

# PsychoPy Builderによる オンライン実験

愛媛大学  
十河 宏行

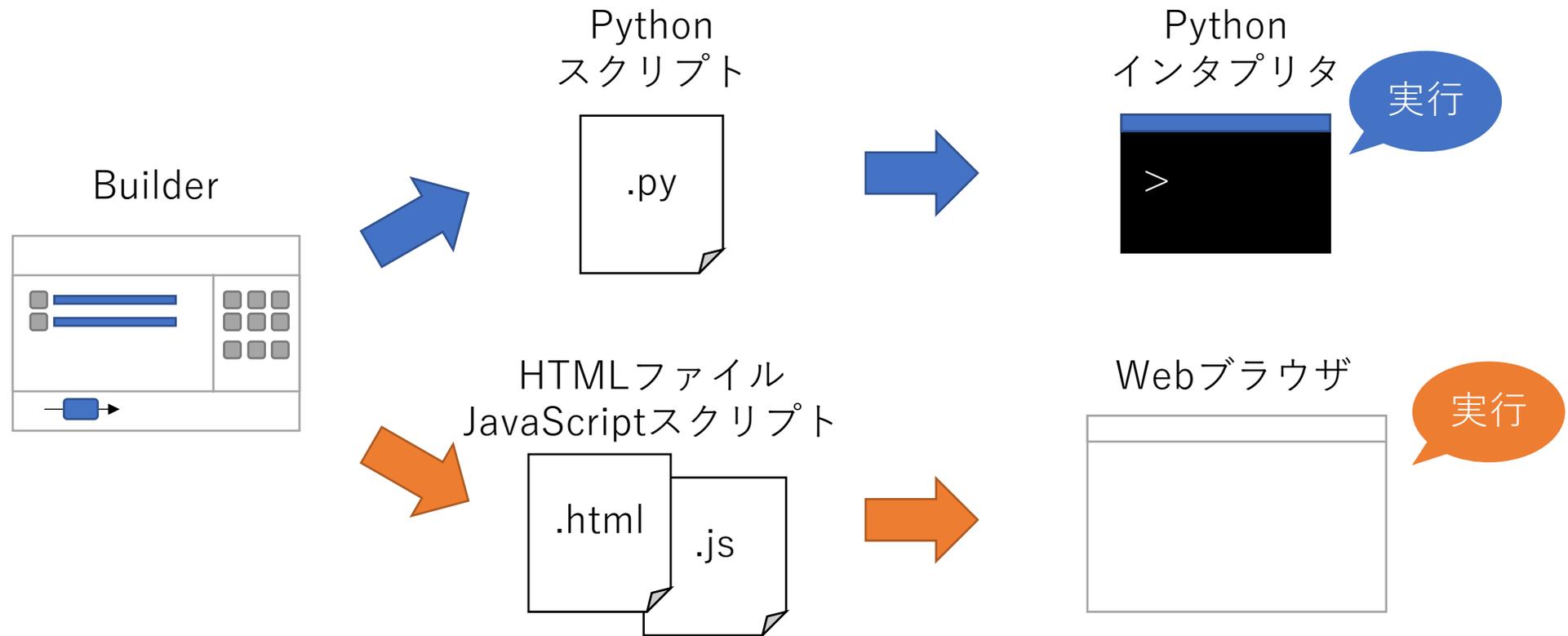
# 内容

- Builderでオンライン実験：その仕組み
- この実験はオンライン化できる？
- ワーク：Stroop課題をオンライン化する

「PsychoPyでつくる心理学実験」第6章程度のBuilderの知識がある人向けの内容です。

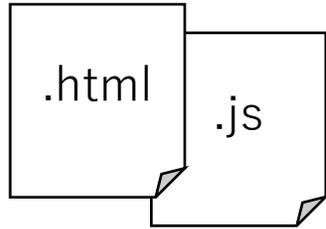
- 条件ファイルを使って刺激の色を変更できる。
- キーボードで反応を記録できる。
- Codeコンポーネントでif文を使って反応に応じたフィードバックができる。

# Builderでオンライン実験：その仕組み



Webブラウザ上で動作する実験をBuilderが出力できるようになった。

HTMLファイル  
JavaScriptスクリプト



アップロード

Pavlovia  
サーバ



スクリプトを  
ダウンロード



データを  
アップロード

実験データ



データを  
ダウンロード

参加者



Pavloviaを使わずに実施するのは  
技術的にかなり難しいです。



**Pavlovia**というwebサービスを使って実験を実施できる。

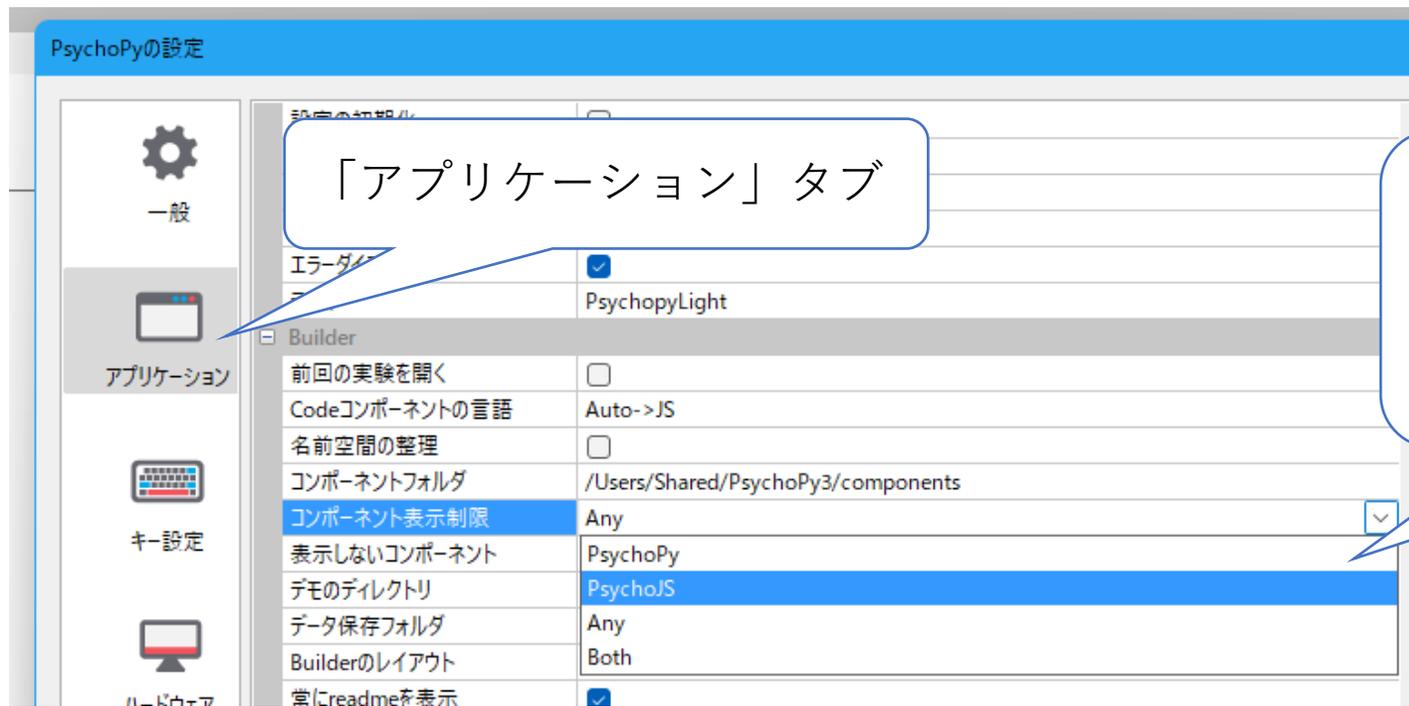
# Builderでのオンライン実験に必要なもの

PsychoPy Builder	なるべく新しいもの (現時点では2021.2.3を推奨)
Pavloviaのアカウント	メールアドレスが必要 登録自体は無料
Pavloviaのクレジット または機関ライセンス	オンライン実験を行うには必要 試してみるだけなら不要 クレジット：実験1回につき £ 0.2 機関ライセンス：年間 £ 1500

£ 1=150円換算で  
30円と225000円

# この実験はオンライン化できる？

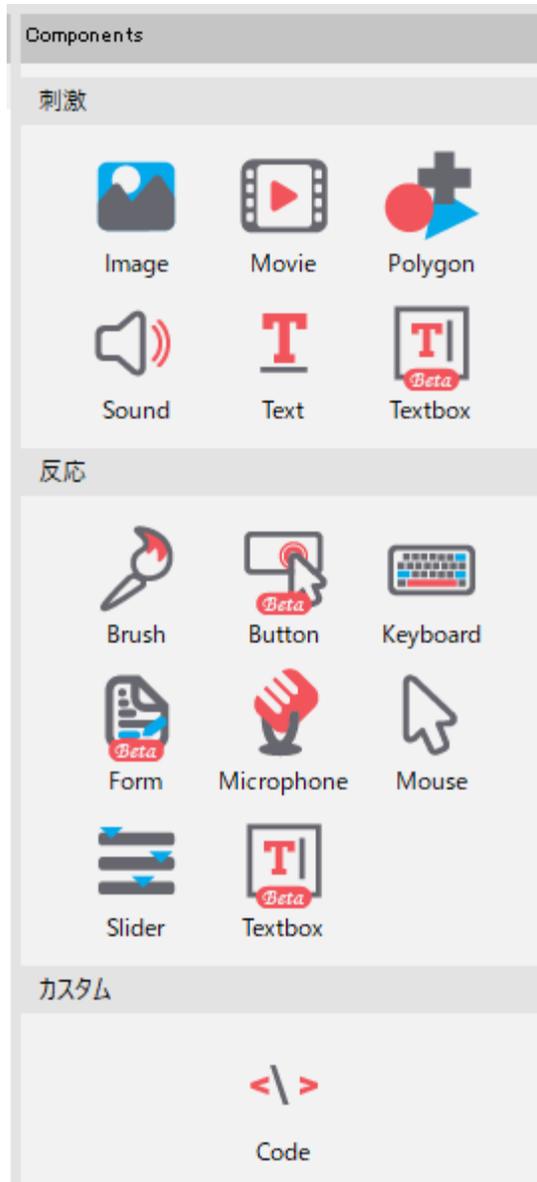
- 実験の作り方自体はいままでと同じだが、**オンライン実験に対応していないコンポーネントが結構ある。**



「コンポーネント表示制限」を **PsychoJS** にするとオンライン実験対応のものだけ表示されるようになる。  
(PsychoPyの再起動が必要)

