Bulletin of the Educational Research and Development, Faculty of Education, Kagoshima University
2021, Vol.30, 135-144

資料

心理学実験実習における遠隔授業
ー jsPsych を用いた遠隔実験の実施について ー

下木戸　隆司［鹿児島大学教育学系（教育心理学）］

Distance learning in psychology experiment training: The implementation using jsPsych

SHIMOKIDO Takashi

キーワード：心理学実験、遠隔授業、jsPsych、Web、JavaScript

はじめに

新型コロナウイルス感染症（COVID-19, coronavirus disease 2019）の影響で，受講生を講義室に集めて実施する従来の対面授業ではなく，情報通信技術を用いて，離れた場所同士で意思疎通や情報発信を行う遠隔授業が注目を集めている。対面授業では，①換気の悪い密閉空間である，②多くの人が手の届く距離に集まる，③近距離で会話や発声を行う，といったウイルス感染防止の点から問題視される「3 つの条件が同時に重なる場」となりやすいためである（文部科学省，2020）。遠隔授業では，Zoom や Microsoft Teams などの Web 会議システムをはじめとして，YouTube や Vimeo などの動画配信システム，Moodle や manaba，Google Classroom などの学習管理システムといった様々なメディア・サービスが利用されている。こうしたサービスを利用することで，受講生の健康・安全に配慮し，ウイルス感染リスクを抑えながら，従来の対面授業と同等かそれ以上の教育効果をあげることが期待されているわけである。

一方，遠隔授業が比較的実施しやすい講義に比べ，実習のように実技・実演を伴うものにおいては，実施が難しいという問題がある。心理学においても，心理学実験をとおしてデータを収集し，その結果を分析し，考察したものをレポートとしてまとめるといった実験実習授業が，学部生に課せられることが多い。[1]　従来ならこうした実験実習は，心理学実験室内で実施されるか，もしくは通常の講義室に実験機材を持ち込んで実施されることが多かったと考えられる。しかしながらいずれの方法も「3 つの条件が重なる」危険性という点では，問題があるといえる。とくに心理学実験室は，実験時の防音，防磁の目的から，遮音・吸音加工や電磁シールドが施された密室となっており，十分な換気が難しいものが少なくない。こまめな換気や，接触感染防止（手洗い，手指や手の触れる場所の消毒），飛沫感染防止（マスクの着用）の対策を徹底化し，なおかつ参加する受講生数を制限するにしても限界があるため，心理学実験実習を遠隔授業で実施する方法を探り，確認して

[1] 鹿児島大学教育学部では，心理学初級実験という名称で心理学領域必修科目として開設されている。他の心理学系の大学・学部でも，名称は異なるかもしれないが，同じような趣旨の実験実習科目が設置されていると思われる。

おくことは重要といえよう。

本稿では，コロナ後の時代における心理学実験実習のあり方として，Web ページを用いた遠隔形式での実験（遠隔実験）について述べる。遠隔実験の重要性は，今後もますます高まっていくことと想定されるが，本稿では教育目的での用途に限定し，あくまで簡単な紹介に留めることにする。より詳しい情報を知りたい方は，関連文献（例えば，Grootswagers, 2020; Hilbig, 2016; 黒木, 2020; Reips & Krantz, 2010）を参照されたい。

Web ページを用いた遠隔実験

遠隔実験を実行する方法として, Web ページに実験制御プログラムを埋め込むものが考えられる。Web ページであれば, Web ブラウザ（Internet Explorer, Microsoft Edge, Google Chrome, Mozilla Firefox, Safari 等）を介して実行可能であり，オペレーティング・システム（OS）や機器の種類に依存しないという利点がある。実験室での個別実験と異なり，遠隔実験では受講生（実験参加者）の所持する機器で実施することが想定され，常に同じものが利用されるとは限らないためである。

それに加えて，プログラムの開発と実行に特別な費用を要しないという利点も大きい。ほとんどの Web ブラウザは無料で利用できるし，Web ページを記述・編集するためのテキスト・エディタと呼ばれるソフトウェアも，多機能・高機能なものが無料で手に入るからである。Windows であればメモ帳が，macOS であればテキストエディットが標準搭載されているので，それらを使用してもよいが，Visual Studio Code（VSCode）や Atom といった HTML，JavaScript に対応した無料のテキスト・エディタを用いた方が便利である。

