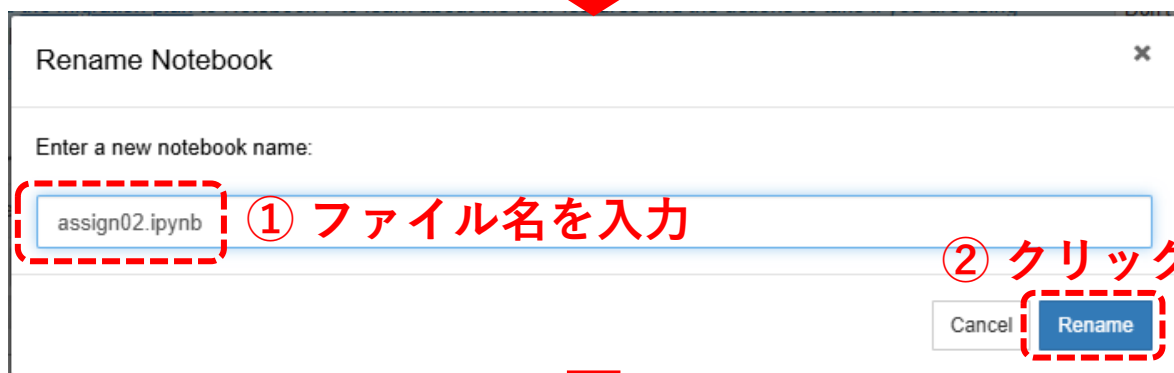
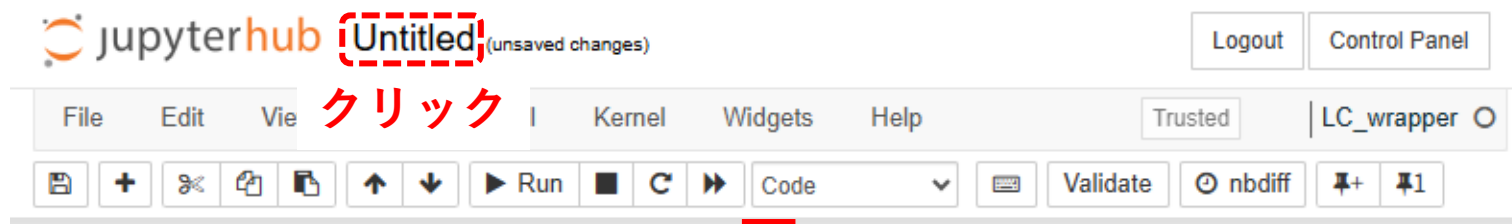
① クリック

② クリック

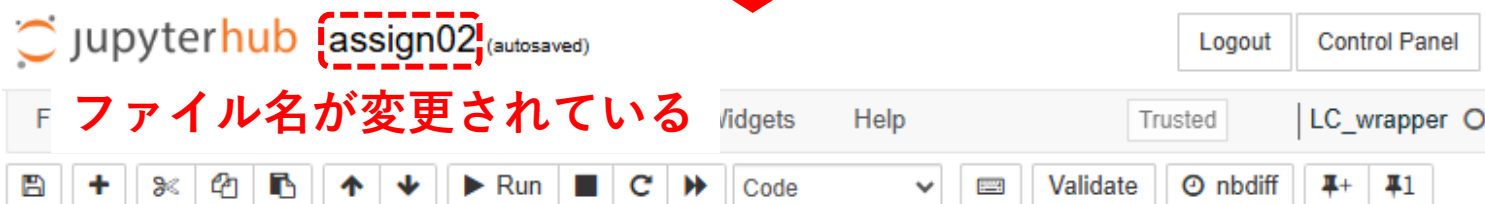
空のJupyter Notebook
ファイルが作成される

課題の作成 (4/24)

- 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（assign02.ipynb）を入力して「Rename」をクリック。

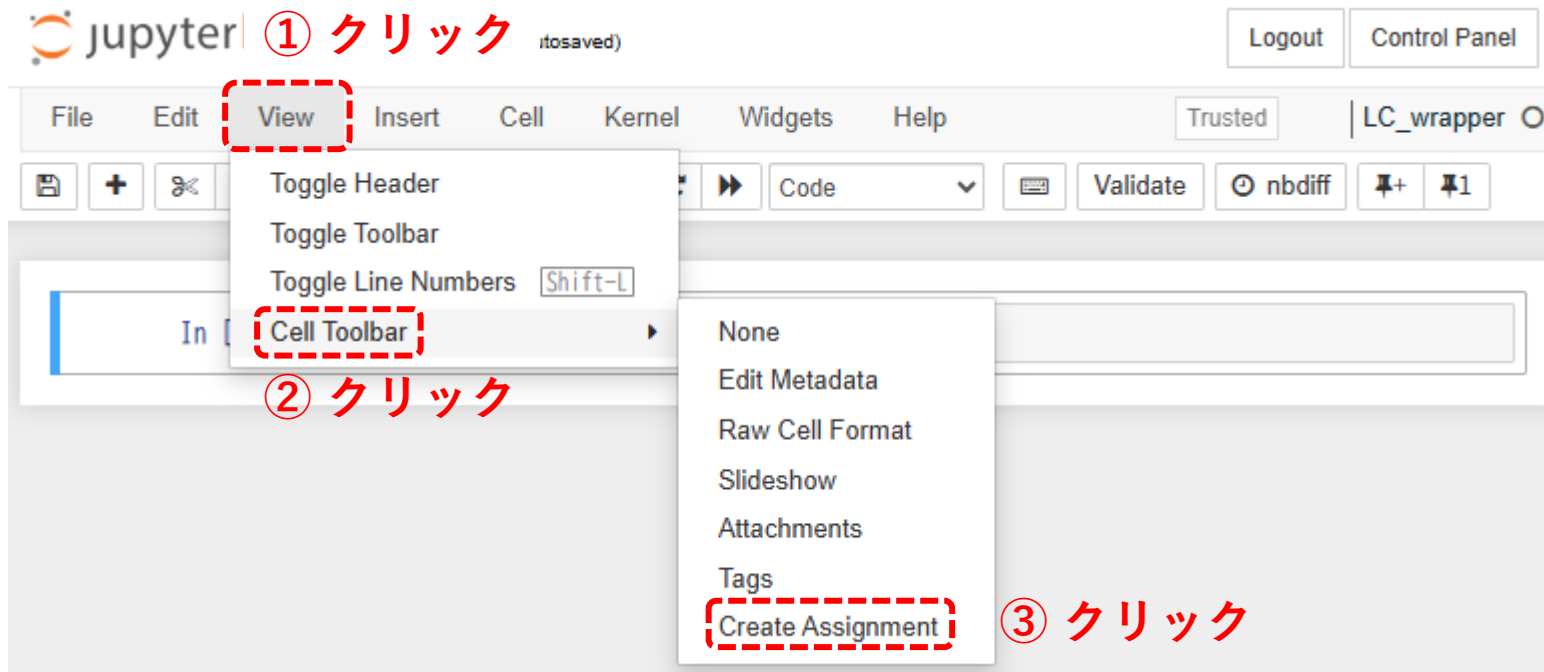


今回は課題名とファイル名が同じですが、同じである必要はありません。実際には、異なる名前にしたほうが分かりやすいと思います。



課題の作成 (5/24)

- 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。

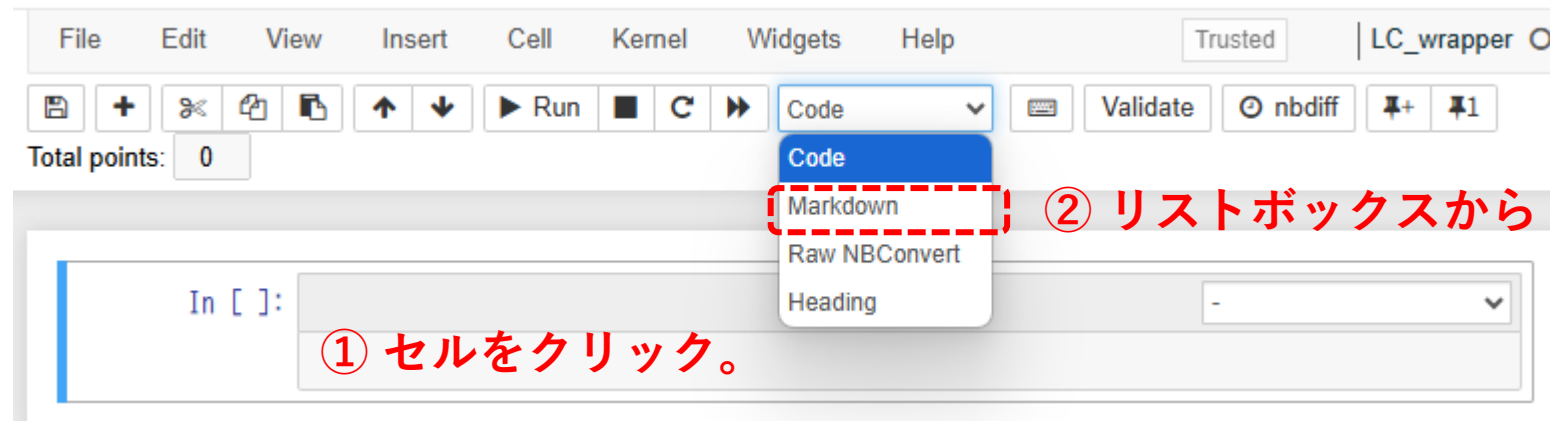


セルの右上にリストボックスが表示



課題の作成 (6/24)