PsychoJSとはPsychoPyのコードをJavaScriptに移植するためのライブラリだと思ってください。





2021.2.3において「コンポーネントの種類」を **PsychoJS** にすると、表示されるコンポーネントは左に示したもののだけになってしまう。

Polygon, Image, Sound, Movie, Text, TextBox  
Keyboard, Mouse, Slider  
Code

が使えるので意外と多くの実験に対応できるが、GratingやDotsを使った従来の実験を移植するのは難しい。

# ワーク：Stroop課題をオンライン化する

- 「PsychoPyでつくる心理学実験」第6章のStroop課題をオンライン化してみる。

- 使用しているコンポーネントは**Text, Keyboard, Code**のみ。  
すべてPsychoJSに対応。

- Codeコンポーネントでは  
if文による反応の正誤のフィードバック  
直近5試行の正答数のフィードバック  
をおこなっている。

# まずPythonで実行して動作確認する

みどり

正答

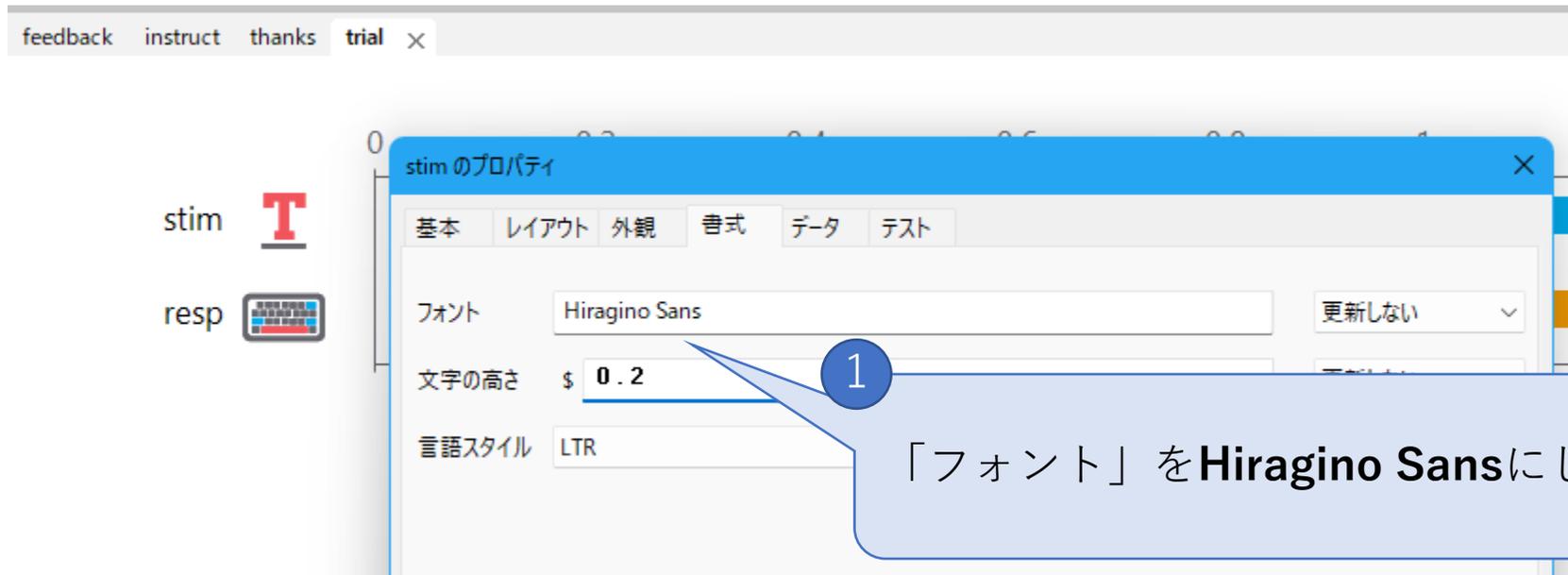


正解です！ RT=0.872 (直近5問中3問正解)

誤答



不正解です！ (直近5問中3問正解)

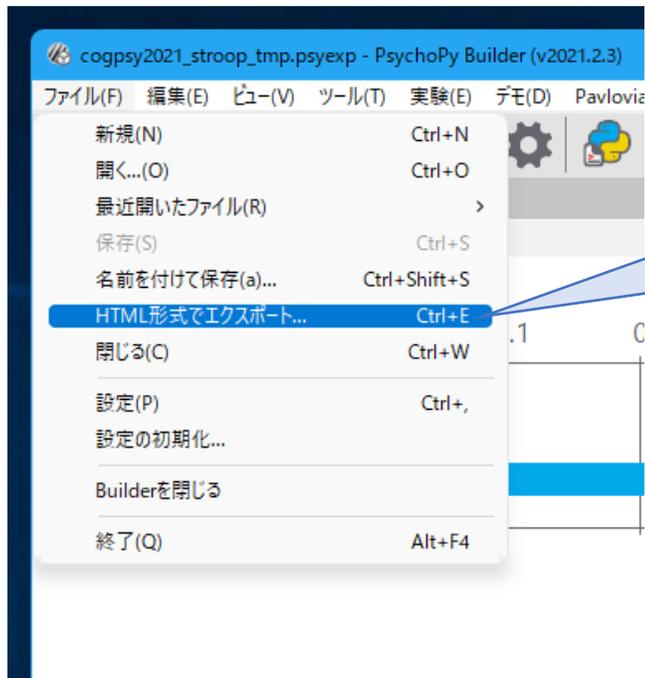


Macで日本語の表示が乱れる場合はTextコンポーネントのフォントを**Hiragino Sans**にすると解消される（macOS 11.62, PsychoPy 2021.2.3で確認）。

