

## 内容

当ツールの概要 .....	1
当パッケージの同梱物 .....	2
使い方・事前の準備 .....	4
使い方・第 1 段階：出席 ID を生成し、印刷 .....	5
使い方・第 2 段階：生成した出席 ID の配布、学生の Moodle への入力 .....	7
第 2-1 段階：Moodle に出席 ID の登録場所となる小テストを作る .....	7
第 2-2 段階：出席 ID を配り、学生に Moodle へ入力してもらう .....	8
第 2-3 段階：学生の入力した出席 ID を Moodle から取り出す .....	8
使い方・第 3 段階：学生の入力した出席 ID を事前生成したものと照合 .....	10
第 3 段階の処理例（パッケージに同梱の例示用ファイルをもとに） .....	12

### 当ツールの概要

当ツールはランダムに ID をたくさん作り、それを教室で学生に配って自分で登録してもらうものです。使い方は以下の 3 段階に分かれます。

- 第 1 段階：出席 ID を生成し、印刷する。
- 第 2 段階：生成した出席 ID を教室で配り、学生にシステムへ入力してもらう。
- 第 3 段階：入力された出席 ID が事前に生成したものと一致するか照合する。

使うための要件は以下です。

- PC 上で Python3 を利用できる。
- 第 2 段階で使うシステムには Moodle を想定する。  
ただし、Moodle 以外のシステムに自分で対応するなら問題ない。

### 【背景】（興味がなければとばして次へ）

教員のみなさんへ。大人数の教室で出席カードを配り、記入してもらったものを回収し、記録をつ

ける、って大変だと思いませんか。

こうした古典的な出席カードのデメリットは以下です：

- 配ったものを回収しなければならない。
- 回収したものを手で PC 上の名簿へ転写しなければならない。
- 転写の際に記録の間違いが生じうる。
- TA を雇うにしても、そのコストが生じる。
- あとで学生が問い合わせで「出席していたはずだ」ということが起きうる。

そこで、ここではランダムにたくさん発生させた ID を印刷して学生に配り、あとは学生にその ID をシステムへ自分で登録してもらうツールを提供します。学生はその登録により「出席」となります。一人一人には異なる ID が与えられます。

この方式のメリットは以下です：

- 配ったものを回収する必要がない。
- 人手で転写する手間が生じない。学生が自分で登録する。
- 教室では ID を配るだけで終わる。TA を雇うにしても、配ってもらうだけですむ。
- 学生は自分の登録をシステム上で確認できる。ID を印刷したカードも学生の手元に残る。教員もシステム上の記録を確認できる。

一方で、この方式のデメリットは以下です：

- 以下に提供するツールは、古典的な出席カードに比べれば複雑である。
- ランダムに発生させた ID を印刷する必要がある。  
印刷したものを裁断する必要もある（ただし、ミシン目付きの用紙なら裁断は不要）。

## 当パッケージの同梱物

### README

- 00\_README.pdf 当説明

### プログラム類

- 01\_id\_generator.py ランダム出席 ID 生成プログラム
- 02\_quizForm\_moodle\_xml.xml Moodle 用の出席 ID 登録（小テスト形式）
- 03\_id\_checker.py ID 照合プログラム

### 例示用ファイル

- t\_id\_001.csv 01\_id\_generator.py で作られた第 1 回授業用出席 ID
- t\_id\_002.csv 同、第 2 回授業用出席 ID

- t\_moodle\_001ans.csv
- t\_moodle\_002ans.csv
- t\_students.csv

第 1 回授業で学生が入力した出席 ID の結果  
同、第 2 回授業で学生が入力した出席 ID の結果  
Moodle 評定表の学生 ID を取り出したもの

## 使い方・事前の準備

以下のインストール法が分からない場合はインターネットで検索してご対応ください。

- あなたの PC に python3 をインストールしてください。  
その際、パスを通す設定にしてください。  
(インストール時に「Add Python to PATH」を選択するなど)