- 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。

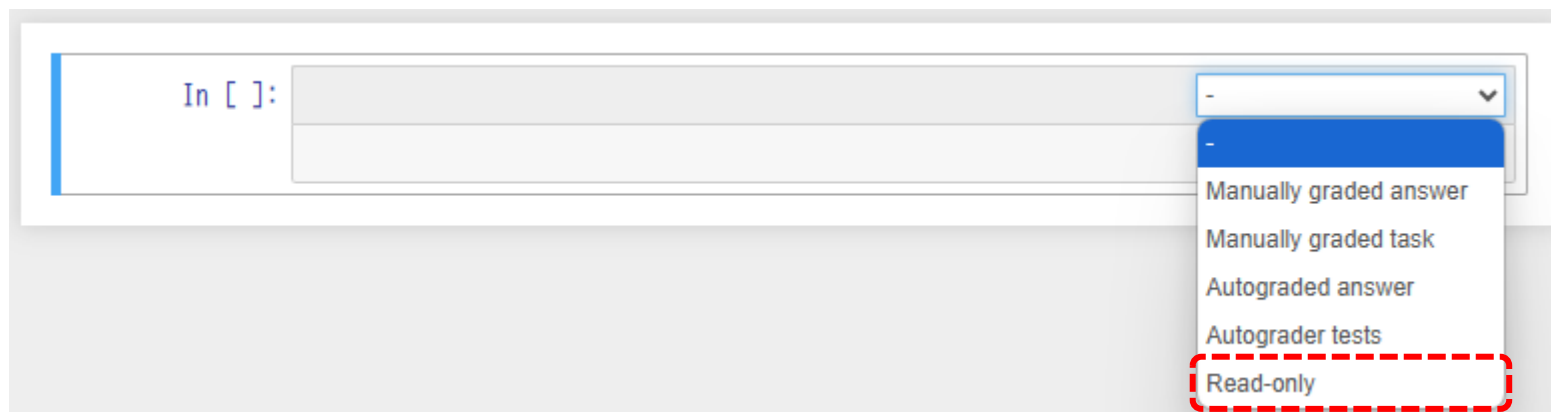


① セルをクリック。

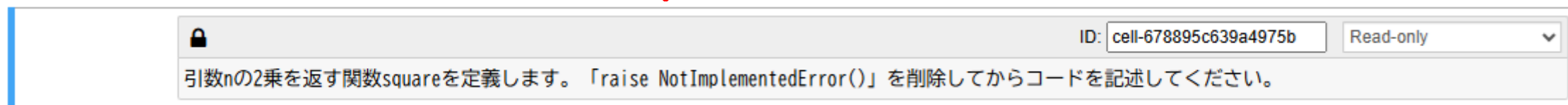
② リストボックスから「Markdown」を選択。

課題の作成 (7/24)

- 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



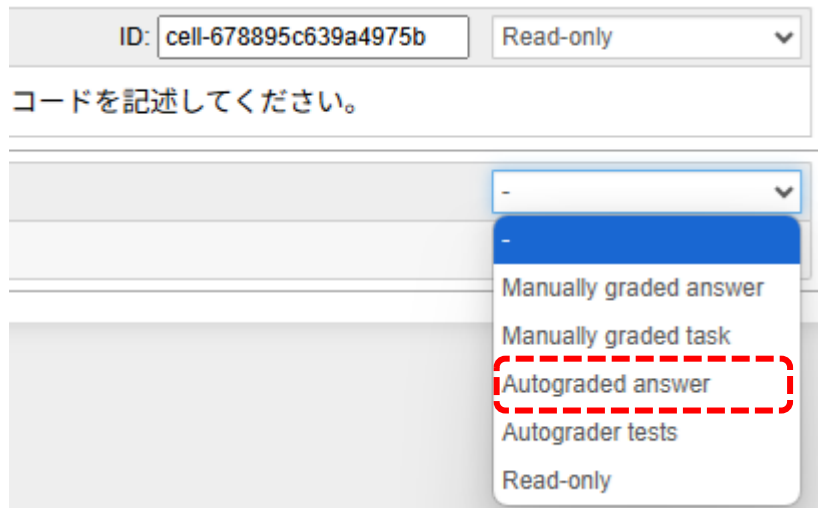
課題の作成 (8/24)

- 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. A red dashed box highlights the 'Run' button (a play icon) in the toolbar, with the text '① クリックしてセルを実行' (Click to execute the cell) in red above it. To the right of the toolbar, there is a 'Trusted' status indicator and a 'LC_wrapper' button. Below the toolbar, there is a cell with a lock icon, an ID field containing 'cell-678895c639a4975b', and a 'Read-only' dropdown menu. The cell contains the text: '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 cell, there is a prompt 'In []:' followed by a text input field. A red dashed box highlights the text input field, with the text '② 次のセルが作成される' (The next cell will be created) in red above it.

課題の作成 (9/24)

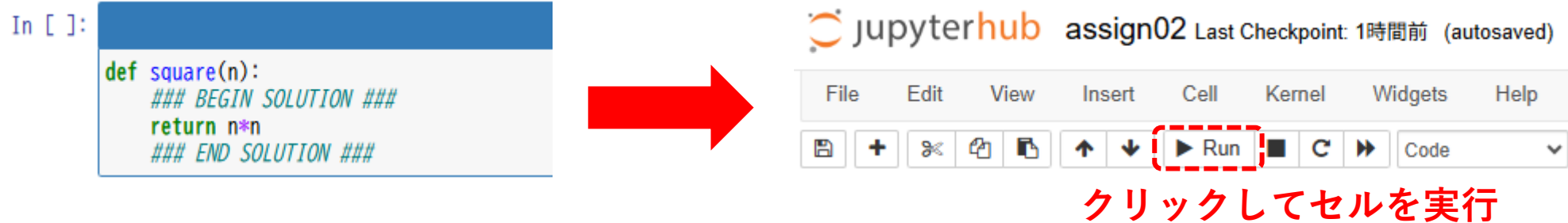
- セルの右上のリストボックスをクリックした後、「Autograded answer」を選択。



① リストボックスから「Autograded answer」を選択

課題の作成 (10/24)

- 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。



```
In [ ]:
```

```
def square(n):  
    ### BEGIN SOLUTION ###  
    return n*n  
    ### END SOLUTION ###
```

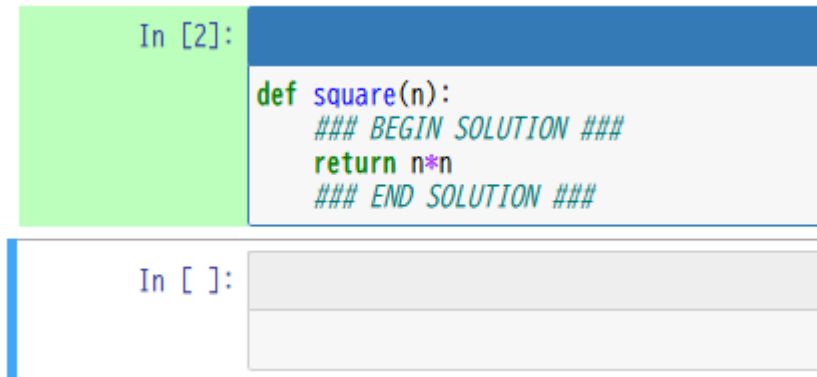
jupyterhub assign02 Last Checkpoint: 1時間前 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run

クリックしてセルを実行

- 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)。



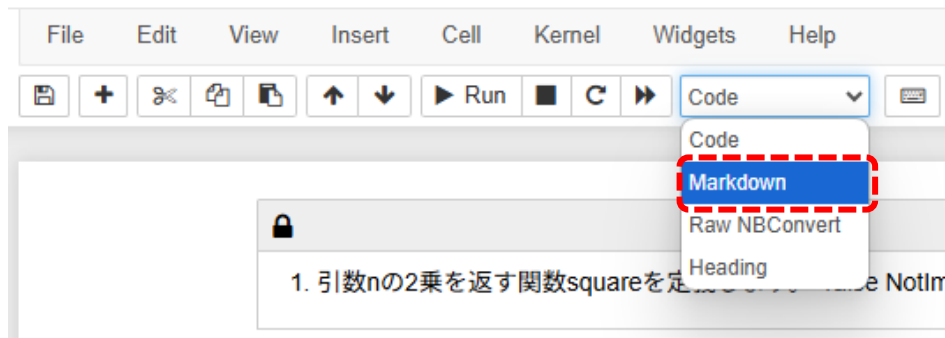
```
In [2]:
```

```
def square(n):  
    ### BEGIN SOLUTION ###  
    return n*n  
    ### END SOLUTION ###
```

```
In [ ]:
```