Mac上で表示が乱れていてもオンラインで実行すると正しく表示されるので、オンラインでしか実行しないなら変更しなくてもよい。

# オンライン用に変換する

オンライン実験未対応のコンポーネントを使っていなければ、**HTML形式でエクスポート**することでオンライン実験用コードを出力できる。

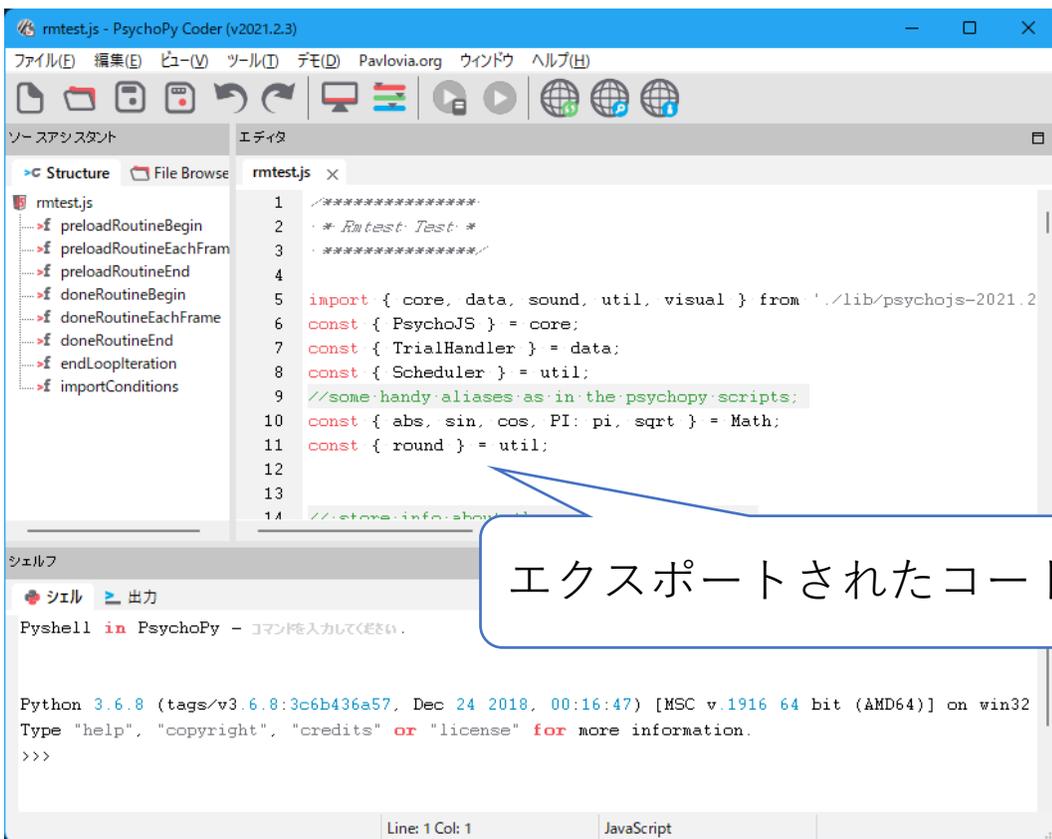


1

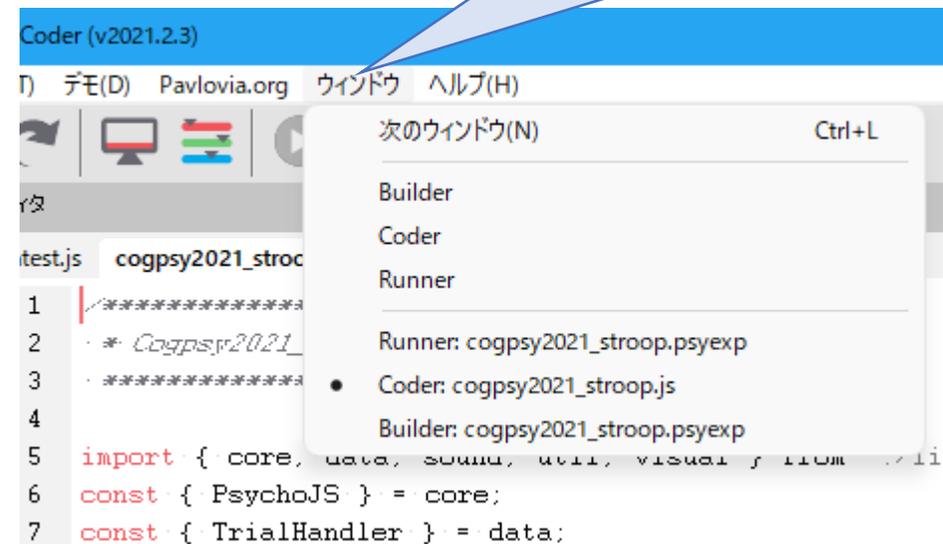
「ファイル」の「HTML形式でエクスポート…」を選択

「実験の設定」ダイアログの「オンライン」から実験保存時に自動的にオンライン実験用ファイルが出力されるようにすることもできます。





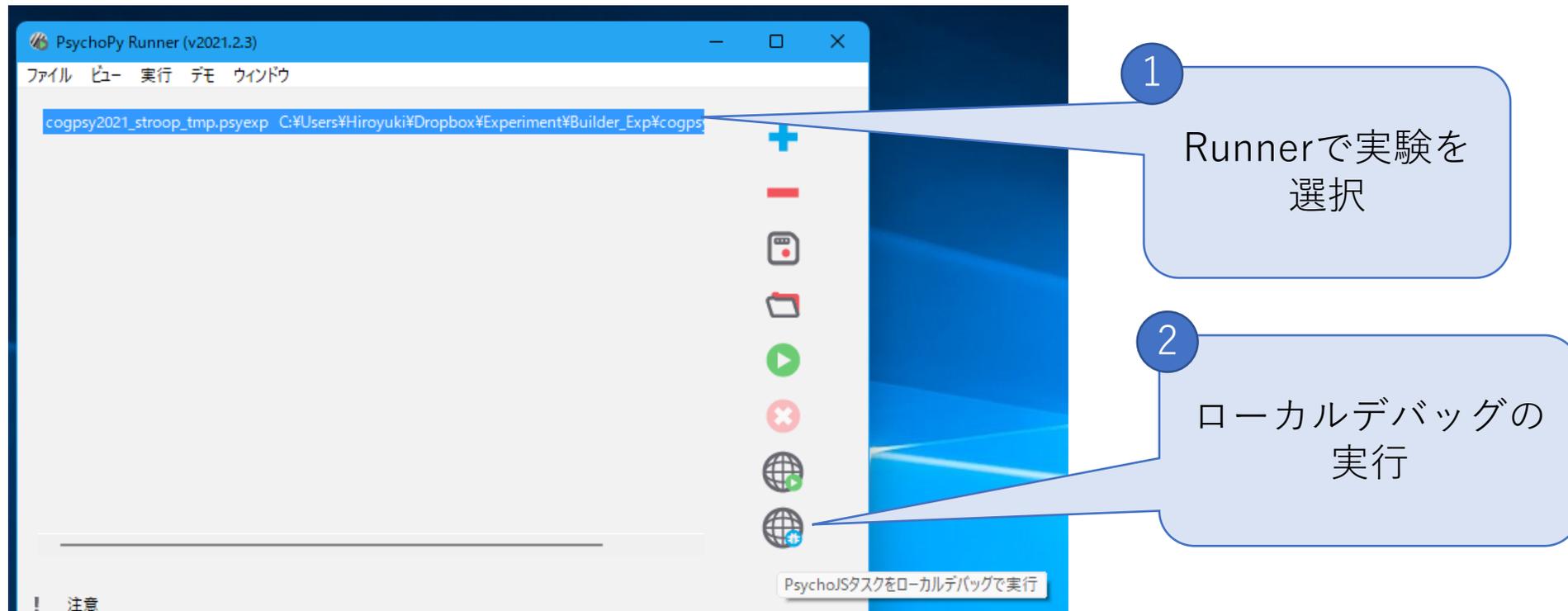
1  
BuilderやRunnerを見失った場合は「ウィンドウ」が便利

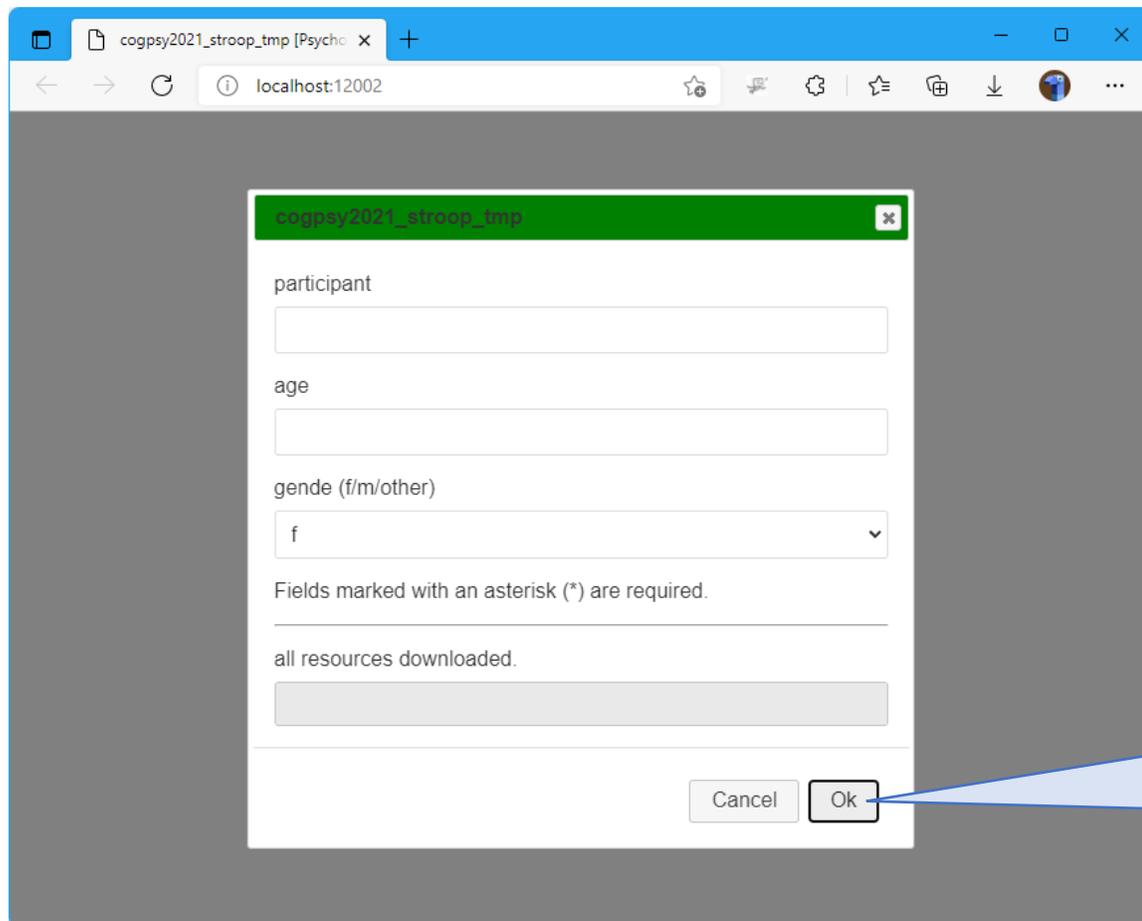


HTML形式でエクスポートすると、PsychoPyのCoderが開いてコードが表示されるが、これは閉じても構わない。BuilderやRunnerのウィンドウを見失った場合はメニューの「ウィンドウ」を使うと便利。

# ローカルデバッグ

自分のPCのブラウザで実験の動作を確認できる（ローカルデバッグ）。  
ローカルデバッグはRunnerからおこなう。





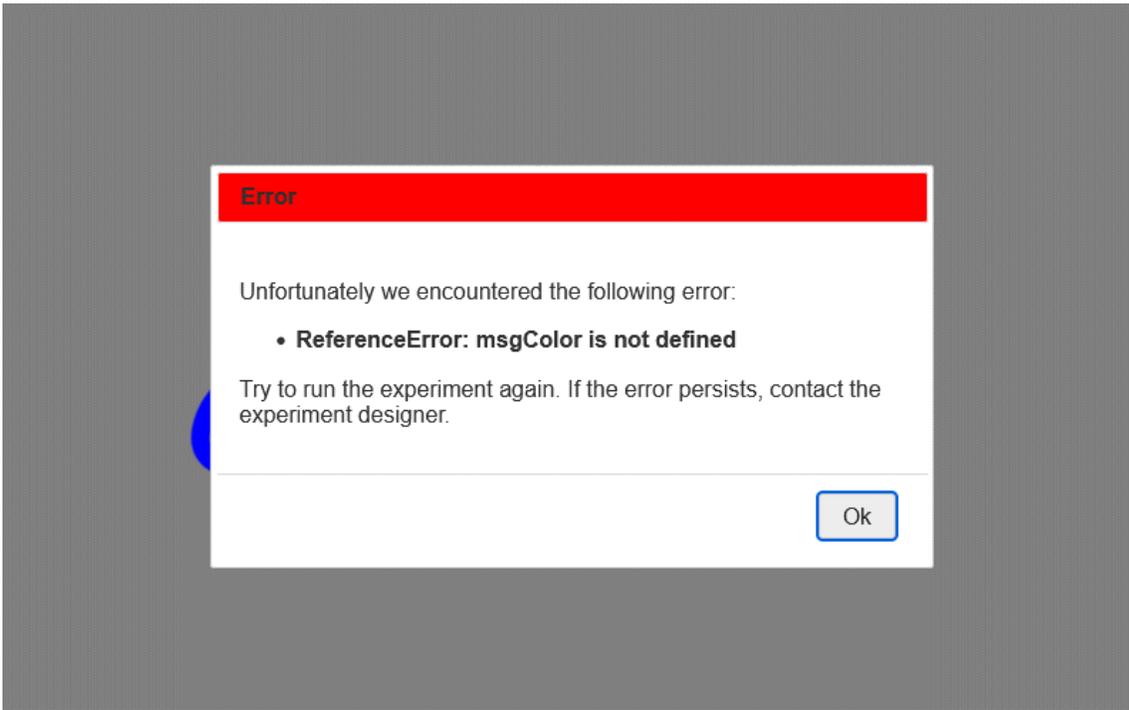
特に初回はブラウザが開くまで  
かなり待たされる事があります。



1

OKをクリック

標準のブラウザが開き、ブラウザ内に実験情報ダイアログが表示される。  
Okをクリックすると実験が始まる。



「PsychoPyでつくる心理学実験」第2章の実験のようなシンプルな実験ならばHTML形式でエクスポートするだけで動作するのだが、今回とりあげる第6章の実験は1回目の反応の直後にエラーで止まってしまう。

- ! Codeコンポーネントを使っている
- ! コンポーネントのプロパティでPythonの式を使っている
- ! 実行時でないとは確定しない条件ファイルから画像ファイルなどのリソースを読み込んでいる

などのときにトラブルが生じやすい。  
エラーで停止はしないが、**回転方向が逆**になるのも頭が痛い問題。

今回の実験ではCodeコンポーネントが問題となっている。

# Codeコンポーネントの自動変換を使う

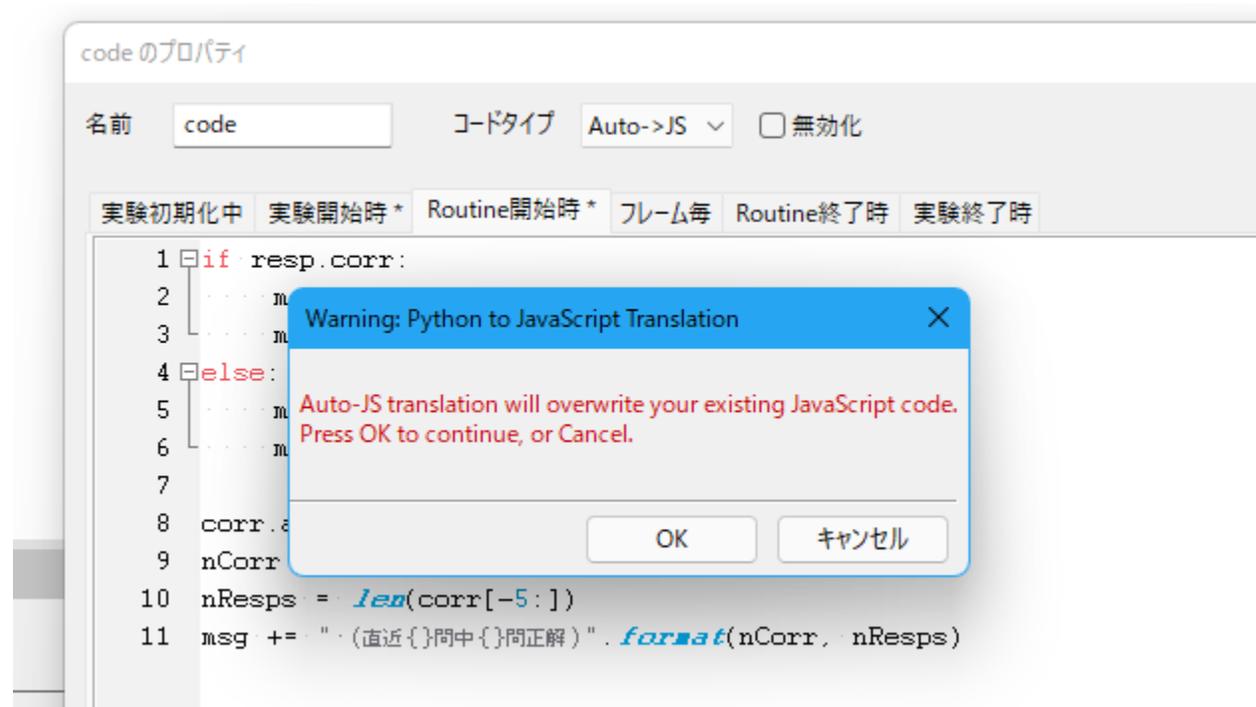
The image shows a screenshot of the PsychoPy Builder interface. The main window displays a routine named 'feedback' with a 'code' component selected. A 'code のプロパティ' (Code Properties) dialog box is open, showing the 'code' component's settings. The 'コードタイプ' (Code Type) dropdown menu is open, showing options: Py, JS, Both, and Auto->JS. The 'Auto->JS' option is highlighted. Three blue callout boxes with numbers 1, 2, and 3 provide instructions:

- 1 feedback ルーチン (feedback routine)
- 2 code (code component)
- 3 コードタイプを Auto->JSに (set code type to Auto->JS)

The dialog box also shows the '名前' (Name) field set to 'code', a '無効化' (Disable) checkbox, and a code editor with the following content:

```
1 msg=""  
2 corr=[]  
3
```

