

第一回 実験実習

高橋 龍



自己紹介



名前：高橋龍

学年：博士後期課程 1 年（大坪研）

出身：宮城県石巻市

最近はまっているもの：パン作り



本講義の目的

メイン

- oTree を用いて、集団実験を自分で作れるようになる

サブ

- Python（プログラミング言語）が実行できる環境にする
- プログラミングに対する抵抗感を下げてもらう
- 自分なりの仮説をどうやって実験に落とし込めばよいかわかるようになる
- 大坪研に院進してもバリバリやっていく素地を作る

まずはやってみよう



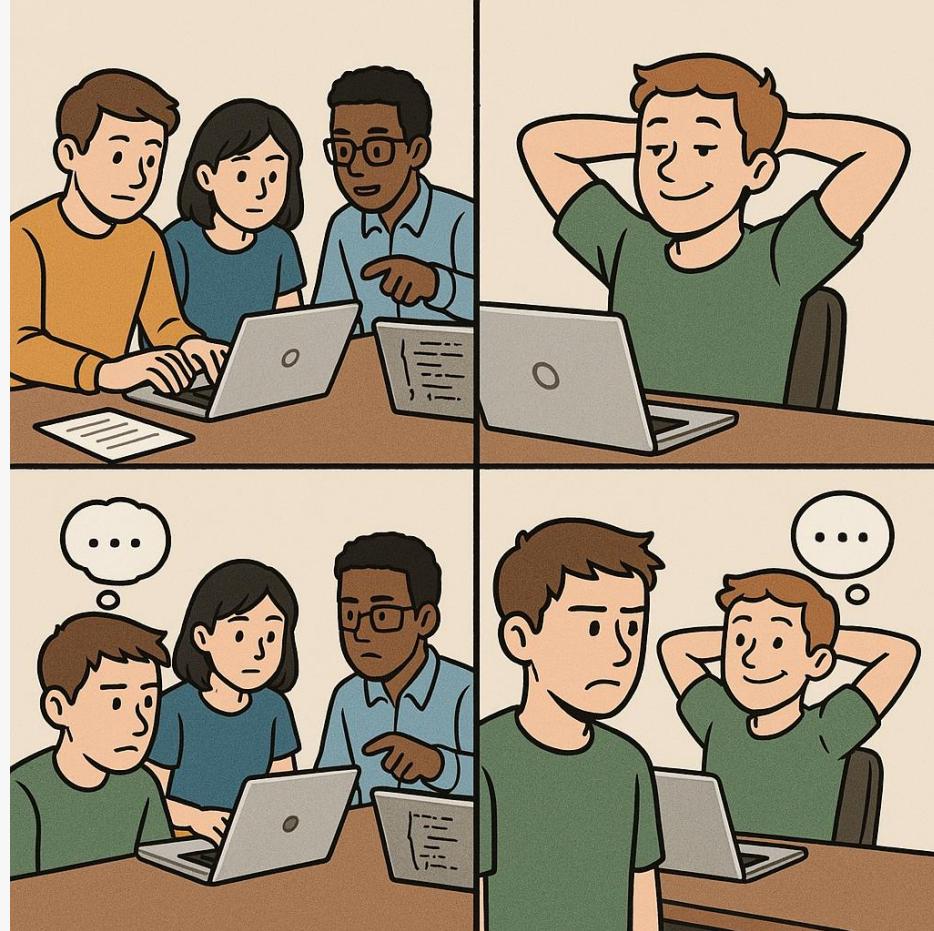
実験が終わるまでは、
以降のページは見ない

実験が終わるまでは、
以降のページは見ない

実験の説明

実験の目的

周囲の意思決定の結果は、自分の意思決定に影響を与えるか？



- 例えば、チーム全体で一つのプロジェクトを行う
- 例) 複雑な実験プログラムを分担して作成するなど
- さぼった方が実は得である状況で、仲間の頑張りが見えていた方が、自分も頑張る？
- さぼった方が実は得である状況で、さぼっている人 (i.e., 得をしている人) を見ると、頑張らなくなる？