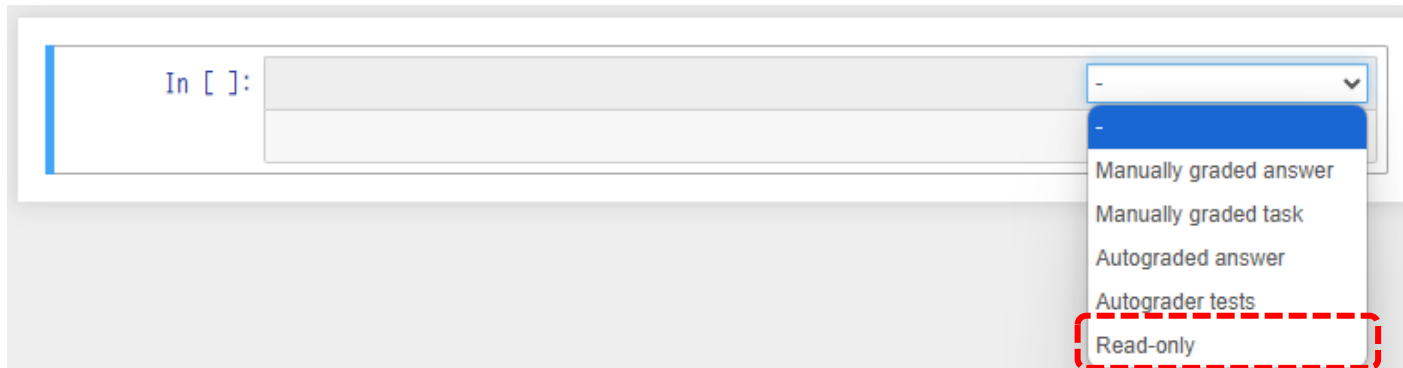
課題の作成 (11/24)

- 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。

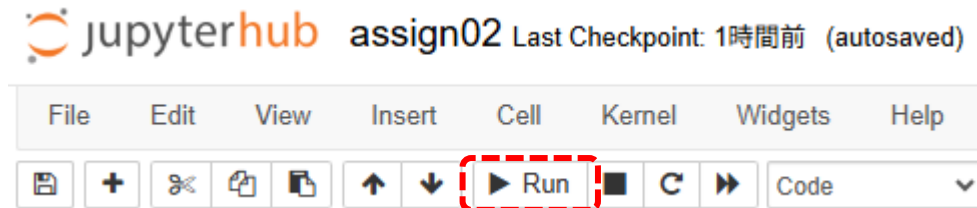


リストボックスから
「Read-only」を選択

課題の作成 (12/24)

- 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。

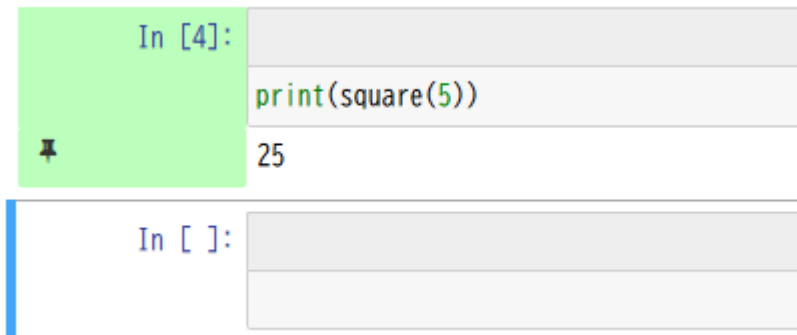


課題の作成 (13/24)

- 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。

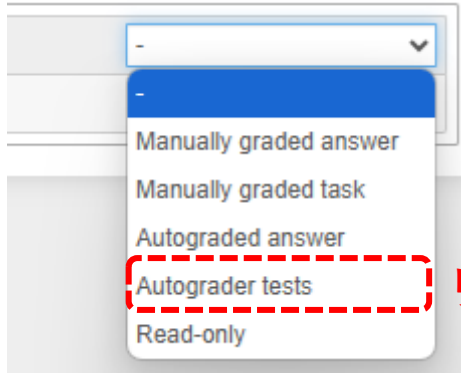


- 実行結果が表示されたらOK（次のセルが自動的に作成される）。



課題の作成 (14/24)

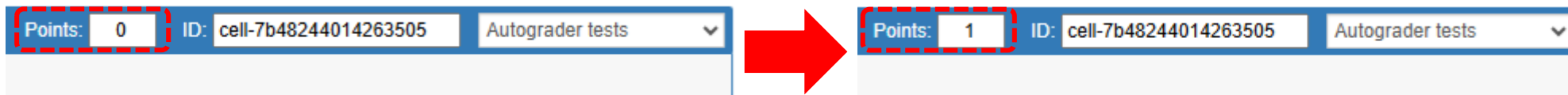
- 作成されたセルの右上のリストボックスをクリックした後、「Autograded answer」を選択。



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。

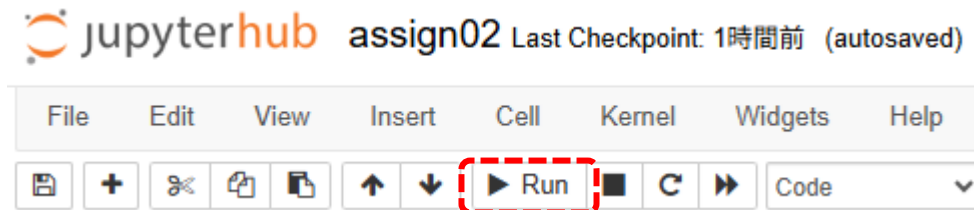
評点を1点に設定



課題の作成 (15/24)

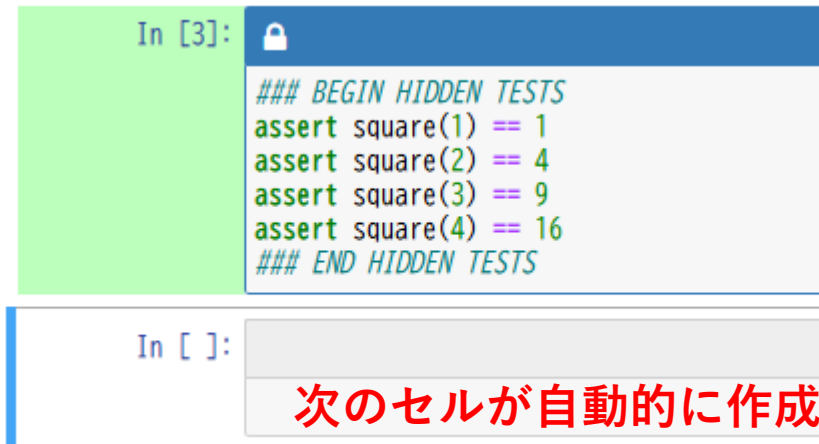
- セルの中に以下のようなコードを記述した後、セルを実行。

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



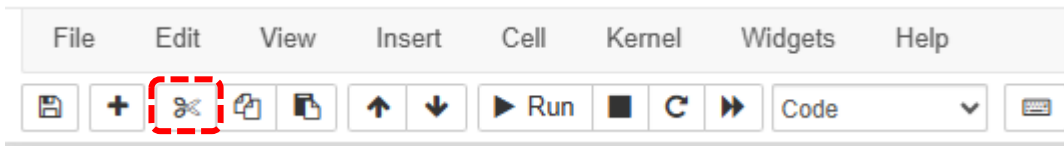
クリックしてセルを実行

- 次のセルが自動的に作成される。



課題の作成 (16/24)

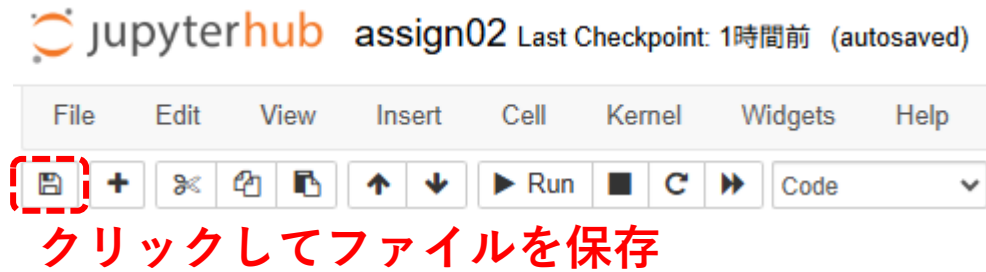
- 末尾のセルは不要なので削除。
 - 末尾のセルが選択されているときに、画面上部のハサミのアイコンをクリック。