CodeコンポーネントにはPythonのコードをJavaScriptに自動変換する機能がある。



Auto->JSにすると、入力済みのJavaScriptコードを上書きしてよいか聞いてくる。今回は入力済みコードがないのでOKを選択する。

1

Routine  
開始時

code      コードタイプ    Auto->JS     無効化

実験初期化中    実験開始時\*    Routine開始時\*    フレーム毎    Routine終了時    実験終了時

```
1 if resp.corr:
2     msg = "正解です! RT={:.3f}".format(resp.rt)
3     msgColor = "lime"
4 else:
5     msg = "不正解です!"
6     msgColor = "red"
7
8 corr.append(resp.corr)
9 nCorr = sum(corr[-5:])
10 nResps = len(corr[-5:])
11 msg += " (直近{}問中{}問正解)".format(nCorr, nResps)
```

```
1 /* Syntax Error: Fix Python code */
```

ヘルプ      OK      キャンセル

右側に変換されたコードが表示される。  
今回の例では「Syntax ErrorがあるからPythonコードを修正しろ」と言われるので何とかしないとイケない。

自分でJavaScriptのコードを書いた方が楽？

はい



コードタイプを**Both**にする。  
オンラインでしか実行しない  
なら**JS**にしてもよい。

いいえ



JavaScriptに自動変換しやすいようにコードを修正する。

今回はこちら



# Codeコンポーネントの問題を修正する

- Codeコンポーネントの自動変換機能はPsychoPyのバージョンアップのたびに改善されているが、まだまだ全面的にお任せというわけにはいかない。
- 今回のケースではPythonの文字列における**format()メソッド**がJavaScriptの文字列にはないことがエラーの原因である。
- **Auto->JSの場合、JavaScriptのコードは編集できないのでPythonのコードを修正する。**

**str()**で文字列に変換してから+で連結すればよい。

[Routine開始時]のPythonのコードの2行目を以下のように修正する。

1

```
msg = "正解です！ RT={:.3f}".format(resp.rt)
```



```
msg = "正解です！ RT="+str(resp.rt)
```

同11行目を以下のように修正する。

2

```
msg += "(直近{}問中{}問正解)".format(nResps, nCorr)
```



```
msg += "(直近"+str(nResps)+"問中"+str(nCorr)+"問正解)"
```



正解です！ RT=0.793650500010699  
(直近2問中4問正解)

修正結果をローカルデバッグで確認するには、その都度**HTML形式でエクスポート**する必要がある。

実行してみると、RTの桁数が大変なことになっている。  
format()での桁数指定にあたる処理をおこなう必要がある。

数値の丸めをおこなう **round()** を使えばよいが、JavaScriptのround()は桁を指定できない。泥臭いが以下のように1000倍して丸めてから1000で割ること  
とで回避できる。

1

```
msg = "正解です！ RT="+str(resp.rt)
```



```
msg = "正解です！ RT="+str(round(1000*resp.rt)/1000)
```

これでコード修正前と同じ動作となった。

# どう修正すればよい？

- PsychoPy Discourse(PsychoPyのコミュニティ)に**PsychoPy Python to Javascript crib sheet**というスレッドがあり、かなり詳しい情報が得られる（英語）。
  - <https://discourse.psychopy.org/t/psychopy-python-to-javascript-crib-sheet/14601>
- 「**PsychoPy/Pavloviaによるオンライン実験**」（朝倉書店）という書籍が出版されるので、そちらも参考にしていただけると幸いです。

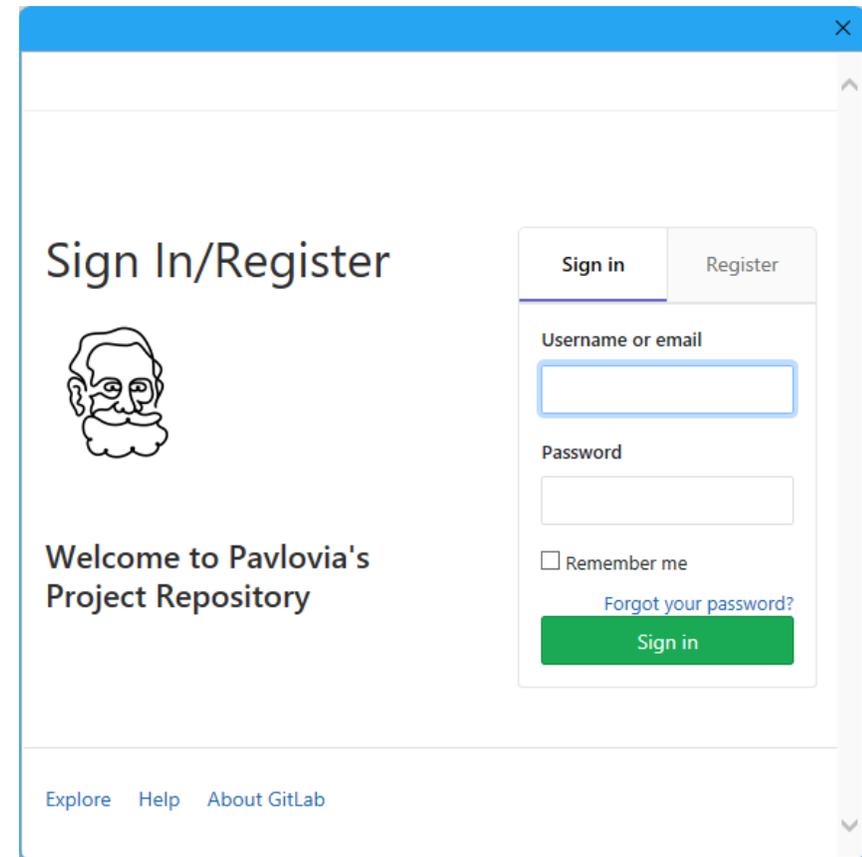
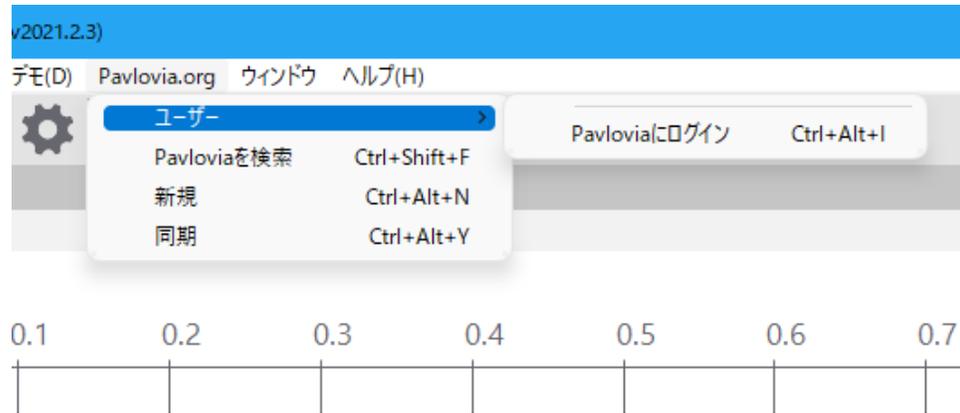
# プロジェクトの作成

- ローカルデバッグで問題がなければPavloviaに転送をおこなう。
- Pavloviaでは実験をプロジェクトという単位で管理する。

- 実験に必要なファイル（刺激用の画像や音声、条件ファイルなど）をひとつのフォルダにまとめる。
- プロジェクト内のファイルは実験実施時に外部からアクセス可能なので、アクセスされたらまずいファイルは置かない。
- ひとつのフォルダにはひとつの実験ファイル(psyexpファイル)。
  - ふたつ以上あっても構わないがPavloviaで実行できるのは最後に「HTML形式でエクスポート」した実験だけ。

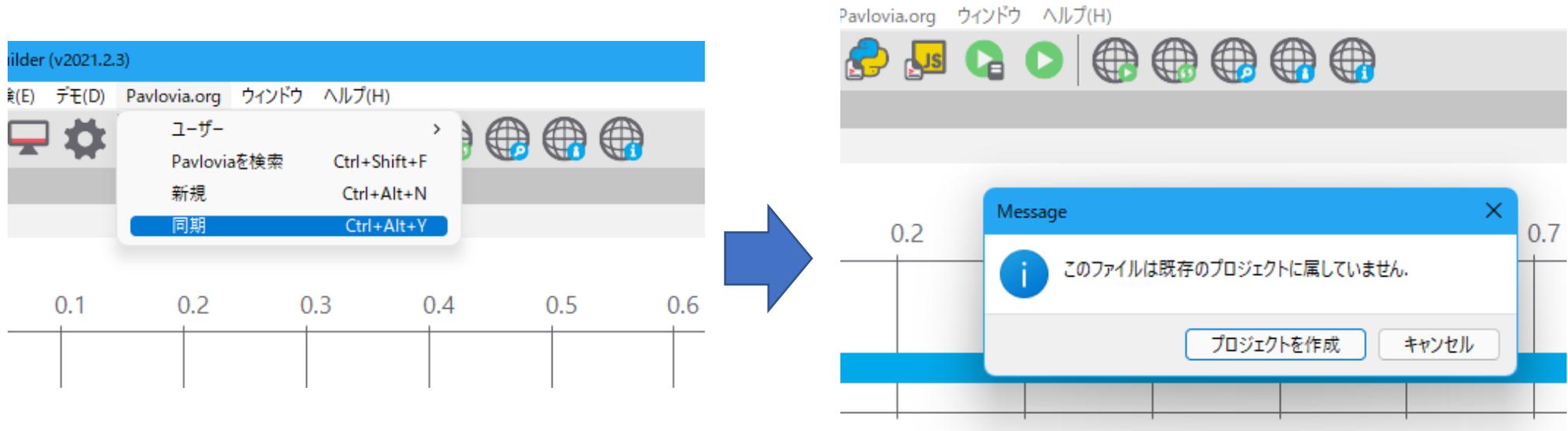
名前	更新日時	
data	2021/12/2	dataに不要なデータがある場合は空にしておくとうい
lib	2021/12/0	PsychoJSのライブラリ
cogpsy2021_stroop.js	2021/12/2	実験本体のJavaScriptコード
cogpsy2021_stroop.psyexp	2021/12/2	
cogpsy2021_stroop-legacy-browsers.js	2021/12/2	
conditions.xlsx	2021/12/0	
index.html	2021/12/2	実験参加用ページのHTML
readme.md	2021/12/2	

配布ファイルを展開して作業してきていれば、フォルダ内は以上のようにになっている。このフォルダをそのままプロジェクトにできる。



複数アカウントを切り替えて使う予定の人はRemember meをチェックしない方がよい。

Builderのメニューの「Pavlovian.org」から「ユーザー」→「Pavlovianにログイン」と選択し、表示されるダイアログにPavlovianのログイン情報を入力する。



すでにPC上でプロジェクトの内容が用意できている場合、Builderで実験を開いてメニューの「同期」でプロジェクトを作成できる。

初回は「このファイルはプロジェクトに属していない」と言われるので「プロジェクトを作成」をクリックする。

名前:

グループ/オーナー:

ローカルフォルダ: C:\Users\Hiroyuki\Dropbox\Experiment\Builder\_Exp\cogpsy2021\_stroop

説明:

タグ (カンマ区切り): PsychoPy, Builder, Coder

パブリックプロジェクト:

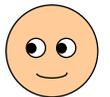
「名前」は参加者に伝える実験参加ページのURLの一部となるので、全角文字や入力が難しい文字は避ける。

「名前」を後から変更するのは難しいのでお勧めしません。  
変更したい場合はプロジェクトを作り直した方が早いです。



パブリックにしなくてもインターネットから実験に参加してもらえるので、とりあえずチェックしない。

こちらは後から簡単に変更できます。



変更をコミット

コミットする変更:  
New: 38 ファイル

変更の概要

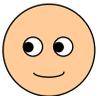
変更の詳細  
(任意)