目的を達成するアイディアとは

- 仲間の頑張りが見える状況 vs. 見えない状況
- 仲間の報酬が見える状況 vs. 見えない状況

アイディアを形にするためのデザインとは

	頑張りが見える	頑張りが見えない
報酬が見える	切磋琢磨	純粋な成果主義
報酬が見えない	プロセス重視	孤独

デザインを実現するためのツールとは

The screenshot shows the oTree documentation website. The sidebar on the left contains links to various documentation sections such as 'oTreeのインストール', 'Pythonについて', 'チュートリアル', 'コンセプト', 'モデル', 'ページ', 'テンプレート', 'フォーム', 'マルチプレイヤーゲーム', 'アプリケーションとラウンド', '処理の割り当て', 'タイムアウト', 'ポット', 'ライブページ', 'サーバ設定', '管理者', 'ルーム', '通貨 (Currency)', 'MTurk & Prolific', and 'その他'. The main content area displays the 'Docs' page for the 'no-self' format. It includes a search bar, a navigation bar with 'Docs > oTree', and a message about the new format. Below this is a large image of a building with a dome, labeled 'oTree A modern open platform for social science experiments'. At the bottom of the page are links for '中文 | 日本語' and a 'デモ' section with a URL.

oTree
latest
Search docs

oTreeのインストール
Pythonについて
チュートリアル
コンセプト
モデル
ページ
テンプレート
フォーム
マルチプレイヤーゲーム
アプリケーションとラウンド
処理の割り当て
タイムアウト
ポット
ライブページ
サーバ設定
管理者
ルーム
通貨 (Currency)
MTurk & Prolific
その他

Docs » oTree

These are the docs for people using the new no-self format. Docs for the models.py format are here. [What changed?](#) | [How to upgrade](#)

oTree

A modern open platform for social science experiments

中文 | 日本語

デモ

<https://www.otreehub.com/projects/?featured=1>

- oTree という心理学実験プログラムを作るためのツールキット
- oTree を使って、提示する情報をランダム化

本講義の立場：巨人の肩に乗ることを徹底

- 先人たちが作ったツールキット（道具箱）を使う
- 先人たちが開発したソフトウェアを使う
- 一人で全部を経験する

具体的には

- oTree を使う
 - 集団実験に必要なものをほとんど用意してくれたツールキット
 - 同期通信，データベース操作，ページ遷移とそのルール，ページの装飾 etc.
 - 全部やるのは本当に大変（私は半年かかりました）
- 生成AIを使い倒す
 - これを使えるかどうかでスキルの伸びも、楽しさも変わります
 - どう使うかは後ほど
- 他人と分担しない
 - デザイン設計から実験プログラムの公開まで全部一人でやる
 - ただし，相談は積極的にしましょう！

本講義の進め方



- 全体的な説明 -> 個人で開発
- 私（高橋）がその会でやることの全体像を簡単に説明
- 説明が終わったら、設定した課題をそれぞれ解く
- 各自 ChatGPT 等と会話しながら進める



「ChatGPT に何を聞けばいいのかわからな
い」
「何がわからないのかわからない」
という場合は、積極的に私（高橋）に質問し
てください

立場別の目標

1. プログラミングをほぼ触ったことがない
2. oTree は初めて (Rとかは経験あり)
3. oTree で実験を 1, 2 回作ったことがある
4. oTree は十分, テストコードは経験なし
5. Framework なしで集団実験アプリを作れる

1. プログラミングをほぼ触ったことがない

- 本講義が想定している受講者です
- 講義と同じペースで進めながら、わからないところはすぐに聞きましょう
- この段階では、わからないことをわからうとする時間がもったいないです
- 何も恥ずかしいことはありません。積極的に聞いてください。
- とはいえ、質問するのはハードルが高いと思います
- ChatGPT はどんな質問でも、何度も聞いても答えてくれます
 - 「なんか集団実験アプリを作れとか言われたんだけど、そんなの言われても意味が分かんない。どうすればいい？」
 - 「中学生でもわかるように説明して」
- 積極的に使いましょう

2. oTree は初めて (Rとかは経験あり)

- 本講義で最も多いと想定されている人です。
- 講義より少し早いペースで進めることを目標にしましょう
- ChatGPT にわからないことをどんどん聞きましょう
 - 聞き方がわからない、呆然とした、5分以上同じところで躊躇している場合、私（高橋龍）に質問してください。
- 自分で調べてみる → わからないところを人に尋ねて解決できるようになります