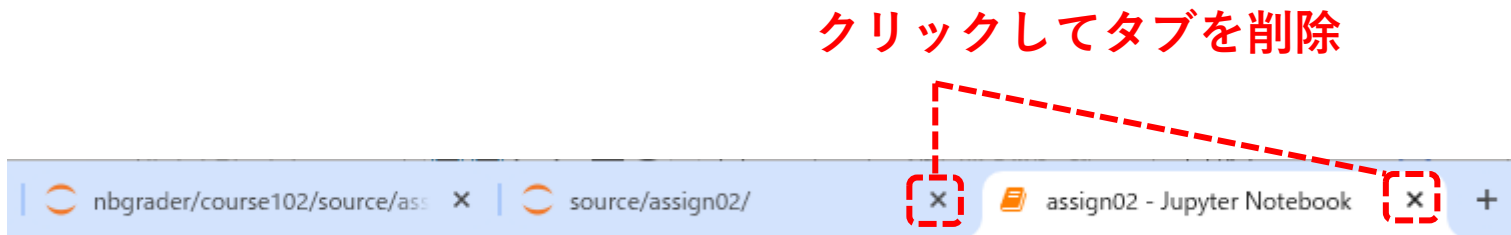
クリックしてセルを削除

課題の作成 (17/24)

- 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。

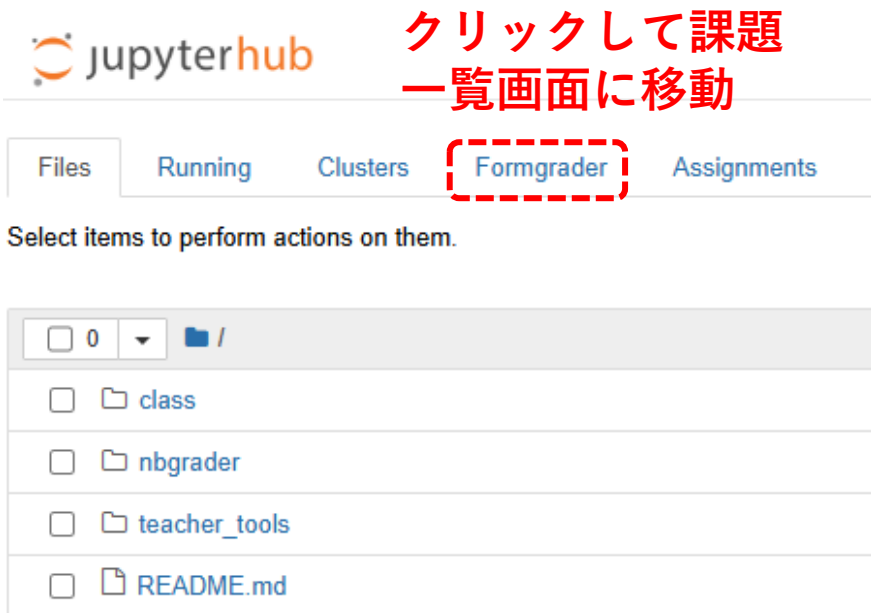


- タブを削除していき、ホームに戻る。



課題の作成 (18/24)

- 画面上部の「Formgrader」をクリックして課題一覧画面に移動。
- 課題「assign02」の行にある「Generate」のアイコンをクリック。

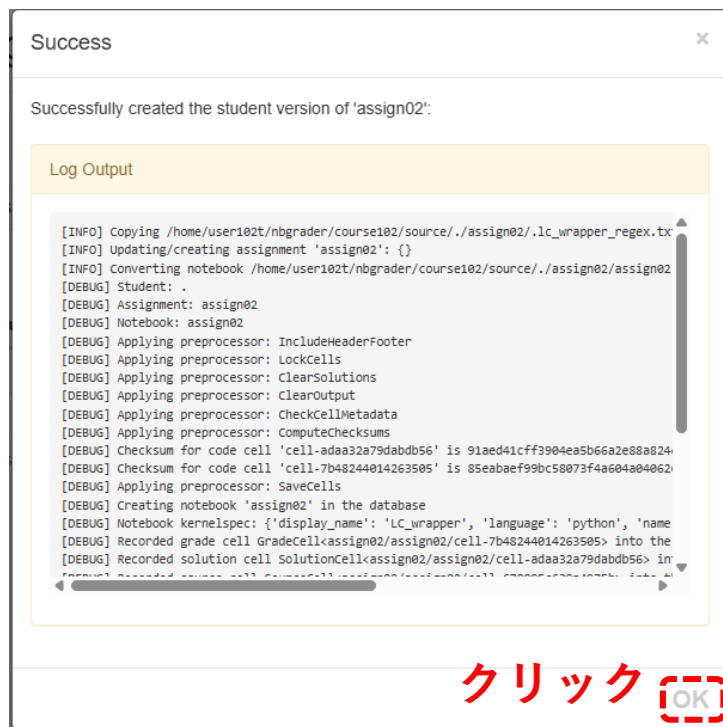


Instructions (click to expand)									
Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	
assign01	None	released						1	
assign02	None	draft						0	
+ Add new assignment...									

Clicking the 'Generate' icon for 'assign02' (highlighted with a red dashed box) will create a distribution file.

課題の作成 (19/24)

- 「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。



Instructions (click to expand)

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
assign01	None	released						1
assign02	None	draft						0
+ Add new assignment...								

クリックして配布用ファイルを表示

クリックして配布用ファイルを表示

課題の作成 (20/24)

- 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 (21/24)

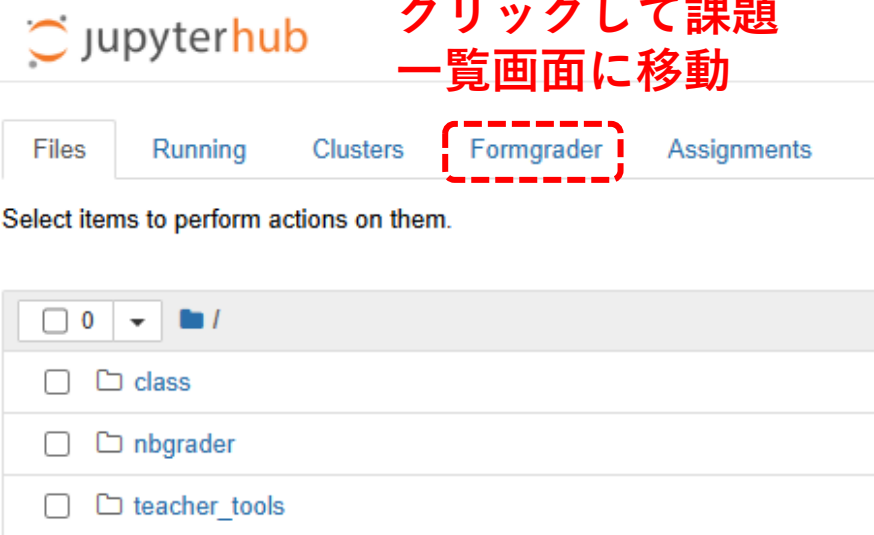
- タブを削除していき、ホームに戻る。

クリックしてタブを削除



- 「assign01」の行の「Release」のアイコンをクリック。

クリックして課題
一覧画面に移動



Instructions (click to expand)									
Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	
assign01	None	released						1	
assign02	None	draft						0	
+ Add new assignment...									

クリックして課題の公開を開始

課題の作成 (22/24)

- 「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						1
assign02	None	released						0
+ Add new assignment...								

回収アイコンや提出数が表示される

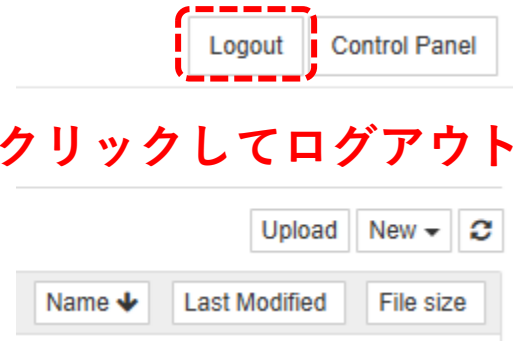
- タブを削除して、ホームに戻る。

クリックしてタブを削除



課題の作成 (23/24)

- 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。



課題の作成 (24/24)

- 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

課題の取得・記入・提出（学生）

学生としてMoodleにログイン

- 学生用のアカウントを使用してMoodleにログイン。

**割り当てられた学生用
アカウントでログイン**

user102s

.....