Windows の場合、インストールをおこなうと、「ターミナル」や「コマンドプロンプト」から「python」を実行できるようになります。Mac、Linux でも同様になんらかのターミナルから実行できます。

- 以下の python ライブラリをインストールしてください。
  - pandas
  - openpyxl
- 当パッケージをフォルダに展開します。  
場所はどこでも結構です。

## 使い方・第1段階：出席 ID を生成し、印刷

### 【すること】

01\_id\_generator.py の置かれた場所で、ターミナルを開きます<sup>1</sup>。  
そして以下を実行してください<sup>2</sup>。

```
$ python 01_id_generator.py 200 14
```

- この例の場合、学生 200 人分の出席 ID を 14 クラス分、生成します<sup>3</sup>。  
数値はあなたの学生数、授業回数に合わせて変えてください。  
学生数、授業回数を省略した場合は、プログラムの持つデフォルト数で生成します。
- 生成される出席 ID はランダムな 8 つの英数字です。  
数字の「1」とアルファベットの「l」など、誤読されやすいものは除外しています。  
出席 ID の例：x5-ak-t7-w2

### 【作られるもの】

- **id\_001.csv** 第 1 回目授業用に生成された出席 ID
- **id\_001\_printer.xlsx** id\_001.csv の印刷用  
(これらが指定された授業回数分、python を実行したフォルダ上に作られます)

### 【ポイント】

以下は第 1 回授業用ファイル（「\_001」がファイル名に付くもの）を例に説明します。  
他の授業回用ファイルも同様です。

- 実行するたびに上書きされるので、必要なものは別フォルダへ移すなどしてください。

---

<sup>1</sup> Windows の場合、このパッケージを展開したフォルダ内で右クリックをし、「ターミナルで開く」または「コマンドプロンプトで開く」を選択するのが最も簡単な方法です。右クリックにそれらのメニューがない場合は、追加する方法を検索してまずご対応ください。

<sup>2</sup> ターミナルで python を実行すると「No such file or directory」などとエラーになる場合、プログラムの置かれたフォルダと異なるフォルダをターミナルが参照しているものと思われます。ターミナルの参照しているフォルダをプログラムの置かれたフォルダに変える必要があります。

<sup>3</sup> 1 回の実行内で生成される出席 ID は授業回全体で互いに重複しないようにしています。そのため、学生が出席 ID を問い合わせるときには、それがどの授業回のものかを特定できます。  
2024-01 時点では 22 種類の英数字から 8 文字の出席 ID を生成しています。したがって、たとえば 300 人の学生へ 15 回の講義をおこなう場合、出席 ID を手にしなかった者がランダムな入力で発行済 ID と偶然に一致する確率は  $(300 \times 15) \div (22 \text{ の } 8 \text{ 乗}) \approx 1/12,000,000$  となります。

- id\_001.csv は第 3 段階で必要になるため、学生に配布した後は、学生問い合わせ期間が終わるまで大切に保存してください。これを紛失したり改変したりすると、学生の出席を確かめられなくなります。
- id\_001\_printer.xlsx は、あなたのプリンタと用紙に合わせて調整してください。プログラム内では下の方で左右のマージンなどの用紙調整をしています。変更可です。
- id\_001\_printer.xlsx を印刷し、はさみや裁断機で裁断してください。ミシン目のついた用紙に印刷することで裁断の手間を省くこともできます<sup>4</sup>。

---

<sup>4</sup> ASKUL のミシン目用紙 : [https://www.askul.co.jp/p/3998878/?int\\_id=recom\\_DtVar](https://www.askul.co.jp/p/3998878/?int_id=recom_DtVar)