3. oTree で実験を 1, 2 回作ったことがある

- 本講義では少数だが受講していることを想定している人です。
- NotebookLM を活用し、リファレンスから自分でアプリを作れるようになります。
 - NotebookLM に oTree 公式のURLをソースに指定して、oTree References を作ります
 - 実現したいことがあったら、つど尋ねましょう
- Qiita や Zenn を見てコードを書くのは禁止！！！
- コピペして開発することから卒業しましょう

4. oTree は十分， テストコードは経験なし

- 本講義でほぼいない（が、いる可能性も捨てきれない）人です。
- TDD（テスト駆動開発）ができるようになります。
- oTree の公式リファレンスを参照し，ボットの使い方をマスターしてください。
- ボットのテストコードを書く → プロダクトコードを実装する，という流れで過去に作成したアプリを再現してみましょう

5. Framework なして集団実験アプリを作れる

- 本講義を受講する必要がない (i.e., 高橋の指導能力を超えてる) 人です。
- MCP サーバーを利用して、自然言語のみで集団実験アプリを作ってみましょう。
- Cline を使って Agent にアプリを作らせてみましょう
- Claude desktop 等を使って MCP サーバーを構築し、自然言語で作りたいものを作らせる体験をしてみましょう。

授業計画（予定）

1. 4/10(木) : 実験の実施, 授業の流れの説明, 環境構築
2. 4/17(木) : 実験の説明と解析結果, simple_pd の作成
3. 4/24(木) : simple_pd のデプロイ
4. 5/1(木) : simple_sd の作成とデプロイ, 自作アプリの宿題
5. 5/8(木) : simple_tg の作成とデプロイ, 自作アプリの宿題
6. 5/15(木) : 自作アプリ案の相談と実装 (1)
7. 5/22(木) : 自作アプリ案の相談と実装 (2)
8. 5/29(木) : 自作アプリ案の実装 (3)
9. 6/5(木) : 自作アプリ案の実装 (4)
10. 6/12(木) : 実験の実施 (1)
11. 6/19(木) : 実験の実施 (2)
12. 6/26(木) : 分析の相談
13. 7/3(木) : 結果発表

環境構築

ChatGPT に登録する

1. <https://chatgpt.com/> からアカウント登録をする

python 等のインストール (Windows 編)

Windows の場合 ([参考](#))

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser  
Invoke-RestMethod -Uri https://get.scoop.sh | Invoke-Expression  
scoop install git  
scoop bucket add extras  
scoop update  
scoop install python  
scoop install vscode
```

python 等のインストール (Mac 編)

Mac の場合 ([参考](#))

1. Homebrew がインストールされていない場合は以下を実行

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

2. 以下を実行して、エラーがでなければOK！

1. デフォルトで入っているはず

```
python --version
```

uv のインストール

1. 以下を実行する

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

2. 以下を実行して、 uv 0.X.X となれば OK

```
uv --version
```

参考

Git Bash のインストール

Windows の人は先ほどインストールしたので OK

Mac の人は `brew install git` でインストールする

```
git --version
```

GitHub アカウントの作成

<https://github.com/> にアクセスして、アカウントを作成する

VS Code をインストール

Windows の人は先ほどインストールしたので OK

Mac の人は `brew install --cask visual-studio-code` でインストールする ([参考](#))