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](#)



demo moodle Home ダッシュボード **マイコース**   学1 [編集モード](#)

マイコースに移動

ダッシュボード

タイムライン

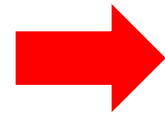
次の7日 [日付で並べ替える](#) [活動タイプまたは活動名で検索する](#)



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。



コース102

コース 参加者 評価 コンビテンシ

▽ 一般

すべてを折りたたむ



アナウンスメント

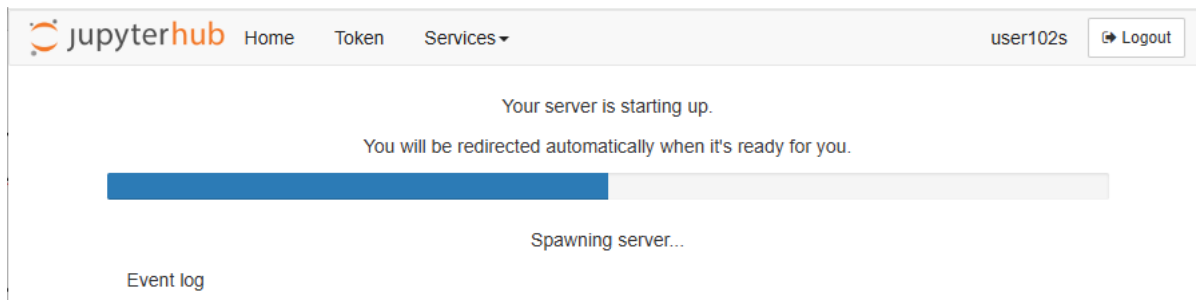


MCJ-CloudHub

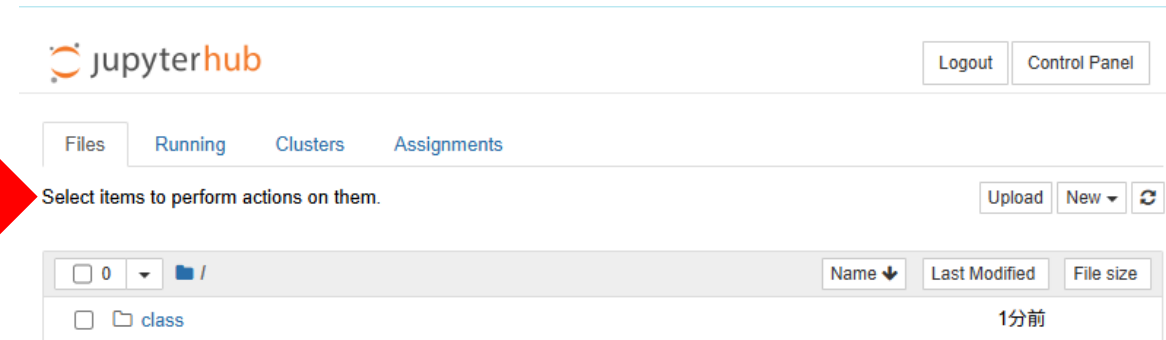
クリックしてMCJ-CloudHubに
アクセス

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。



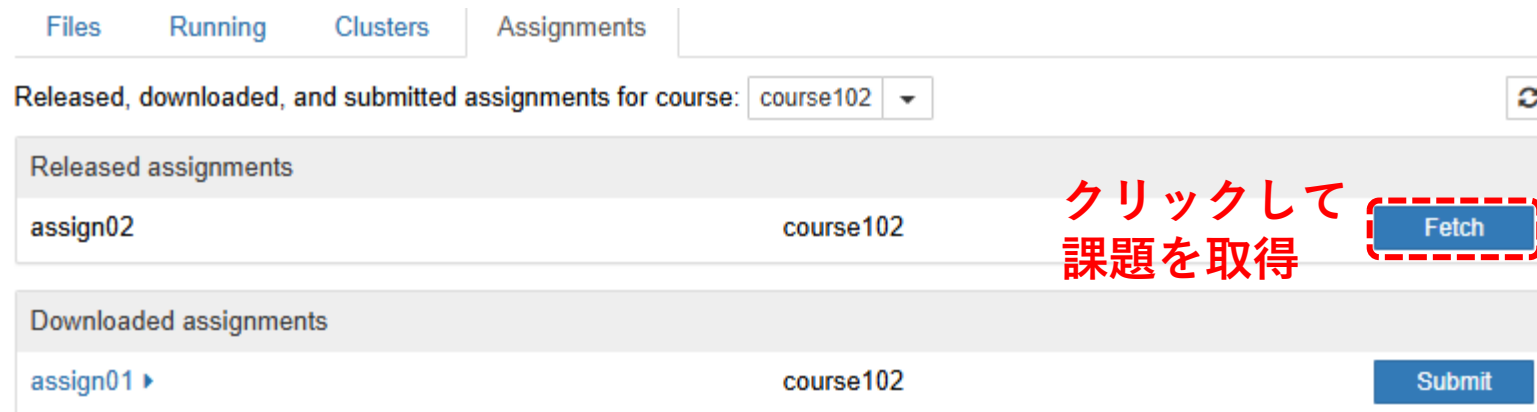
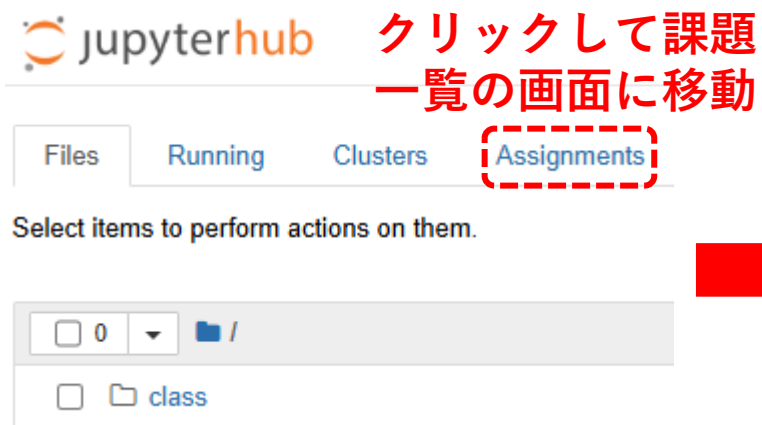
Jupyter環境が起動するまで待機



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。
- 「assign02」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「assign02」が表示されるので、課題名をクリック。

Released assignments

There are no assignments to fetch.

Downloaded assignments

assign01 ▶	course102	Submit
assign02 ▶	course102	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、「assign02」をクリック。

Downloaded assignments

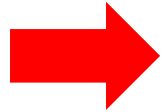
assign01 ▶	course102	Submit
assign02 ▼	course102	Submit
assign02		Validate

ファイル名をクリックして編集を開始

解答の作成 (1/4)

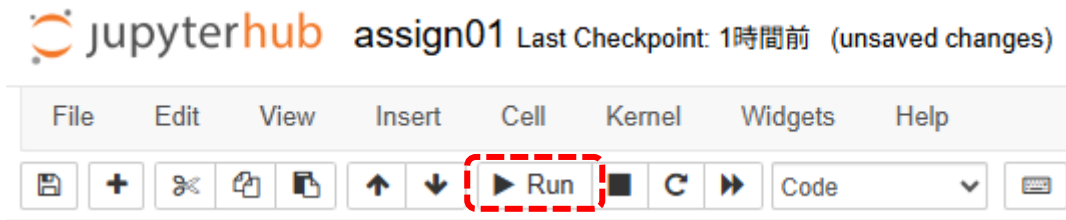
- プログラムを記入する課題の場合、「raise NotImplementedError()」を削除した後に解答を記入。