## 使い方・第2段階：生成した出席IDの配布、学生のMoodleへの入力

第2段階はさらに3つの段階に分けて進めます。

### 第2-1段階：Moodleに出席IDの登録場所となる小テストを作る

#### 【すること】

以下のファイルをMoodleの問題バンクにインポートしてください。

そして、この問題を使い、授業回ごとにMoodle小テストを設置してください。

「Class01 出席登録場所」、「Class02 出席登録場所」、という風に、です。

Moodleでは「小テスト」形式ですが、実際には学生が出席IDを登録するためだけのものです。

02\_quizForm\_moodle\_xml.xml

#### 【ポイント】

- Moodleの問題バンクや小テストの設定法についてはMoodleのヘルプをお探しく下さい。
- このxmlファイルの中に入っているのは1問のみです。  
8つの空欄の穴埋めです。
- このxmlをMoodleの問題バンクにインポートすると、「cNavi」というカテゴリ下に入ります。
- いちどMoodleの問題バンクにインポートすれば、あとはその問題バンクから小テストを作るだけです。すなわち、問題バンクへのインポートを毎回する必要はありません。
- Moodleで小テストを設定する際、以下のオプションを採用することを勧めます。

➤ 「評点」	> 「受験可能回数」	= 無制限
➤ 「評点」	> 「評定方法」	= 最後の点数
➤ 「問題の挙動」	> 「直近の解答内容を反映させる」	= Yes
➤ 「レビューオプション」		= 「受験」「答えの正誤」のみ Yes
➤ 「活動完了」	>	
	「完了するには学生はこの活動を閲覧する必要があります」	
	「この活動を完了するには学生が評定される必要があります」	
	「この活動を完了するには学生は合格点に達する必要があります」(満点)	

- このMoodle小テストでは、学生の入力する文字が想定されるものかだけをチェックします。

想定されない文字には「不正解」、想定される文字には「正解」を表示します。

これは、この時点で学生の誤入力をできるかぎり防ぐためです。

一方で、学生が入力する出席 ID が第 1 段階で生成した出席 ID と一致するかは、ここではチェックしていません。そのチェックは第 3 段階でおこないます。つまり、学生は、Moodle 小テスト上で満点を得ても、自分の入力した出席 ID が作成済のものと同じかどうかは分かりません。この点は事前に授業で教員から学生のみなさんへお知らせいただくと良いかと思います。

### 第 2-2 段階：出席 ID を配り、学生に Moodle へ入力してもらう

印刷した出席 ID を学生一人一人に配ります。

学生にはその出席 ID を上で設置した小テストを通じて Moodle へ入力してもらいます。

授業回ごとに異なる場所へ入力してもらうことになります。

ある学生の出席 ID が他の学生の目にあまり触れないよう、配布法についてはご配慮ください。

配布の際、以下を学生へお伝えください。

- 配られた出席 ID はほかの学生にみせないでください。  
同じ出席 ID が複数の学生から登録された場合、出席が取り消されます。
- Moodle に入力する際、Moodle は入力される文字が想定されるものかどうかだけをチェックし、問題がなければ正解とします。正しい出席 ID かどうかまではチェックされていません。

### 第 2-3 段階：学生が入力した出席 ID を Moodle から取り出す

#### 【すること】

第 2-1 段階で設置した Moodle 小テストから、学生の受験結果（出席 ID の入力結果）をとりだします。

手順は以下です。

- Moodle に設置した小テスト > 「受験結果」 > 「解答」へ行きます。  
「解答」のところはデフォルトでは「評定」になっているかもしれないので注意を。
- 以下のオプションを設定します。

- 受験者 = 小テストを受験したことのある登録済みユーザ
- 受験状況 = 「終了」のみ  
さらに「ユーザごとに最大 1 件の終了した受験を表示」 = Yes

➤ テーブルデータ カンマ区切り値 (.csv)

- 以上の設定でダウンロードし、保存します。  
ここでは学生の入力結果を「**m\_001ans.csv**」に保存したとして第3段階に行きましょう。
- 以上の設定をすると、保存したファイルに同じ学生が複数回掲載されることはありません。  
もし同じ学生が複数回掲載されていると、続く過程でエラーが起きます。

## 使い方・第3段階：学生の入力した出席 ID を事前生成したものと照合

### 【すること】

id\_checker.py を置いた場所でターミナルを開き、以下を実行します。

以下は第1回授業のファイル（「\_001」のもの）を例にしています。

実際には照合したい授業回のもを指定してください。

```
$ python 03_id_checker.py id_001.csv m_001ans.csv
```