```
code -v
```

VS Code の拡張機能をインストール

- [Prettier](#)
コードを自動整形してくれるフォーマッター
- [Black Formatter](#)
Python コードを自動で美しく整形
- [Indent-Rainbow](#)
インデントに色をつけて視認性をアップ。
- [Pylance](#)
Python 開発の補完と型チェックを強化。
- [Python](#)
VSCode での Python 開発をサポート。
- [Gemini Code Assistant](#) (or [GitHub Copilot](#))
AI によるコード補完・提案アシスタント。
- [vscode-icons](#)
ファイルアイコンを分かりやすくカラフルに。
- [Python Indent](#)
Python のインデント処理をより自然に。
- [HTMLHint](#)
HTML の文法チェックを自動でしてくれる
- [HTML CSS Support](#)
HTML と CSS の補完を強化。
- [Rainbow CSV](#)
CSV ファイルをカラフルに表示・編集。
- [Auto Close Tag](#)
HTML や JSX タグを自動で閉じてくれる。
- [Auto Rename Tag](#)
開始タグ・終了タグの名前を同時に編集。
- [Bootstrap 5 Snippets](#)
Bootstrap 5 の便利なスニペットを提供。
- [Django](#)
Django テンプレート・構文のサポートを追加

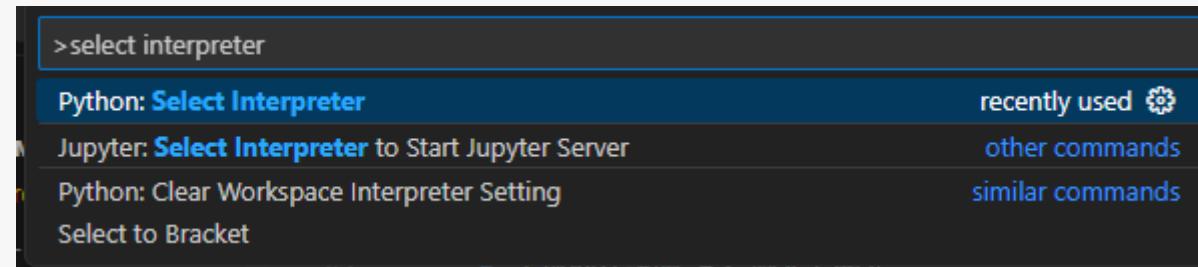
テンプレートとなるリポジトリをクローン

以下のコマンドを実行する

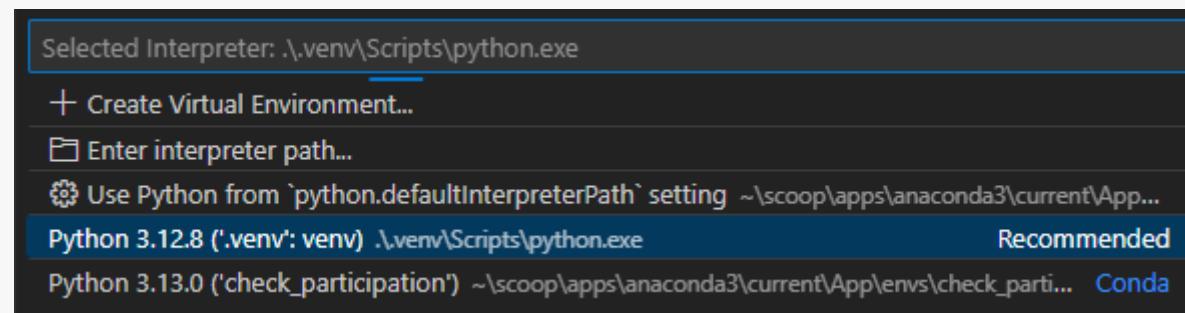
```
git clone https://github.com/ryu-thakahashi/otree-template.git  
otree-template/otree-template.code-workspace
```

仮想環境を有効にする

1. 「Cmd (Ctrl) + Shift + P」を押す
2. 「Select interpreter」と入力し、選択する

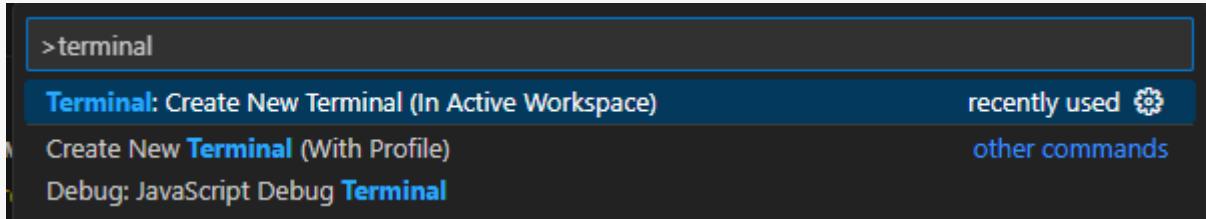