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



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

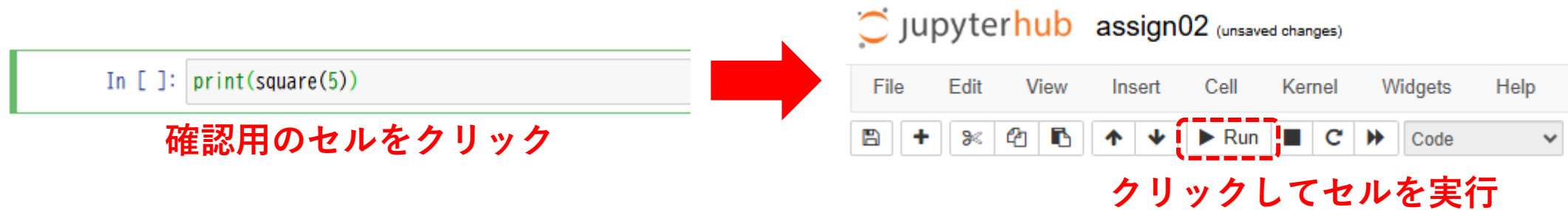
- プログラムを記入できたらセルを実行。



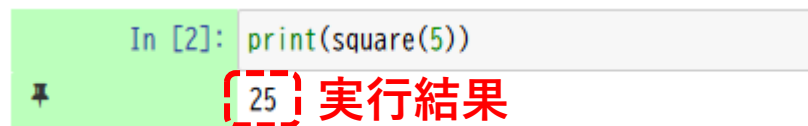
クリックしてセルを実行

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。

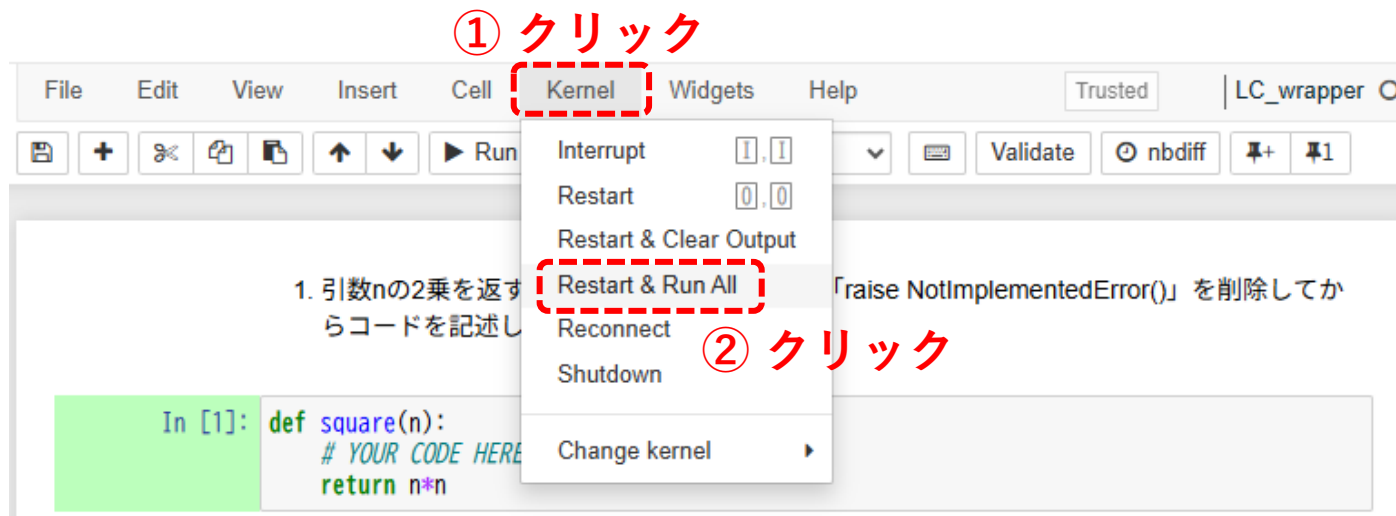


- 実行結果の表示に基づいて、解答の正誤を判断。
 - 教師側はそうように確認用のセルを作成すること。

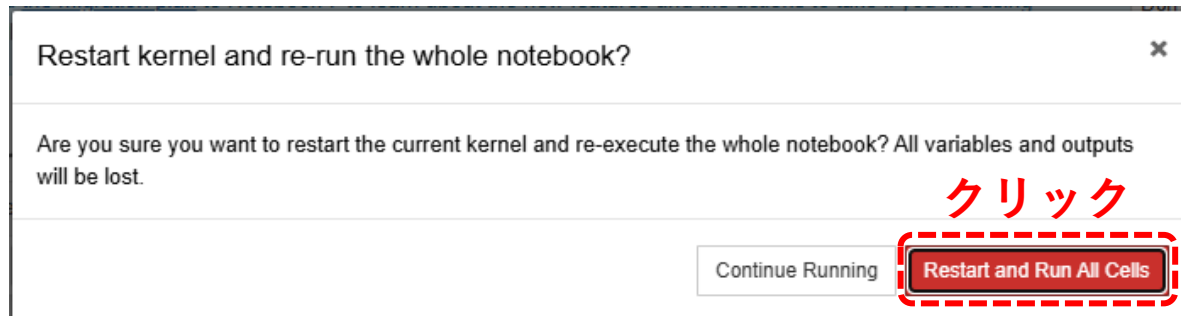


解答の作成 (3/4)

- 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。

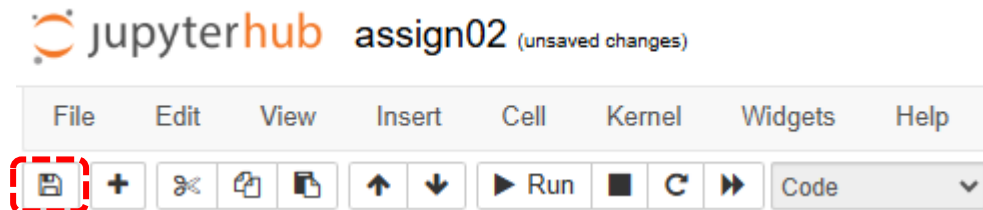


- 「Restart and Run All Cells」をクリック。エラーが発生しなければOK。



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



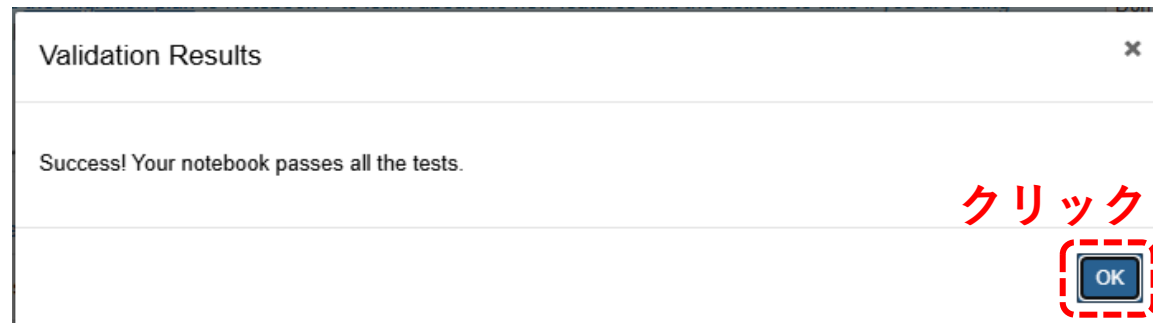
解答の提出 (1/4)

- 課題ファイルについて「Validate」をクリック。

Downloaded assignments		
assign01 ▶	course102	Submit
assign02 ▼	course102	Submit
assign02		Validate

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



解答の提出 (2/4)

- 課題「assign02」について、「Submit」をクリック。

Downloaded assignments		
assign01 ▶	course102	Submit
assign02 ▼	course102	Submit
assign02		Validate

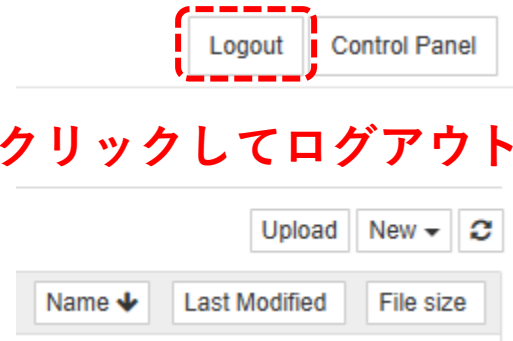
クリックして
解答を提出

- 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	course102	Fetch Feedback
2025-04-21 05:39:45 JST (view feedback)		
assign02	course102	Fetch Feedback
2025-04-21 17:58:50 JST		

解答の提出 (3/4)

- 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。



解答の提出 (4/4)

- 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

提出物の回収・採点（教師）

教師としてMoodleにログイン

- 教師用のアカウントを使用してMoodleにログイン。

割り当てられた教師アカウントでログイン

user102t

.....

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](#)



demo moodle Home ダッシュボード **マイコース**   教1 [▼](#) 編集モード 

マイコースに移動

ダッシュボード

タイムライン

次の7日 [▼](#) 日付で並べ替える [▼](#)



