

JUPYTERHUBとNBGRADERで 授業・試験・レポートをDXして みた件

東京大学 情報基盤センター / 大学院情報理工学系研究科 / 工学部電子情報工学科

田浦健次郎

tau@eidos.ic.i.u-tokyo.ac.jp

はじめに

- 一教員として自分の授業の効率向上を試みたという話です（ご笑覧ください）
- 担当授業（情報系学科）
 - 1年生：数学・物理をプログラミングで考える
 - 3年生：オペレーティングシステム
 - 4年生：プログラミング言語
 - 大学院：Parallel and Distributed Programming

Jupyter notebook, Jupyterlab

- Jupyter **notebook** <https://jupyter.org/>
 - Webブラウザでのプログラム実行環境
 - ポイント：以下をひとつで配布できる
 - 概念説明（文章）
 - 実行できる例（コード）
 - 演習（学生が書いて実行するコード入力欄）
- Jupyter**lab**
 - notebookの進化型



The screenshot displays a Jupyter notebook interface with several code cells and their outputs. The first cell shows a simple arithmetic calculation: `In [14]: (1 + 2) * 3` and `Out[14]: 9`. The second cell is titled "1-2. オブジェクト型 (I) / Object Type (I)" and contains a class definition: `In [15]: class obj: def method(): return 1`, with output `Out[15]: class obj: object method = <function obj.method()>`. The third cell is titled "1-3. リスト / List" and shows a list creation: `In [16]: [1, 2, 3]` and `Out[16]: [1, 2, 3]`. The fourth cell is titled "1-4. 配列 / Array" and shows an array creation: `In [17]: [1, 2, 3]` and `Out[17]: array([1, 2, 3])`. The interface includes input fields for code and output areas for the results.

Jupyterhub, nbgrader

- Jupyterhub <https://jupyter.org/hub>
 - Jupyterを複数ユーザで使うためのサーバ
 - 素のJupyter notebookでは各学生用に立ち上げる（または立ち上げさせる）必要がある
 - 授業など、多人数が利用する環境向け
- nbgrader <https://nbgrader.readthedocs.io/en/stable/>
 - Jupyter Notebookを作成、多人数に配布、回収、採点、返却できる、拡張機能
 - 自動採点の機能もあるが使っていない