キャンセル OK

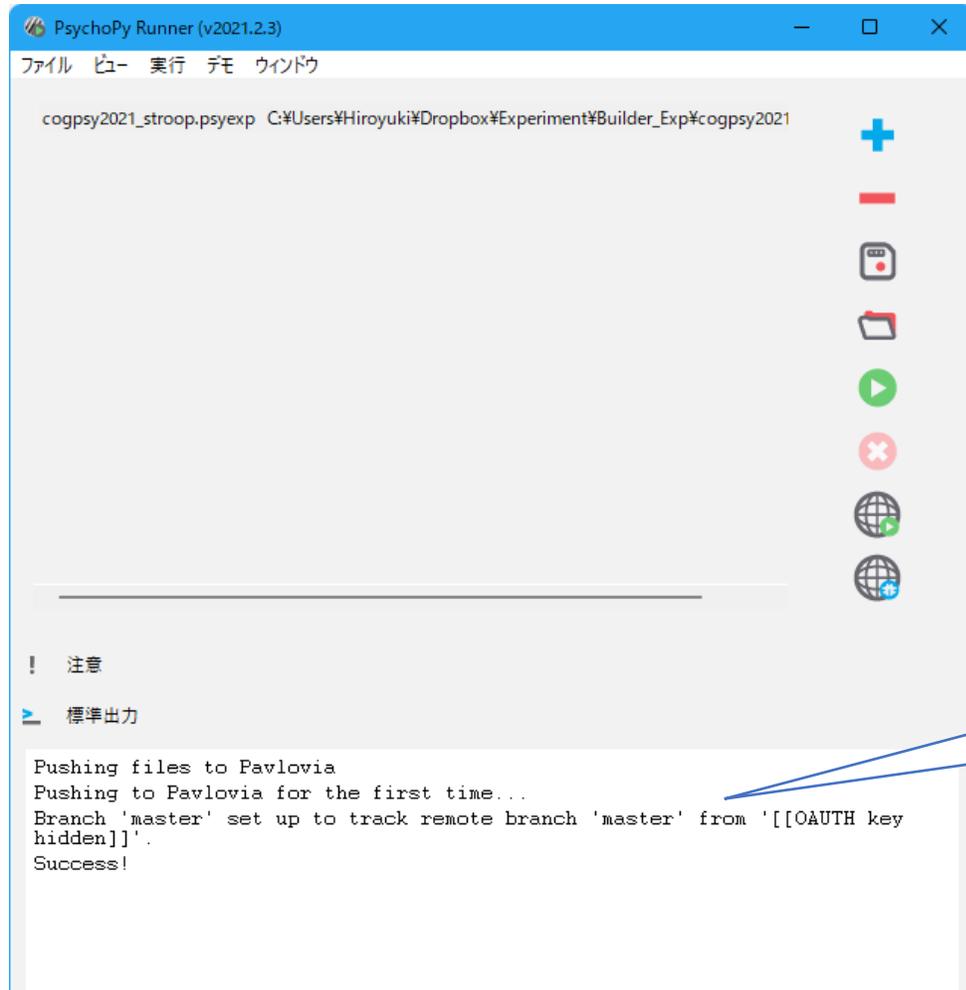
コミットの概要がわかるように1行でタイトルをつける。コミットの履歴はPavlovia上で確認できる (必須)。

必要に応じてさらに詳しく変更内容を書く。

日本語が使えるが、将来的に実験を公開するつもりなら英語で書くのがお勧め。



ファイルの追加や削除、修正などをプロジェクトに反映させる作業を コミット と呼ぶ。Pavloviaではすべての変更が記録される。



Runnerの標準出力に  
Success!と表示されている。

正常に終了したときにダイアログ等が表示されないののでわかりにくい  
が、Runnerの「標準出力」に簡単なメッセージが表示される。

1 Dashboard

2 Experiments

3

Docs Explore Dashboard Store Sign Out

Profile Experiments Participants Credits

Experiment status inactive, piloting, running

search experiment names, descriptions, designer names, and groups

<input type="checkbox"/>	Name	Namespace	Status	Visibility	Saving Format	Pavl	
<input type="checkbox"/>		nsogo	170291	inactive	private	CSV	0
<input type="checkbox"/>	cogpsy2022_stroop	hsogo	190810	inactive	private	CSV	0

先ほど作成したプロジェクトがある。  
クリックするとプロジェクトのページが開く。

不要になったプロジェクトの削除は  
このページでおこないます。  
(後述のGitLabからも可) 😊

ここからPavloviaでの作業である。ブラウザで<https://pavlovia.org>に  
ログインし「Dashboard」の「Experiments」を開く。  
**Builderは後で使うので終了せずそのままにしておく。**

The screenshot shows the Pavlovia project management interface for a project named 'cogpsy2022\_stroop' by user 'hsogo'. The interface is divided into several sections:

- Project Header:** Displays the project name, user profile, and creation/last update dates.
- Keywords:** A section for adding or editing keywords, with a 'View code' button.
- Status:** A section with three status cards: 'INACTIVE', 'PILOTING', and 'RUNNING'. The 'INACTIVE' card is highlighted with a red border.
- Recruitment:** A section for managing recruitment, including a 'Policy' dropdown (currently 'DISABLED') and a 'Constraints' section.
- Saving Results:** A section for configuring how results are saved, including a 'Format' dropdown (currently 'CSV') and a 'Save incomplete results' toggle.
- Sessions:** A section for viewing session data, including a 'Download results' button and two summary cards for 'Piloting' and 'Running' sessions.

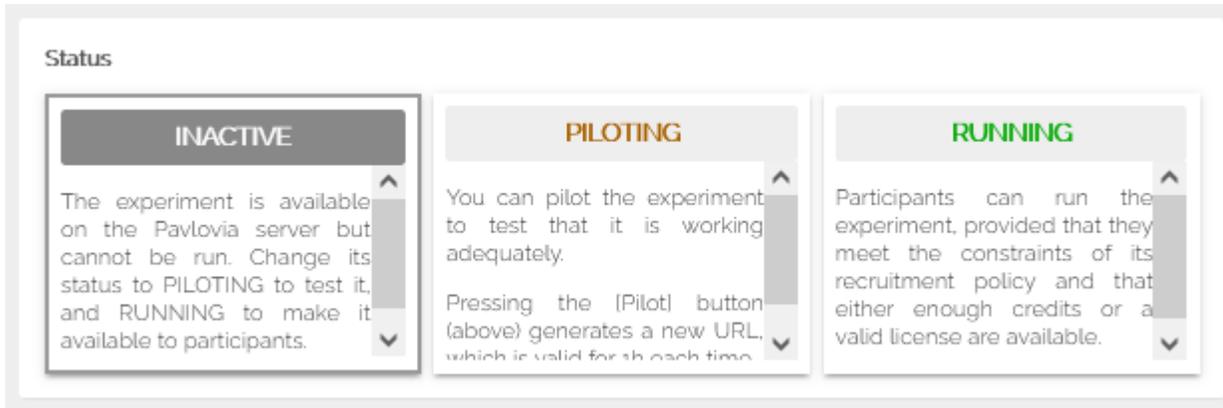
プロジェクトの  
コードページへ  
(詳細ページ)

プロジェクトの  
ステータス変更

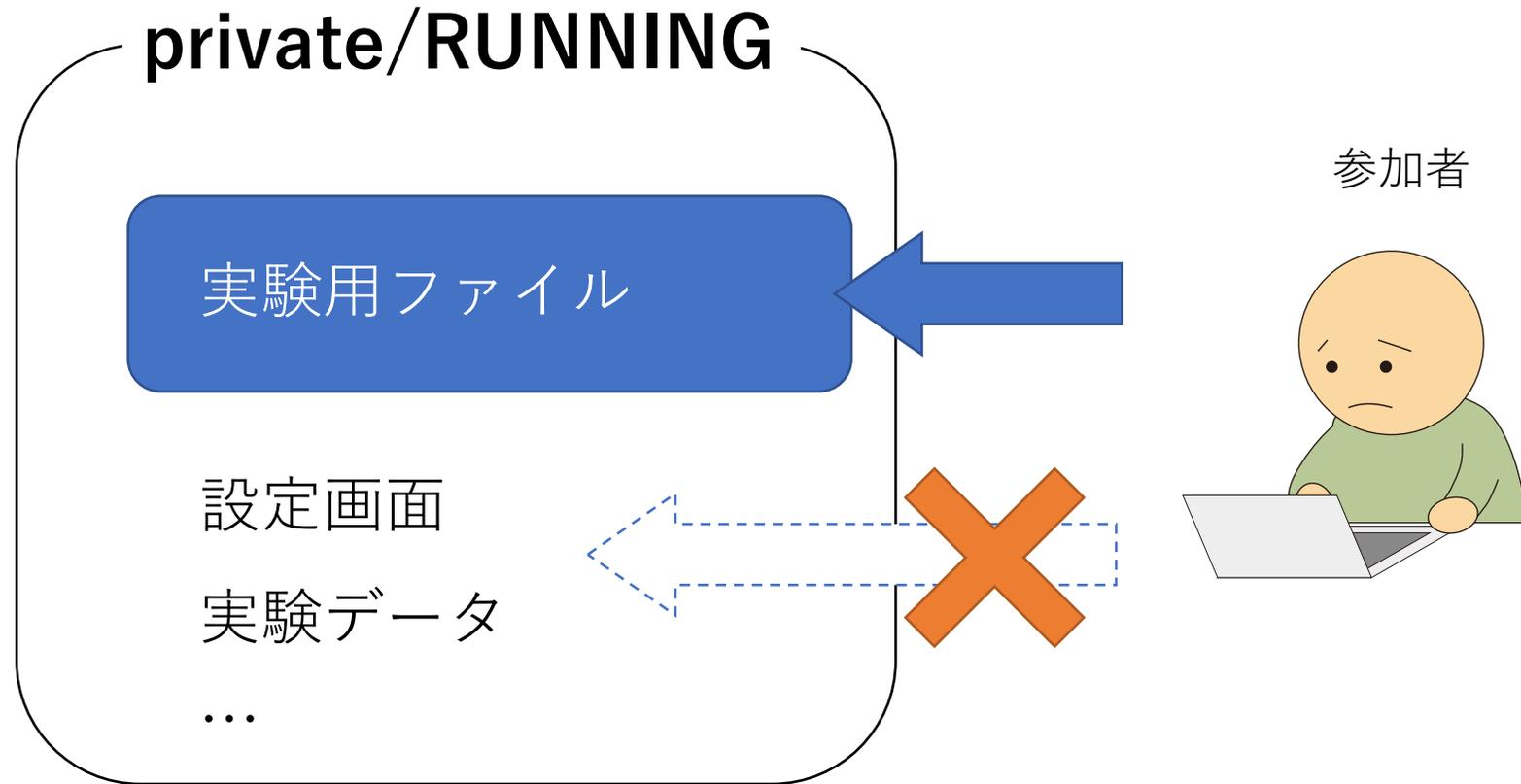
参加者募集に  
関する情報

データ保存形式  
などの設定

結果の  
ダウンロード



<b>INACTIVE</b>	実験を実行することはできない。研究終了後に実験とデータを公開する用途などに適している。
<b>PILOTING</b>	実験の動作確認ができる。一定時間実行可能なURLが発行され、実験終了後のデータは実行者の端末にダウンロードされる。
<b>RUNNING</b>	実験を実施する。データはPavloviaサーバ上に保存される。 <u>保存時にクレジットが消費される</u> （機関ユーザはクレジット不要）。



プロジェクトがプライベートでも、ステータスがRunningなら参加者は実験にアクセスして参加することができる。

Status

**INACTIVE**

The experiment is available on the Pavlovia server but cannot be run. Change its status to PILOTING to test it, and RUNNING to make it available to participants.

**PILOTING**

You can pilot the experiment to test that it is working adequately.

Pressing the [Pilot] button (above) generates a new URL, which is valid for 1h each time.

**RUNNING**

Participants can run the experiment, provided that they meet the constraints of its recruitment policy and that either enough credits or a valid license are available.

Running Mode

CREDIT LICENSE

**You do not have enough credits assigned to this experiment to run it by consuming them.** pilot your experiment but participants will not be able to run it until you assign credits to experiment or your institution obtains a license.