アクションを必要とする活動はありません。

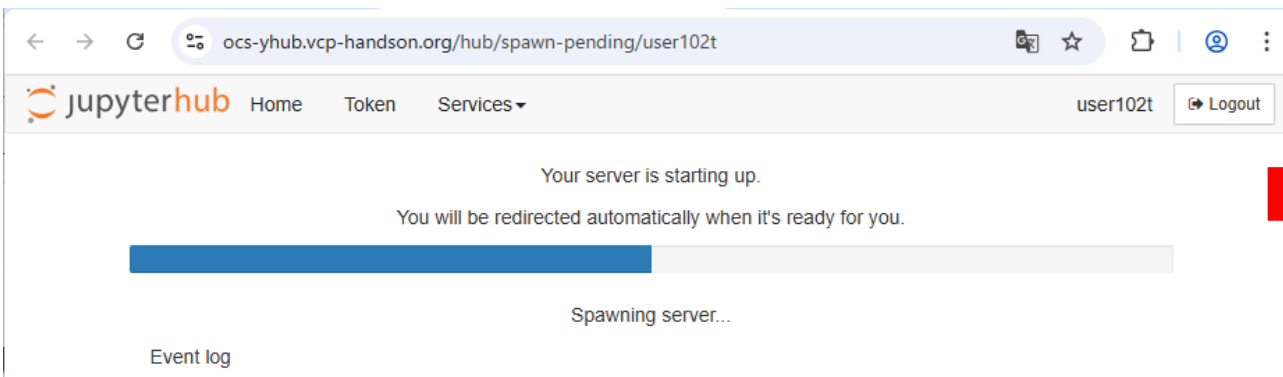
コースに移動してMCJ-CloudHubを起動

- 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。

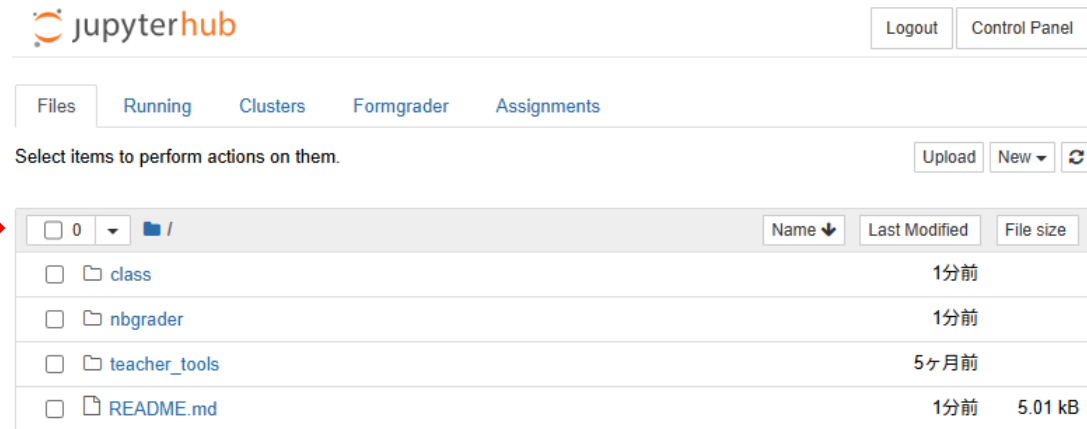
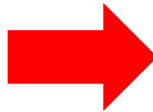
The image shows a two-step process for accessing MCJ-CloudHub from Moodle. On the left, the 'マイコース' (My Courses) page is displayed, showing a list of courses. A red dashed box highlights a course card for 'コース102' (Course 102). A red arrow points from this card to the right, where the 'コース102' (Course 102) page is shown. On this page, under the '一般' (General) section, there is a card for 'MCJ-CloudHub' with a red dashed box around it. Red text labels indicate the actions: 'クリックしてコースに移動' (Click to move to course) and 'クリックしてMCJ-CloudHubにアクセス' (Click to access MCJ-CloudHub).

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。



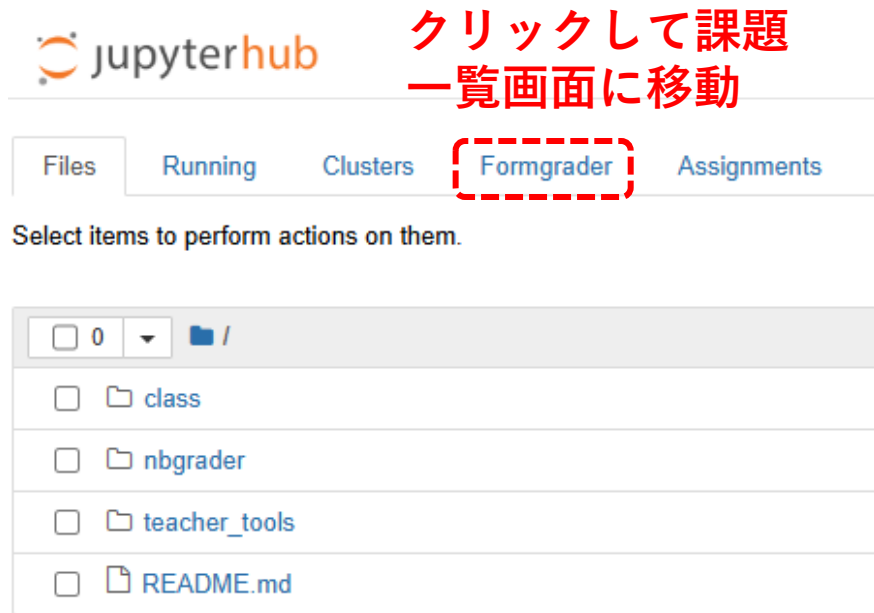
Jupyter環境が起動するまで待機



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「assign02」の行にある回収アイコンをクリック。

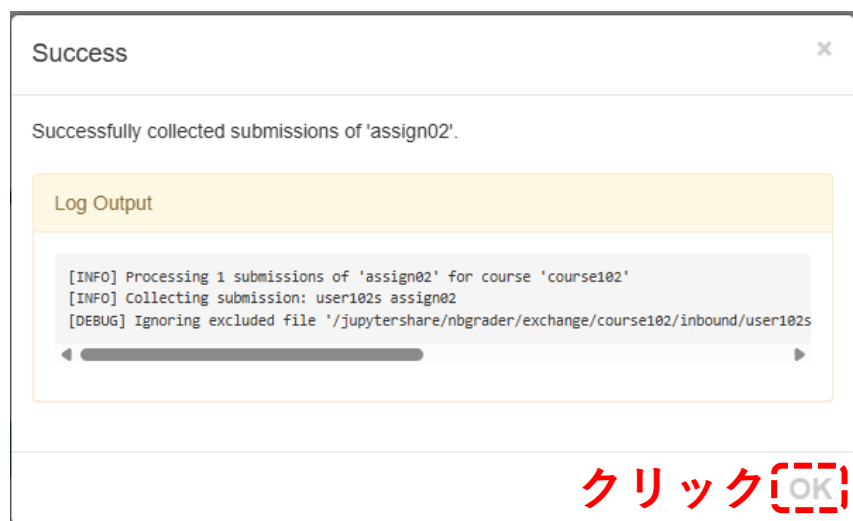


Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions
assign01	None	released						1
assign02	None	released						0
+ Add new assignment...								

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
assign02	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 (1/3)

- 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
assign02	None	released						1
+ Add new assignment...								

1
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
学生, 102	user102s	2025-04-21 17:58:50 JST	needs autograding				

クリックして自動採点を開始

課題の自動採点 (2/3)

- 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 'assign02' for student 'user102s'.

Log Output

[INFO] Copying /home/user102t/nbgrader/course102/submitted/user102s/assign02/.lc_wrapper

[INFO] Copying /home/user102t/nbgrader/course102/submitted/user102s/assign02/timestamp.

[INFO] Creating/updating student with ID 'user102s': {}

[INFO] SubmittedAssignment<assign02 for user102s> submitted at 2025-04-21 08:58:50

[INFO] Overwriting files with master versions from the source directory

[INFO] Copying /home/user102t/nbgrader/course102/source/./assign02/.lc_wrapper_regex.tx

[INFO] Sanitizing /home/user102t/nbgrader/course102/submitted/user102s/assign02/assign0

[INFO] Converting notebook /home/user102t/nbgrader/course102/submitted/user102s/assign0

[DEBUG] Student: user102s

[DEBUG] Assignment: assign02

[DEBUG] Notebook: assign02

[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': 'pyth

[DEBUG] Overwriting submitted notebook kernelspec: {'display_name': 'LC_wrapper', 'lang

[DEBUG] Applying preprocessor: OverwriteCells

[WARNING] Attribute 'checksum' for cell cell-7b48244014263505 has changed! (should be:)

クリック[OK]



Student Name	Student ID	Timestamp	Status	Score	Autograde	Generate Feedback	Release Feedback
学生, 102	user102s	2025-04-21 17:58:50 JST	graded	1 / 1	⚡	💬	✉

提出物の状態が「graded」に変化

課題の自動採点 (3/3)

- 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

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。
- 今回のセミナーでは説明を省略しますが、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。

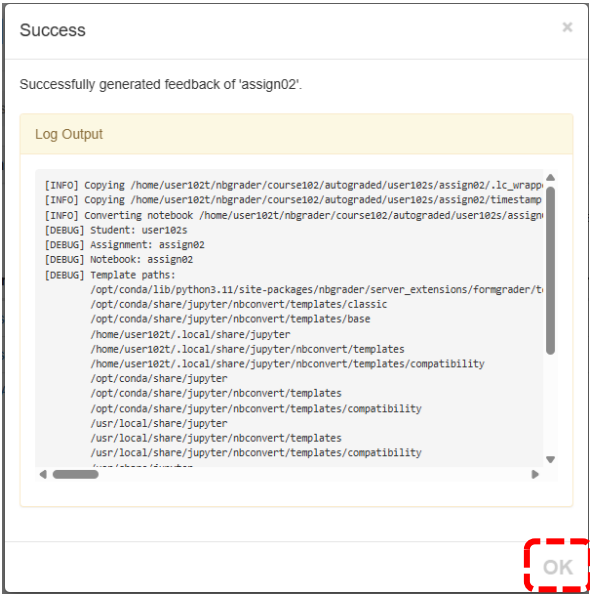
フィードバックの作成・公開（1/5）

- 課題「assign02」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		
assign02	None	released						1		