- nbgrader実演

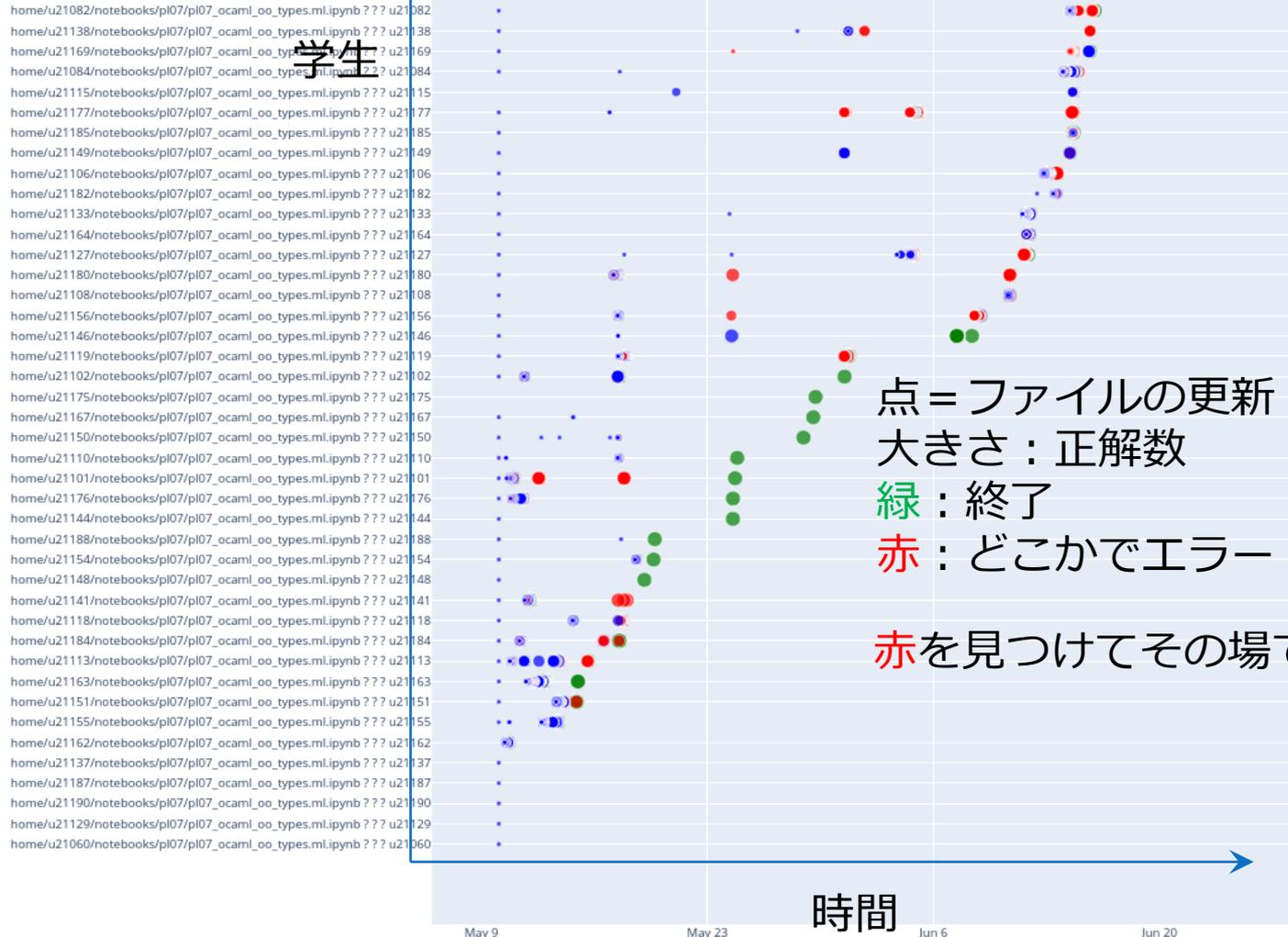
nbgrader自動採点機能

- プログラム課題を自動でテストする機能
- Pythonコーディング課題以外（自由テキスト記述など）には使えない模様
- ⇒ 「マニュアル採点の効率化」に注力
 - その一部として自動テストを適宜導入

授業用に構築している環境

- <https://taulec.zapto.org:8000/>
 - もしよろしければchatにまいたuser/pwdでご覧ください（編集・実行はできません）
- 教材・課題はほぼすべてnotebook
- nbgraderで配布、回収・採点
- 試験（2020冬）もこれで実施した
 - 試験と課題で、システムとしての差はない

進捗把握ツール



教材作成と採点の効率化 (DX)

- ここから先はややマニア向け・個人差のある話になります

問題点：GUI, Webブラウザの作業
はスピードが出ない

朗報：データ形式

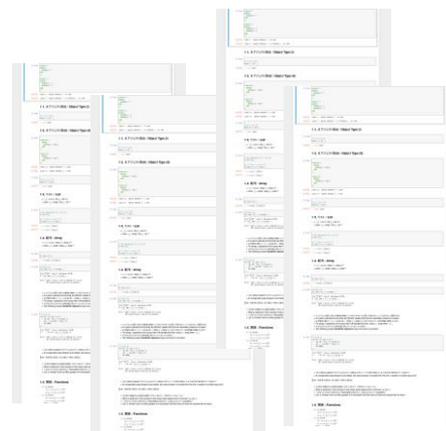
- notebook (*.ipynb) ファイルは単なるjson形式
- nbgrader のデータ（学生の解答など）はひとつのSQLデータベース (sqlite3)
- ⇒プログラムでの読み込みや生成が簡単

教材（試験問題）作成効率化

- 教材（試験問題）はテキストエディタで、書きやすい形式で、高速に書く
 - Markdown形式
 - プログラム + 文章部分をコメントで
- テキストファイルを ipynb に変換するツールを作成