または

```
$ python 03_id_checker.py id_001.csv m_001ans.csv students.csv
```

- **id\_001.csv** 第1段階で作成したファイル。  
作成した出席 ID が入っています。  
照合したい授業回のもを指定してください。（例は第1回授業です）
- **m\_001ans.csv** 第2段階で Moodle からダウンロードしたファイル。  
学生の入力した出席 ID が入っています。  
照合したい授業回のもを指定してください。（例は第1回授業です）
- **students.csv** （省略可）結果に出力したい学生を取めたファイル。  
例えば評定表（成績表）の学生を取めれば、それら学生をその順で出力。  
省略すると m\_001ans.csv 内の学生が同じ順で出力されます。

### 【エラーになる場合】

2つ目に指定するファイル（上の例だと m\_001ans.csv）の該当する列がうまく読み込まれていないと思われる場合は、03\_id\_checker.py をなんらかのテキストエディタで開き、以下を指定しなおしてください。Moodle の仕様によって記録列が異なります。

- c\_sn1 学生名の記されている列番号（1列目が0、2列目が1、..）
- c\_sid 学生 ID の記されている列番号
- c\_aid 出席 ID の記されている列番号

### 【作られるもの】

- **m\_001ans\_at.csv** 学生とその授業回の出欠記録。  
2つ目に指定したファイルに「\_at」をつけたファイルが作られます。

このファイルの内容

列名	内容
student_ID	学生番号 3つ目のファイルが与えられればそのファイル内の学生順で、 そうでなければ2つ目のファイル内の学生順で、出力されます。
attendance	=1：出席      /      =0：欠席（または欠席扱い）

---

	(出欠はこの記録をご利用ください。 他学生と重複する ID を持つ学生には 0 を出力しています)
attendance_ID	学生の入力した出席 ID
dup_check	=1：重複なし / =0：記録なし /=-1：他学生と重複あり (他学生と重複する場合にどうするかは教員の考えによります)
attendance_ID_found	=1：事前発行済の ID と一致 / =0：一致しない
name	学生名

---

### 第3段階の処理例（パッケージに同梱の例示用ファイルをもとに）

パッケージに同梱する例示用ファイルを使って以下をお試しください。

#### 第1回授業の例だと

```
$ python 03_id_checker.py t_id_001.csv t_moodle_001ans.csv
```

または

```
$ python 03_id_checker.py t_id_001.csv t_moodle_001ans.csv t_students.csv
```

(2つ目の例では、t\_moodle\_001ans.csv に記録がある学生のうち t\_students.csv に掲載されていない者がいるため、warning が出ます。聴講生がいる場合などにこうなります)

#### 第2回授業の例だと

```
$ python 03_id_checker.py t_id_002.csv t_moodle_002ans.csv
```

または

```
$ python 03_id_checker.py t_id_002.csv t_moodle_002ans.csv t_students.csv
```

それぞれ、詳細は以下です：

- 当該科目の受講生は 15 人+ゲスト聴講生 1 人。
- C01 の Moodle 小テスト結果ファイル
  - 16 人のうち 8 人に入力を依頼。
  - うち 6 人はこちらの与えた ID を入力。
  - 残り 2 人は出席票を得ず自分でトライした ID を入力。
  - 1 人は言語設定 = 英語。
- C02 の Moodle 小テスト結果ファイル
  - 16 人のうち C01 とは異なる 8 人に入力を依頼。
  - 依頼した 8 人のうち、2 人が 1 つの ID を共有、別の 3 人が別の 1 つの ID を共有した。つまり出席票を他人に渡したという想定である。
  - 残り 3 人には固有の出席票 ID を渡した。
  - 1 人は言語設定 = 英語。
- student.csv
  - Moodle の評価表から学生 ID を抜き出したもの。
  - 15 人分。聴講生は評価対象外のため評価表に掲載されていない。

(以上です)