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

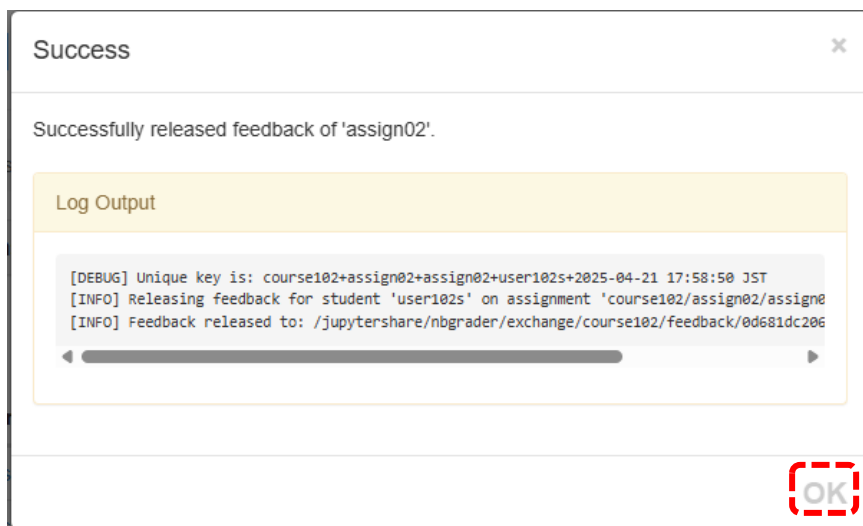
フィードバックの作成・公開 (2/5)

- 課題「assign02」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		
assign02	None	released						1		

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。



OK クリック

フィードバックの作成・公開 (3/5)

- タブを削除して、ホームに戻る。

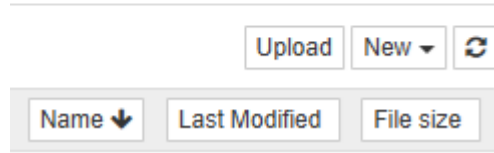
クリックしてタブを削除



- 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



フィードバックの作成・公開 (4/5)

- MCJ-CloudHubのタブを削除。

クリックしてタブを削除



フィードバックの作成・公開 (5/5)

- 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

フィードバックの取得（学生）

学生としてMoodleにログイン

- 学生用のアカウントを使用してMoodleにログイン。

**割り当てられた学生用
アカウントでログイン**

user102s

.....

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](#)



demo moodle Home ダッシュボード マイコース

マイコースに移動

ダッシュボード

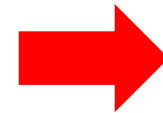
タイムライン

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。



コース102

コース 参加者 評価 コンビテンシ

一般

すべてを折りたたむ



アナウンスメント

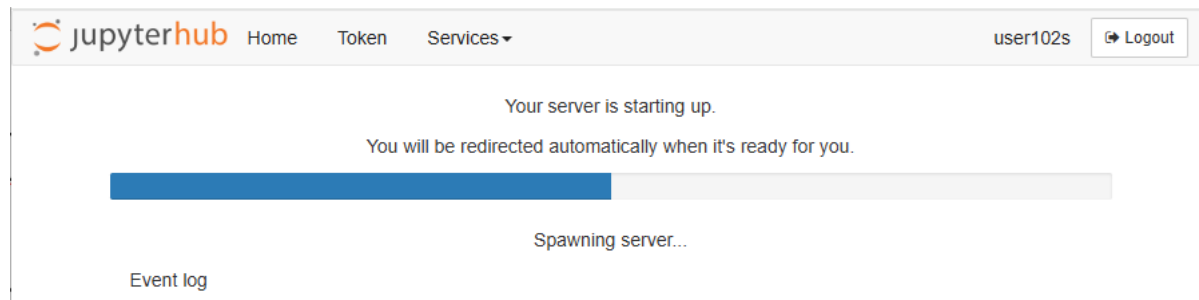


MCJ-CloudHub

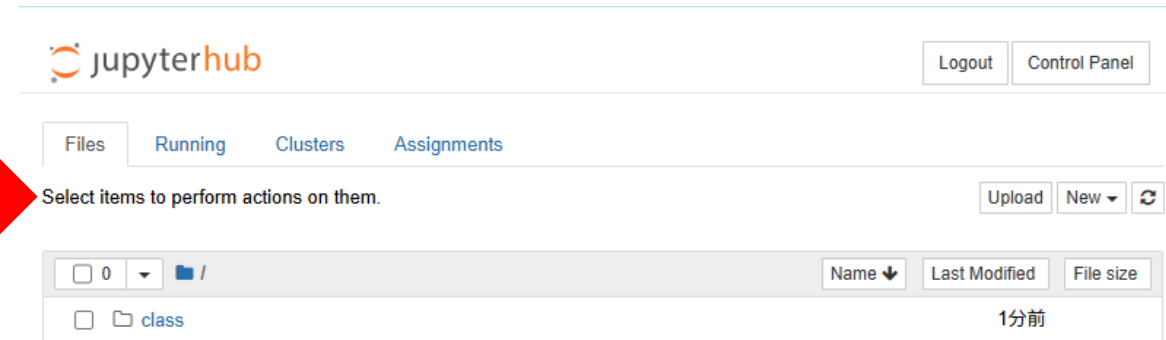
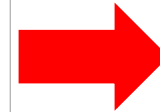
クリックしてMCJ-CloudHubに
アクセス

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

フィードバックの取得 (1/6)

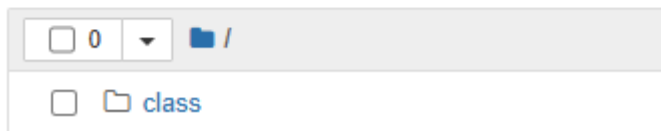
- 画面上部の「Assignments」をクリックして課題一覧画面に移動。
- 「assign02」のフィードバックが公開されているので、「Fetch feedback」をクリック。



クリックして課題一覧の画面に移動



Select items to perform actions on them.



Downloaded assignments		
assign01 ▶	course102	Submit
assign02 ▶	course102	Submit
Submitted assignments		
assign01	course102	Fetch Feedback
2025-04-21 05:39:45 JST (view feedback)		
assign02	course102	Fetch Feedback
2025-04-21 17:58:50 JST (feedback available to fetch)		

クリックしてフィードバックを取得

フィードバックの取得 (2/6)

- フィードバックが公開されている場合、「view feedback」というリンクが作成されるので、これをクリック。

Submitted assignments		
assign01	course102	Fetch Feedback
2025-04-21 05:39:45 JST (view feedback)		
assign02	course102	Fetch Feedback
2025-04-21 17:58:50 JST (view feedback)		

クリックしてフィードバックを表示

- 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	/ course102 / assign02 / feedback / 2025-04-21 17:58:50 JST
<input type="checkbox"/> ..	
<input type="checkbox"/> assign02.html	

クリックしてフィードバックを表示

フィードバックの取得 (3/6)

- フィードバックの内容を確認。

assign02 (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てみましょう。

In [2]:

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

25

In [3]:

Grade cell: cell1-7b48244014263505

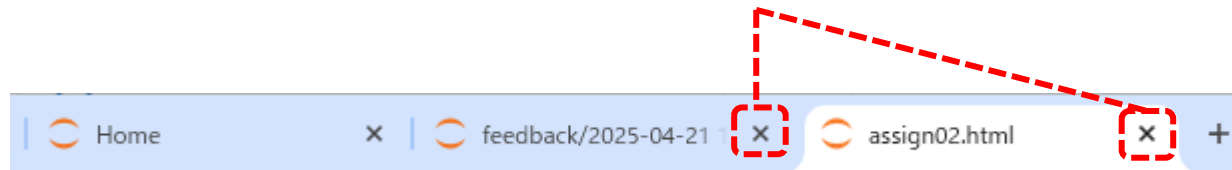
Score: 1.0 / 1.0 (Top)

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

フィードバックの取得 (4/6)

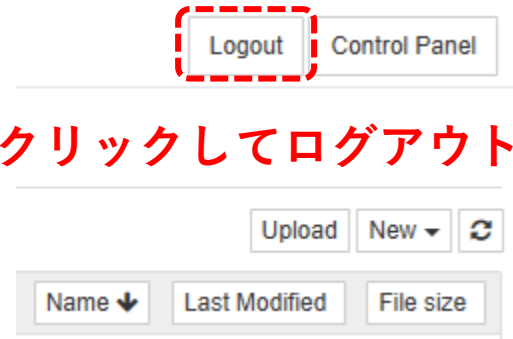
- タブを削除していき、ホームに戻る。

クリックしてタブを削除



フィードバックの取得 (5/6)

- 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。



フィードバックの取得 (6/6)

- 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

アンケートへのご協力をお願い

- 続いて課題の自動採点の操作や演習についてアンケートを行いますので、ご協力いただけますと幸いです。