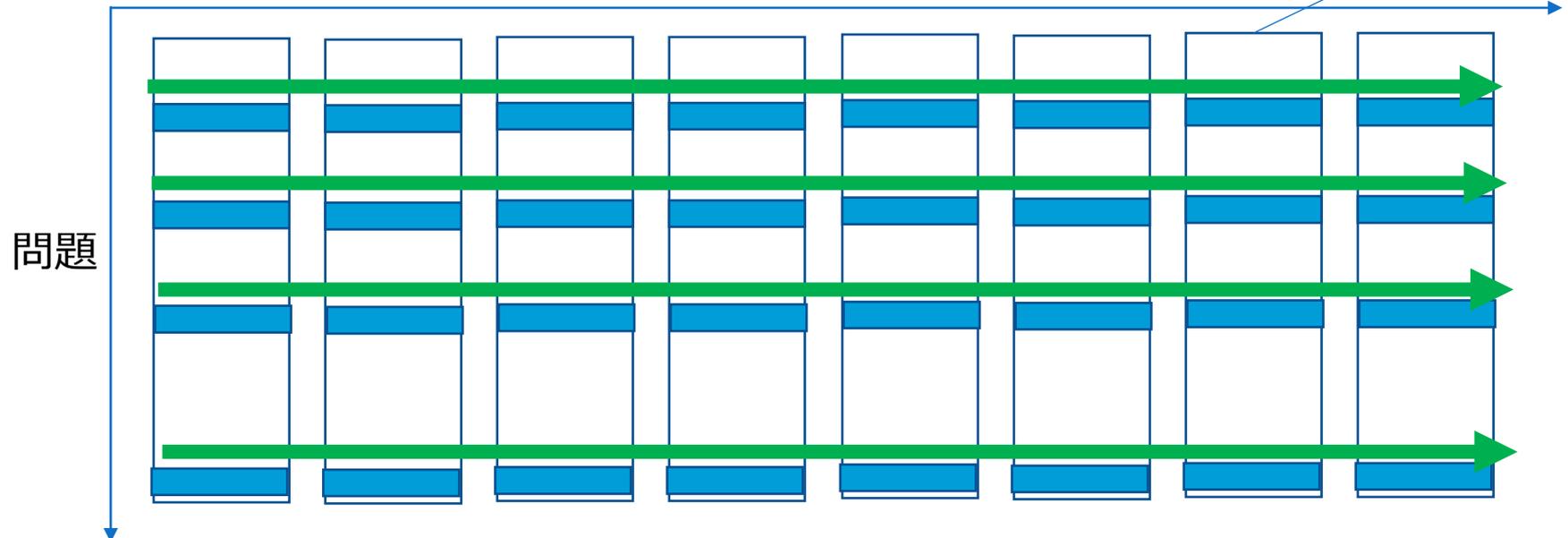


変換



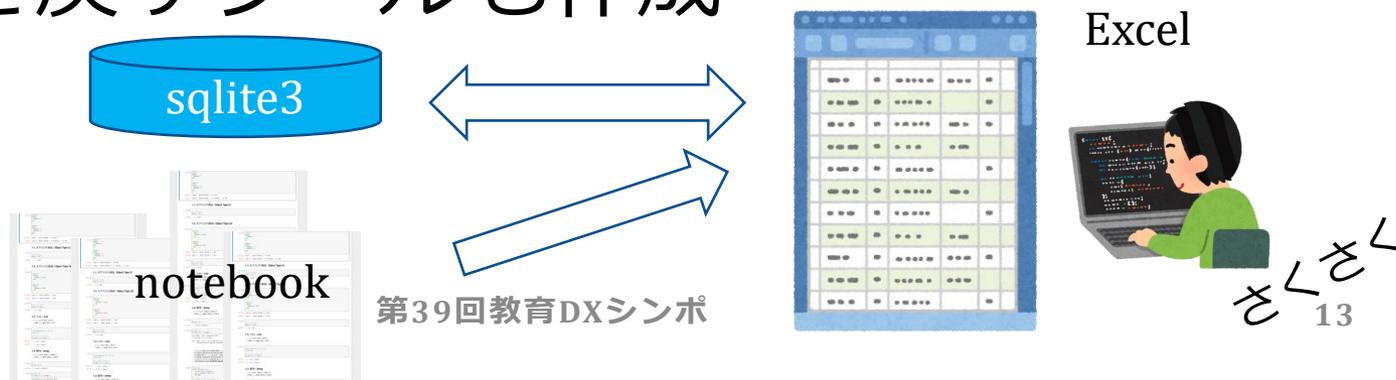
採点作業が遅い理由

- ブラウザ上での作業
 - ページを開く
 - 該当の解答部分までスクロール
 - 同じ問題だけを一気に閲覧・採点できない^{ページ}
- 学生



採点スピードアップのアイデア

- すべての「学生 - 問題 - 解答 - 採点」の表をひとつのExcelファイルにする
- データベース (sqlite3) + notebook からそのようなExcelを抽出するツールを作成
 - 採点はExcel上で「採点」列を埋めていく作業
- Excelに記入した採点結果をデータベースに書き戻すツールも作成



重宝するExcelの機能

- フィルタ作成
 - 一つの問題だけを表示
 - 異なり解答の種類を一望できる
 - 同じ答え（通常正解）の行だけを表示 全部に同じ点数を記入... 一人当たり1秒以下
 - 「ゆらぎのない採点」ができる
- 条件付き表示
 - まだ採点していないセル（空のセル）を色付きで表示
 - 採点のし忘れ防止、目の負担軽減

重宝するExcelの機能

- Pivotテーブル
 - 採点が終わった後、学生を行に、問題を列に、点数をデータにして集計表を作成

つまり...

- 当たり前ですが肝は「データ第一で」でした
 - GUIの手作業は遅い（スピードが出ない）
 - 遅い手作業で結局何をしているか？データ（採点結果）をいじっているだけ
 - 非効率なUIで無理やり作業をするより、作業効率最大のUI（テキストエディタとExcel）で作業をして「データ変換」してほしい結果を得るのが効率的
- これは業務DXでも頻出パターン
- RPA人材 << Python（プログラム）でデータ読み書きできる人材

限界、必ずしもそうでないこと

- 自由テキストの採点にも有効（「プログラミング課題」限定ではない）
 - 一問の答だけを集めて一覧できるのは有用
- コンピュータに向かって打ち込むのが思考の妨げになるような問題には不適切
 - 数式や図をいっぱい書いて考えるような問題
- コンピュータ入力が苦手で著しく不利になる人がいるクラスでは使用に注意

今後

ご清聴ありがとうございました

- まだ試行錯誤中なので自分でしか使っていませんが
 - 変換ツールなど、コードの公開
 - Jupyterhub環境（クラウド）を情報基盤センターとして提供しないのか? > [じぶん](#)
 - 進捗把握ツールでLA（データはあるので）
- などを進めていきたいと思っています
- ありのままでもいいから見たい・使ってみたいという方にはお送りします