0 620

assigned	available	reserved	consumed
0	620	0	0

Update assigned credits Release reserved credits

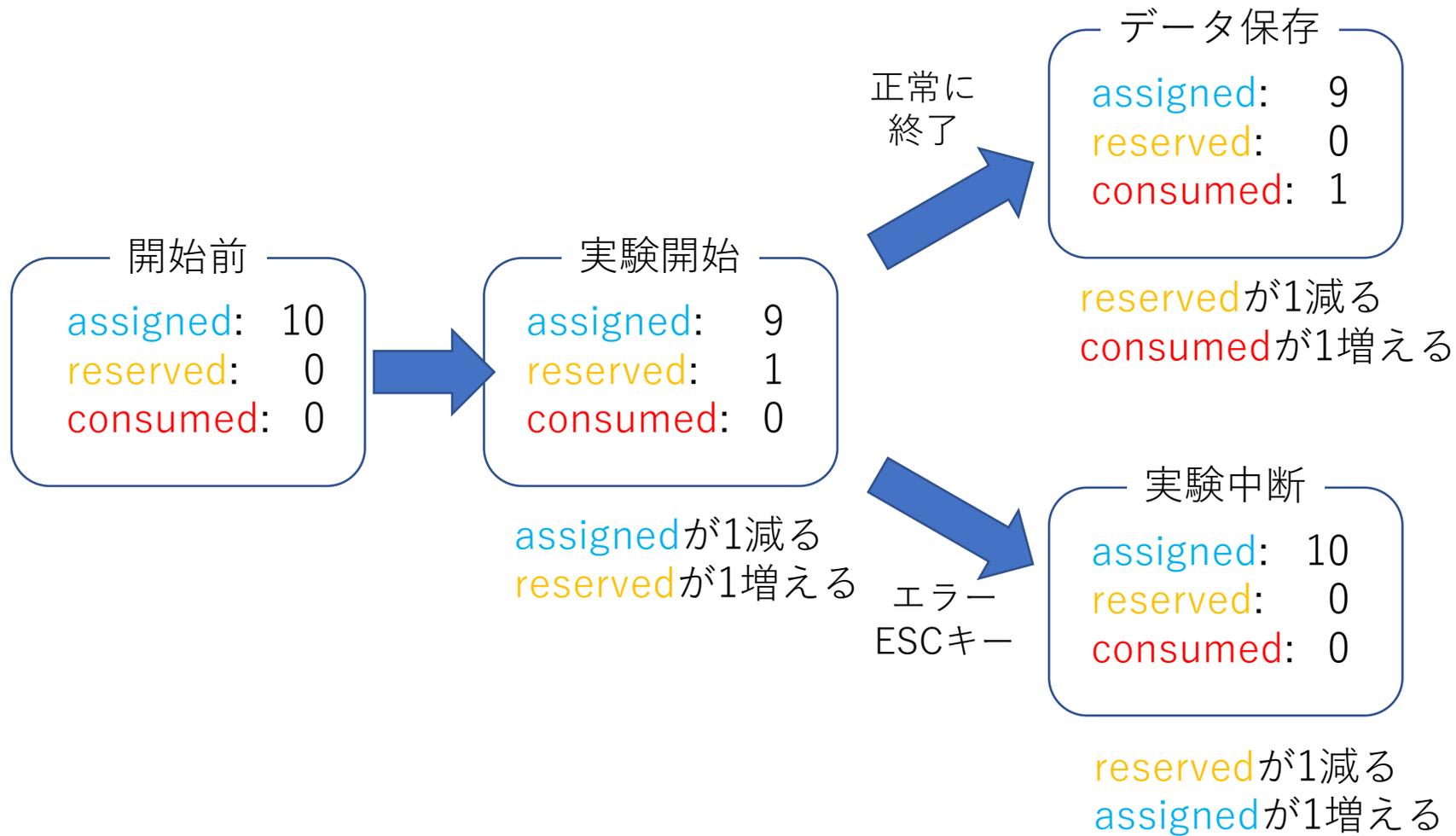
1 Running

2 クレジットを割り当て

3 割り当てを更新

実験を実施するときにはステータスをRunningに変更する。

機関ライセンスでカバーされないユーザの場合はクレジットを割り当てる必要がある。



実験が開始されるとクレジットは1減って予約された状態となり、実験が終了した際に初めて消費される。実験が中断された場合は予約が解除されてクレジットは元に戻る。

1  
LISENCE

**Status**

<b>INACTIVE</b> The experiment is available on the platform but cannot be run. Pressing the [Pilot] button (above) moves the experiment to PILOTING to allow participants to run it. Pressing the [Run] button (above) moves the experiment to RUNNING to make it available to participants.	<b>PILOTING</b> You can pilot the experiment to test that it is working adequately. Pressing the [Pilot] button (above) generates a new URL, which is valid for 1h each time.	<b>RUNNING</b> Participants can run the experiment, provided that they meet the constraints of its recruitment policy and that either enough credits or a valid license are available.
---	--	---

**Running Mode**

CREDIT **LICENSE**

This experiment is covered by a license. All documents have been approved. Participant can run it freely without credits being consumed.

license number: [REDACTED]  
institution: University of Nottingham  
expiration date: 2020-09-30  
license manager: [REDACTED]

機関ライセンスでカバーされるユーザであれば、クレジットを割り当てずに実験を実行することができる。

クレジット割り当てバーの代わりにライセンス情報が表示されます。



Recruitment

Policy

URL

Constraints

Anyone meeting the experiment constraints can run it by opening the experiment URL provided that either enough credits or a valid license are available.

Url: [https://run.pavlovia.org/hsogo/cogpsy2021\\_stroop](https://run.pavlovia.org/hsogo/cogpsy2021_stroop)

実験参加用URL  
これを参加者に伝えてアクセスしてもらう。

Saving Results

Format

CSV  DATABASE

Experimental results are saved in .csv files, which are stored on the Pavlovia server and in the experiment's GitLab repository.

You can retrieve results by pressing the [Download results] button, or downloading from the repository, or by git pulling from your local repository.

Incomplete Results

Save incomplete results

All results will be saved, even when the participant does not complete the experiment.

If the experiment's running mode is CREDIT, a credit will be consumed at the end of every session, even those prematurely interrupted.

**⚠ This is an experimental feature currently available for:**

- PsychoPy/PsychJS: 'unversioned', 3.2, 3.2.0, 3.2.2, and all 2020.x, 2021.x, 2022.x
- jsPsych: all versions
- lab.js: all versions

中断された実験のデータを保存するか否かを選択する。チェックすると実験終了時に通信トラブルがあってもデータが残る可能性があるが、何度も中断する参加者がいると割り当てたクレジットがその都度消費されてしまう。

データ保存形式。  
CSVならプロジェクトにCSV形式で保存される（Pythonで実行するときと同じ）。  
DATABASEならプロジェクト外のDBに保存される。こちらに設定しておくで、研究終了後にプロジェクトを公開しても、データは公開されない（実験実施時のログは公開される）。

サーバに保存されている実験データをZIP形式でまとめてダウンロードする。CSV形式の場合はプロジェクトを同期すればデータをダウンロードできるがDATABASE形式ならこの操作が必要。

Sessions

Download results

Piloting	Running
completed- 2 (50.0%) aborted- 2 (50.0%) last session: 2021-06-04 07:27:38	completed- 20 (71.4%) aborted- 8 (28.6%) last session: 2021-07-24 06:15:26

最後まで行われた回数、中断された回数、最後に実行された時刻の情報。



GitLabは後で少しだけ触れます。



PavloviaプロジェクトはGitLabというサービスによって管理されている。  
View codeをクリックするとGitLabのプロジェクト管理画面に移動できる。  
プロジェクトのパブリック/プライベートの切り替えや、プロジェクトへの  
メンバー追加といった作業はそちらでおこなう。

# プロジェクトの更新

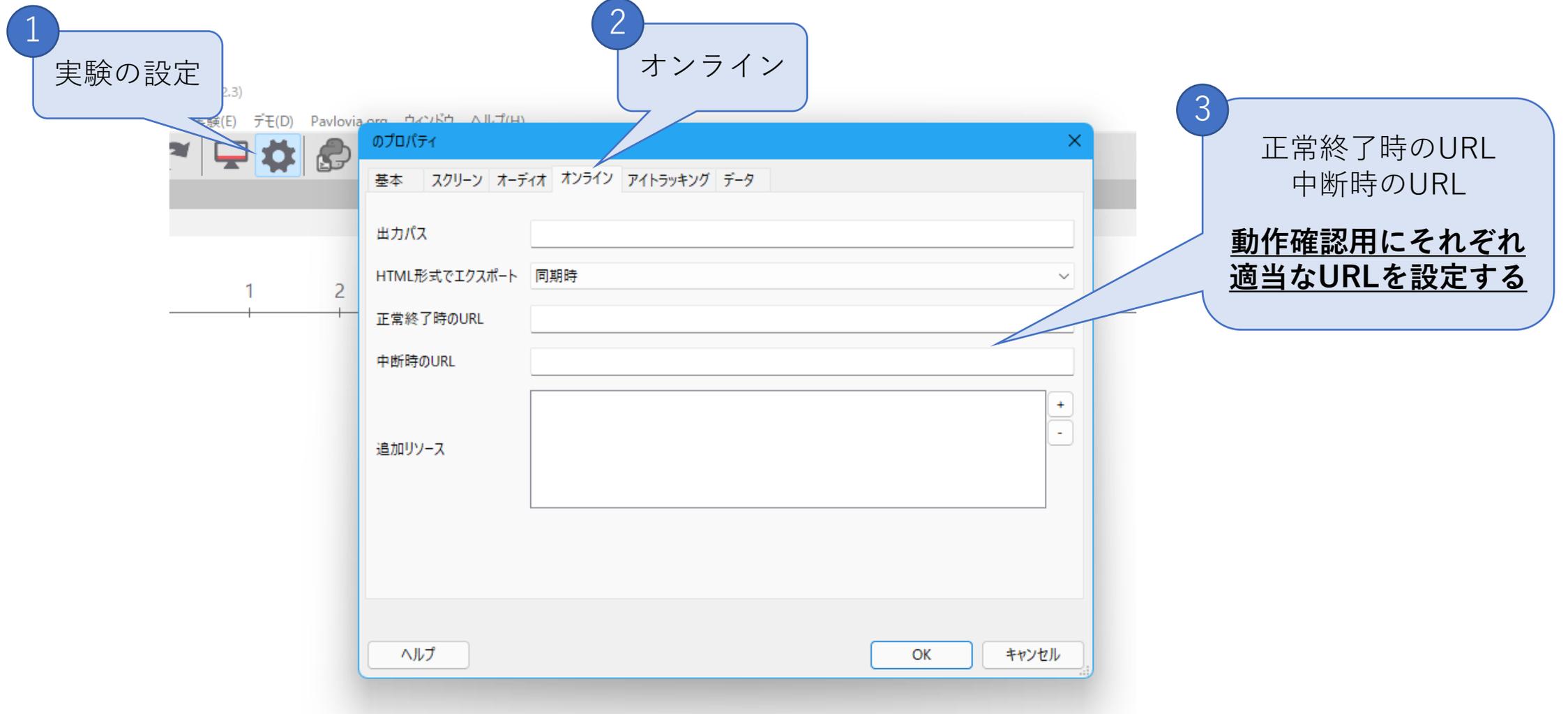
プロジェクトの更新の練習を兼ねて、実験終了時に指定したURLへ移動するようにしてみよう。

この機能を利用すると、実験終了時にアンケートフォームへ移動したり、実験を中断した人に専用のメッセージを表示したりすることができるほか、SONA System(<https://www.sona-systems.jp/>)などの**実験参加者募集システムと連携**させることもできる。