3. .venv という名前の interpreter を選択する



oTree が動くか確認

1. 「Cmd (Ctrl) + Shift + P」を押す
2. 「terminal create」 と入力し、選択する



3. uv sync を入力し、Enter を押す

A screenshot of a terminal window. The command 'uv sync' is typed in and highlighted with a red selection bar. The terminal output shows some configuration information about ssh-agent.

4. otree --version で動くか確認する

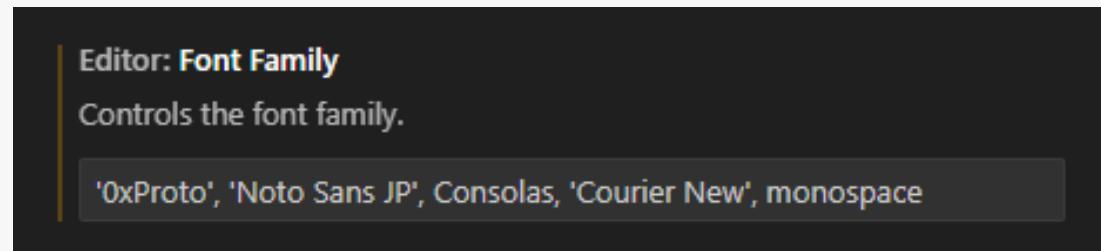
A screenshot of a terminal window. The command 'otree --version' is typed in and highlighted with a red selection bar. The terminal output shows the version '5.11.1'.

VS Code のもろもろの設定

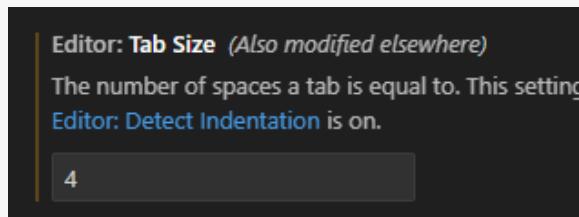
「Cmd (Ctrl) + ,」を押して設定を開く

- フォントを設定する
 - [0xProto](#) をインストール ([参考](#))
 - [Noto Sans JP](#) をインストール
 - フォントを以下のように設定
- 保存時にコードをフォーマットする

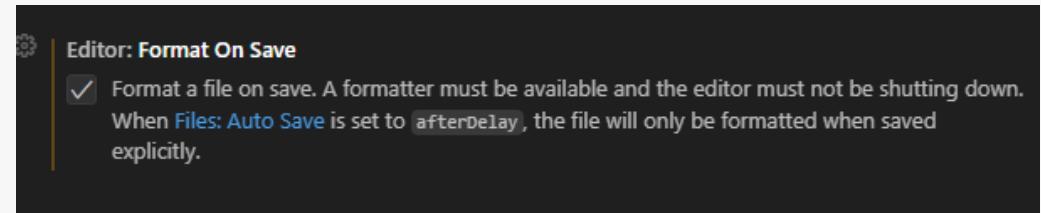
'0xProto', 'Noto Sans JP', Consolas, 'Courier New', monospace



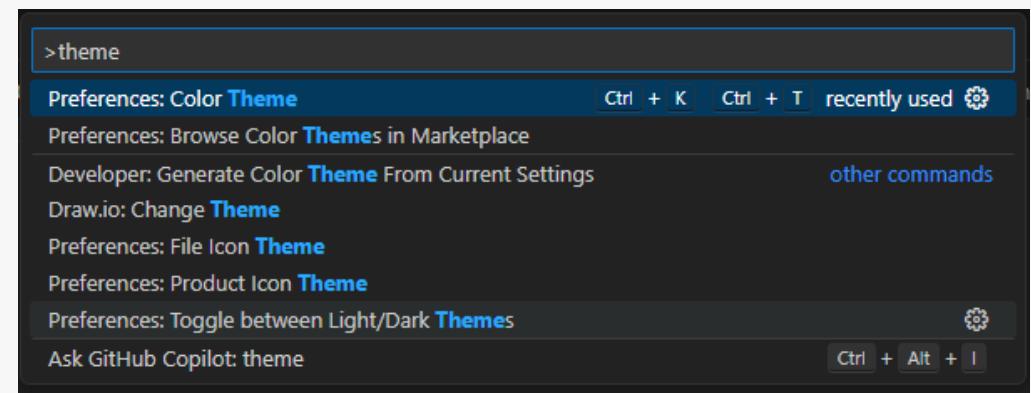
- タブサイズを変える



- Theme を変える



- 「Cmd (Ctrl) + Shift + P」を押す
- 「Color Theme」と入力し、選択する



Render に登録する

<https://render.com/> にアクセスしてアカウントを作成する