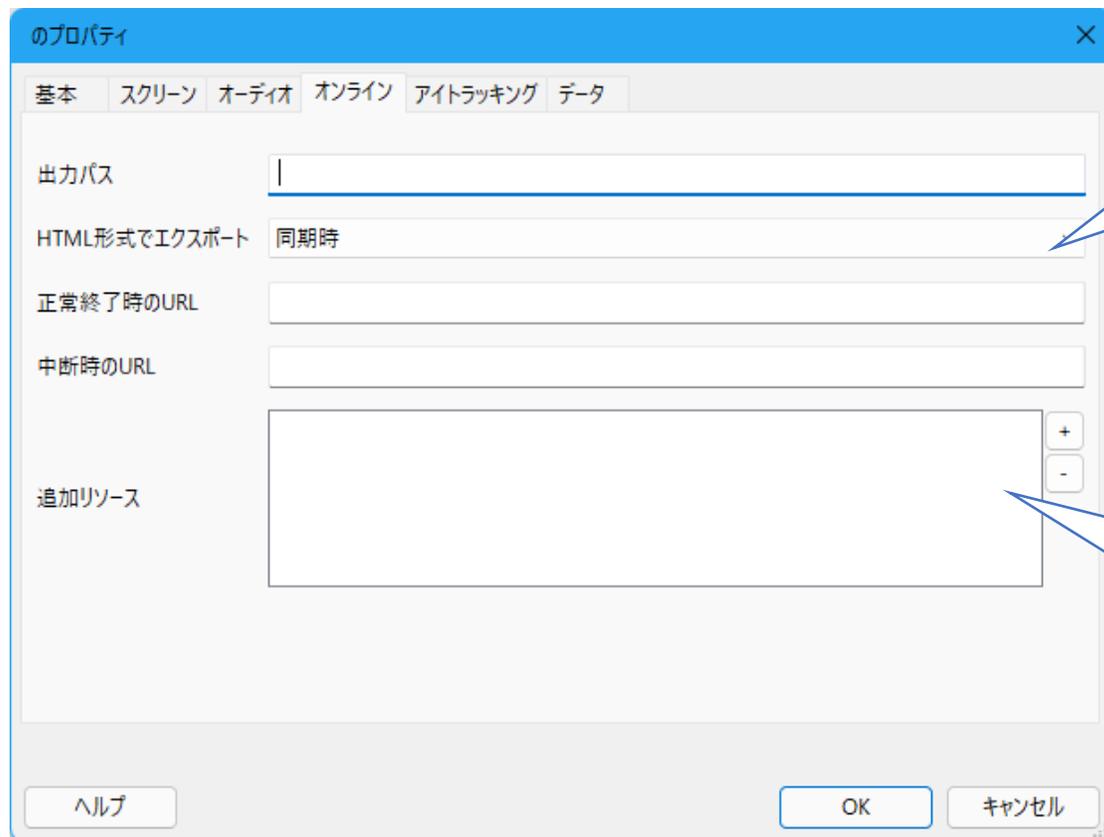
Pavloviaにログインしたブラウザは、後で使うのでそのままにして  
おいて、PsychoPy Builderに戻る。

先ほど同期をした後にBuilderを閉じずに  
残していればすぐに戻れるはず。





「実験の設定」ダイアログの「オンライン」タブを開く。  
ここで実験中断時、正常終了時の移動先URLを指定できる。



エクスポートするタイミングの指定

実験に必要なリソースの指定

Builderによる自動検出がうまくいかない場合に有効。

ちなみにこの「オンライン」タブではHTML形式でのエクスポートをおこなうタイミングや、実験に必要な画像ファイル等のリソースを指定することができる。

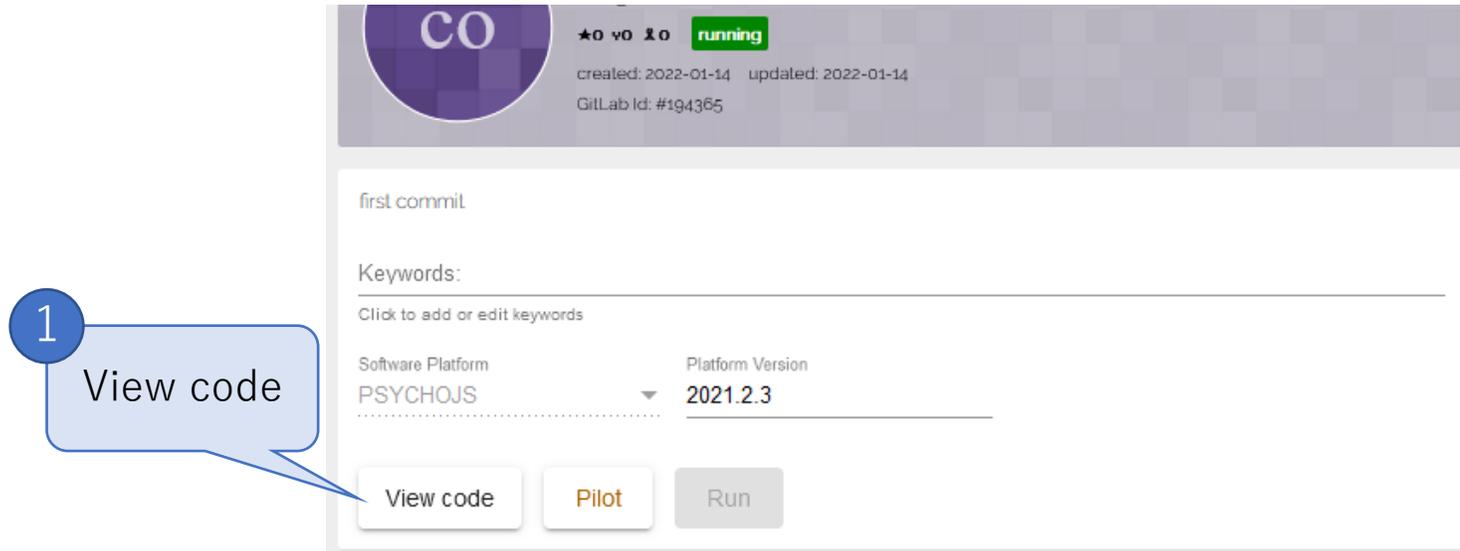
実験中断時、正常終了時のURLを指定したら、ふたたびHTML形式へのエクスポートとPavloviaへの同期をおこなう。

すでに先ほどBuilderでPavloviaへのログイン設定をしているので、その手順を飛ばして同期の操作ができる。

標準では「同期時」にHTML形式でのエクスポートをするよう設定されているのでエクスポートの手順は不要ですが、筆者はいつも手作業でエクスポートしています。



手順を忘れた人はスライド27-31を復習しましょう。



同期できたらPavlovianに戻り、Pavlovian側に変更が反映されていることを確認しよう。

先ほどのView codeボタンをクリックしてGitLabへ移動する。

You need to sign in or sign up before continuing.

## Sign In/Register



Welcome to Pavlovia's Project Repository

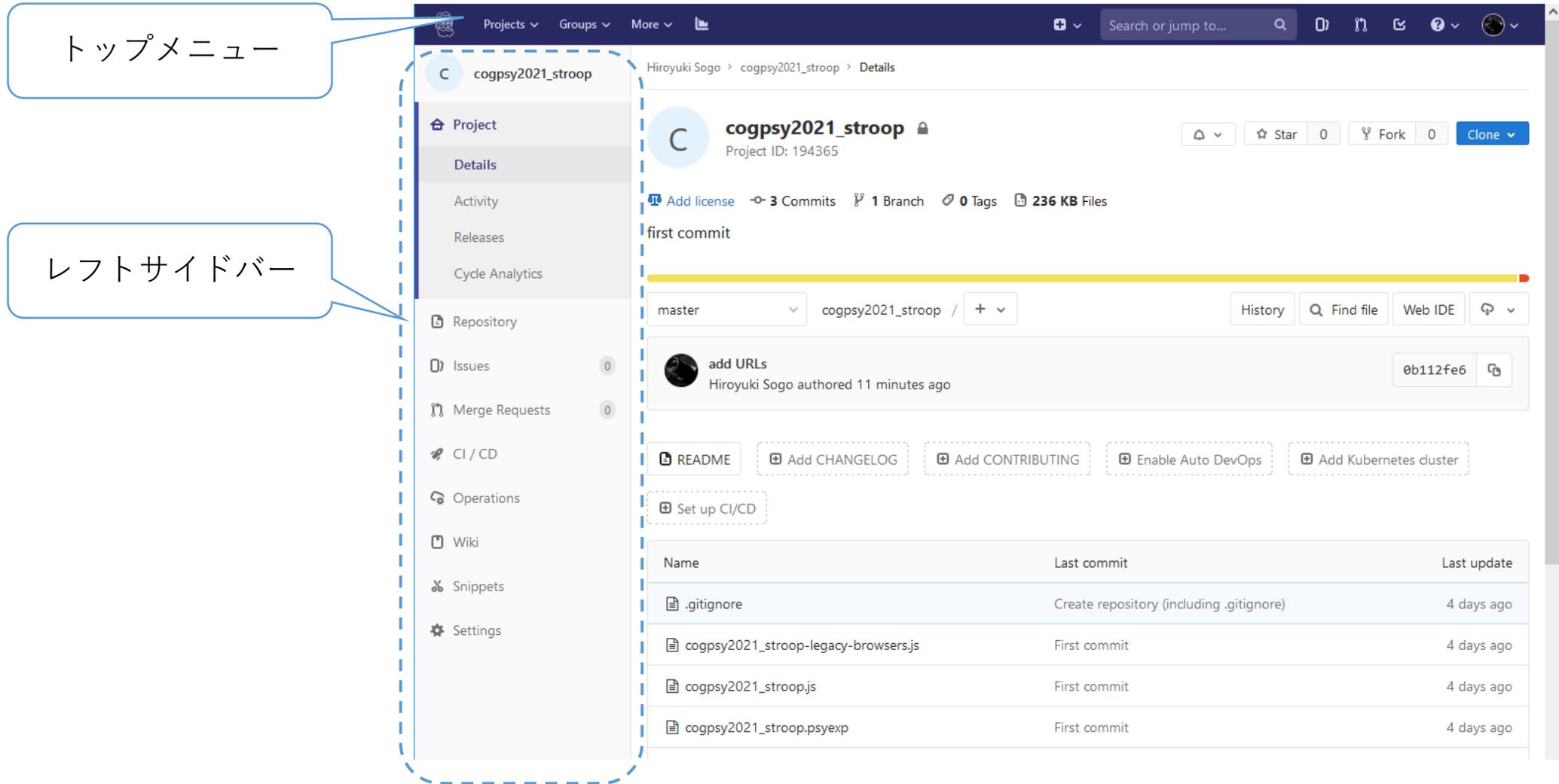
Sign in	Register
Username or email	
<input type="text" value="hsogc"/>	
Password	
<input type="password" value="....."/>	
<input type="checkbox"/> Remember me	<a href="#">Forgot your password?</a>
<input type="button" value="Sign in"/>	

1

ログイン情報は  
Pavloviaと同じ

GitLab上のプロジェクトページがブラウザで開く。

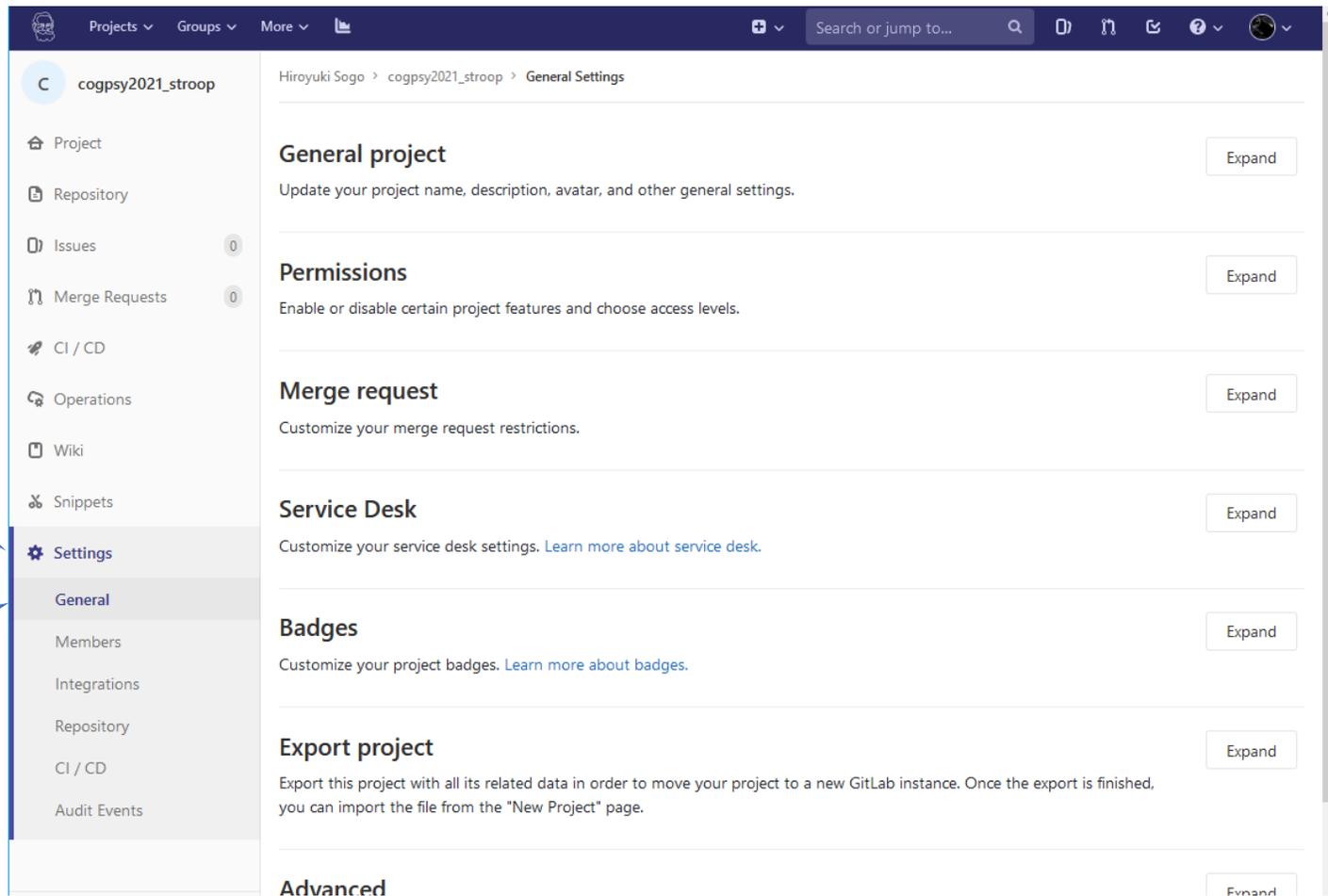
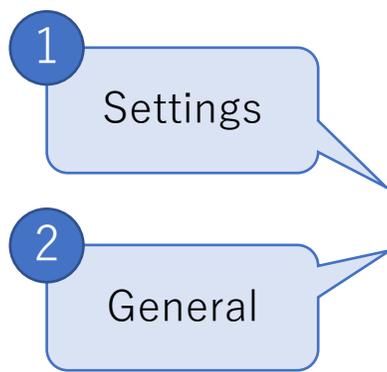
まだGitLabにログインしていない場合、GitLabのログイン画面が表示されるので、**Pavloviaにログインするときと同じユーザー名とパスワードを入力する。**



Pavloviaでは実験を実施するための操作をしたが、GitLabではプロジェクトの公開設定やメンバー管理などをおこなう。

The image shows a screenshot of a project activity page. On the left, a sidebar contains a list of project-related items: Project, Details, Activity, Releases, Cycle Analytics, Repository, Issues (0), Merge Requests (0), and CI / CD. Two callout boxes are present: one labeled '1' pointing to the 'Project' item, and another labeled '2' pointing to the 'Activity' item. The main content area displays a list of activities for the project 'cogpsy2021\_stroop'. The activities include: 'Pushed to branch master' (14 minutes ago), 'Pushed new branch master' (4 days ago), and 'Created project Hiroyuki Sogo / cogpsy2021\_stroop' (4 days ago). A callout box on the right points to the first activity, stating '先ほどのURL追加が記録されている' (The URL addition from before is recorded). The top navigation bar includes 'Projects', 'Groups', 'More', and a search bar.

レフトサイドバーのProject→Activityでプロジェクトの変更履歴を確認できる。実験データの保存日時やファイルの修正などがすべて記録される。

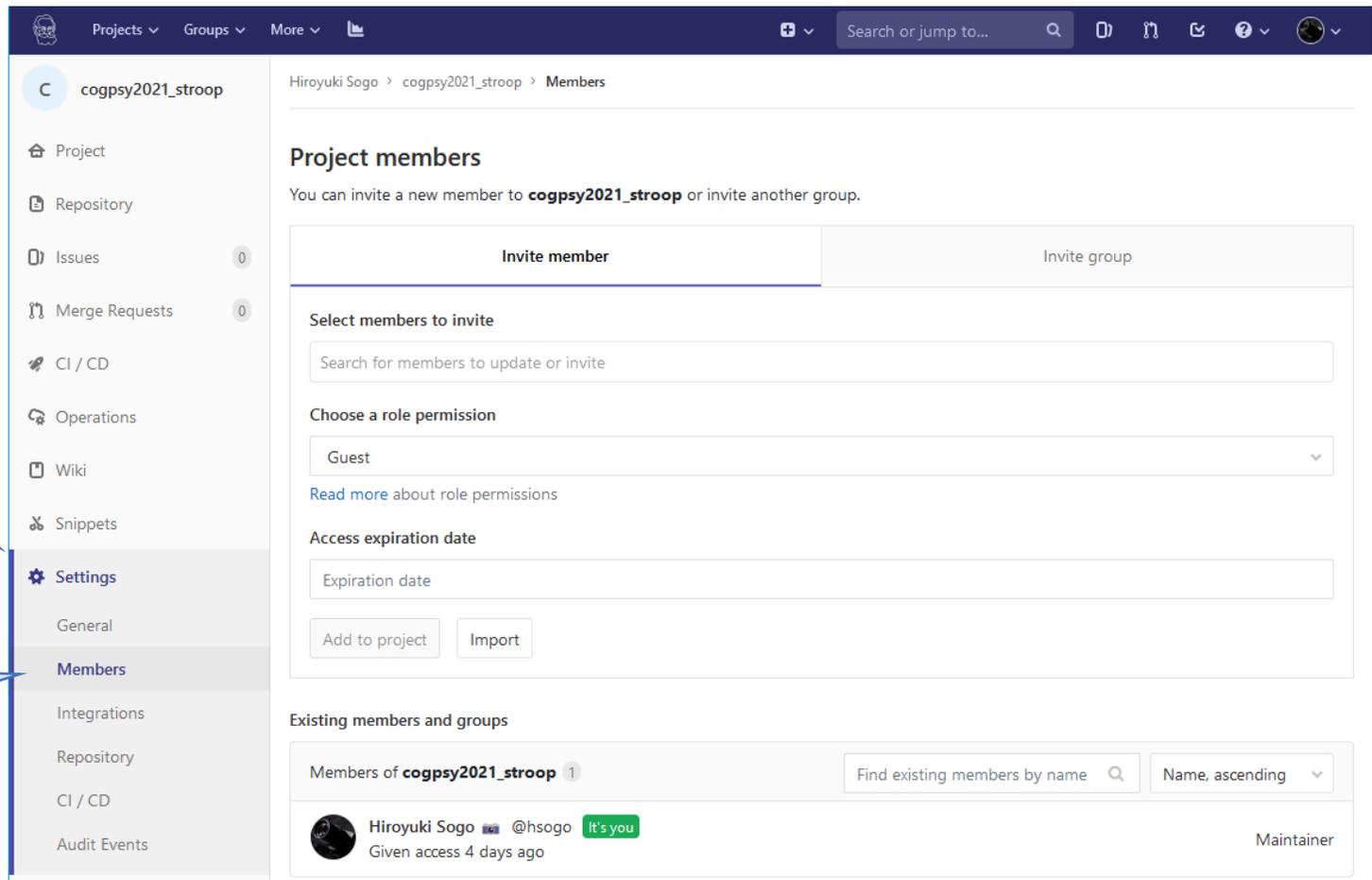


Settings→GeneralでプロジェクトのPrivate/Public切り替え、削除、アーカイブ化などをおこなえる。

他にもいっぱいできることがありますが  
多くはソフトウェア開発向けかな…



- 1 Settings
- 2 Members



Settings→Membersでプロジェクトにメンバーを追加できる。  
他の研究者と共同で実験を実施したい場合に便利。

The screenshot shows the Pavlovia interface for an experiment named 'piloting'. At the top, it displays 'CO', '★0 v0 10 piloting', and dates 'created: 2022-01-14 updated: 2022-01-19' along with 'GitLab Id: #194365'. Below this, there is a 'first commit' section and a 'Keywords:' field. A 'Software Platform' dropdown is set to 'PSYCHOJS' and the 'Platform Version' is '2021.2.3'. There are three buttons: 'View code', 'Pilot', and 'Run'. A blue callout box with the number '2' points to the 'Pilot' button and is labeled 'Pilot'. Below the buttons is a 'Status' section with three columns: 'INACTIVE', 'PILOTING', and 'RUNNING'. The 'PILOTING' column is highlighted in orange. A blue callout box with the number '1' points to the 'PILOTING' column and is labeled 'PILOTING'. The 'INACTIVE' column text reads: 'The experiment is available on the Pavlovia server but cannot be run. Change its status to PILOTING to test it, and RUNNING to make it available to participants.' The 'PILOTING' column text reads: 'You can pilot the experiment to test that it is working adequately. Pressing the [Pilot] button (above) generates a new URL, which is valid for 1h each time.' The 'RUNNING' column text reads: 'Participants can run the experiment, provided that they meet the constraints of its recruitment policy and that either enough credits or a valid license are available.'

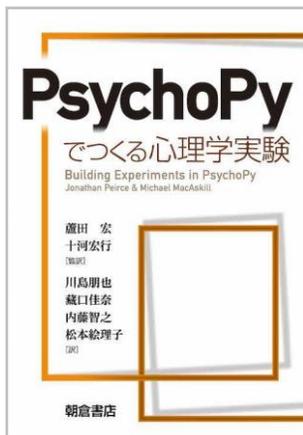
実験中断・正常終了時の移動先URLを設定した効果を確認しよう。  
ローカルデバッグでも確認できるが、ここではPavloviaに移動して  
Pilotingを試してみよう。



数試行おこなったあと、ESCキーを押すと全画面表示が解除される。さらにESCキーを押すと中断時に移動するように指定したページが表示される。

時間があれば、最後の試行まで実行するとデータのダウンロードが行われたあとに指定したページが表示されることを確認する。

# 書籍紹介



## 「PsychoPyでつくる心理学実験」朝倉書店

PsychoPy Builderによる基礎的な実験の作成から、専門的な内容までを網羅した一冊。PsychoPyのバージョンがかなり古い(1.90系)のが難点だが、大部分の内容現在のバージョンにも通用する。

今回オンライン実験化したプロジェクトは本書第6章の実験。

2022年  
5月1日  
発売予定

## 「PsychoPy/Pavloviaによるオンライン実験」朝倉書店

PsychoPy BuilderでPavlovia用の実験を作成するためのノウハウを紹介。今回のワークショップより高度な内容（プロジェクトの管理、Google Formとの連携、マウスの活用、無作為化、文字入力、スライダーの使い方、PythonとJavaScriptの両方で使える関数など）を扱っている。