Web ページ以外の遠隔実験の方法として，プログラムの実行ファイルを配布し，受講生（実験参加者）のパーソナル・コンピュータ（PC）上で実行するというのも考えられる。ただしこの方法は，実行ファイルを作成（コンパイル）した PC と，受講生の PC の動作環境が一致しないと，正しく実行されないという問題がある。実験者が Windows で作成した実行プログラムを，受講生が macOS で実行しようとしても，通常は起動すらできない。[2] 両者が同じ Windows 環境である場合でも，ある PC で作成した実行ファイルを配布し，別の PC で起動を試みた際に，実行時エラー（runtime error）が発生して動作しないこともある。これらの問題を回避するには，受講生の所持する PC で一つ一つ確認しておく必要があるため厄介である。[3]

Web ページを利用して実験制御プログラムを作成するには，Web ページの仕組みについて理解を要する。ゼロからプログラムを作成する場合はもちろん，既存のプログラムを使い回し，再利用する場合でも，最低限度の知識は必要である。以下に Web ページの仕組みについて簡単に述べる。

[2] Wine や VirtualBox など，Windows 環境の仮想化ソフトウェアを利用して，実行ファイルを起動する方法が考えられる。ただしこの場合でも，これらのソフトウェアを予め導入しておく必要があることに加え，Windows 環境下と同等の動作をするかは要確認である。

[3] 多くは，実行ファイルが求める Microsoft Visual C++ Runtime Library や Microsoft .net Framework 等が，PC でインストールされていない（使用可能な状態になっていない）ことが原因である。これらのファイルは，通常は Windows の更新時に自動でインストールされるものの，何らかの理由で正しくインストールされなかった場合は，手動で実行する必要がある。

Web ページの仕組み

Web ページは，HTML（HyperText Markup Language）や CSS（Cascading Style Sheet），JavaScript と呼ばれる言語で構成されることが多く，それぞれ Web ページの構造，装飾，動的処理を担当している。一般的に Web ページは，UTF-8 と呼ばれる文字コードによって記述される。

HTML　HTML タグと呼ばれる制御記号によって Web ページの構造を規定しており，ただ文章を表示するだけでなく，ある部分を見出しとして大きな文字で表示したり，段落として文章を区切り，間に空白を挟んだりして，見やすいものに体裁を整えることができる。それに加え，画像や動画，音声などのデータを埋め込んだり，クリックすると他の Web ページに移動するリンクを張ったりすることも可能である。HTML は CSS や JavaScript の記述を読み込み，それらをまとめるはたらきも担っている。

CSS　Web ページのレイアウトや背景色，文字の字形（フォント），大きさ，色といった，Web ページの見栄えや装飾にかかわる部分を規定する。一般的には，独立した CSS ファイルを作成しておき，それを HTML ファイル内で読み込ませることが多い。そうすることで，Web ページのレイアウト修正を行ったときに，その CSS ファイルを読み込むすべての Web ページに一括して反映されるので，一つ一つ手直ししなくてすむ。つまり Web ページの更新・管理が容易になるのである。このような利点もあり，インターネット上で閲覧できる Web ページの多くは，HTML だけでなく，CSS を利用して記述されている。

JavaScript　Web ページで動きのある表現を可能にする。アニメーションを動かしたり，ボタンをクリックすることでメニュー画面が切り替わったりするだけでなく，Google Maps を利用して Web ページ内に地図を表示し，その地図を動かしたり，拡大縮小させたりすることもできる。Web ページに写真を複数枚用意しておき，マウスの動きと連動して，スライドショーのように写真を切り替えることも，JavaScript を使えば容易である。近年では，JavaScript は動的な Web ページをつくるのみならず，ゲームやソーシャル・ネットワーキング・サービス等のアプリケーション開発にも利用されている。JavaScript を利用して心理学実験の制御プログラムを作成することもでき，それを支援するための jsPsych（de Leeuw，2015）や common.js（水野・松井，2014）といった便利なライブラリ（フレームワーク）も開発されている。

JavaScript は，いわゆるオブジェクト指向言語と呼ばれるプログラム言語である。C/C++などを用いてプログラミングをした経験のある人であれば，条件分岐やループ処理等，C/C++とよく似た部分も多いため，比較的容易に修得が可能と思われる。しかし一方で，クラスという抽象的なひな形を定義してから，その実例（インスタンス）を実体化してオブジェクトを生成するのではなく，既存のオブジェクト（プロトタイプ）を利用することでオブジェクトの生成を行う等の違いもあり，JavaScript 特有の仕様もあるので注意が必要である。

jsPsych

本稿では，JavaScript を使用して心理学実験の制御プログラム作成を支援するフレームワークであ

る，jsPsych（de Leeuw，2015）を取り上げて解説する。jsPsych はタイムライン（timeline）という変数に，教示や刺激呈示・反応取得といったイベントを記述し，実験の流れを時系列的に構成していくところに特徴がある。JavaScript で実験者が一から実験制御プログラムを作成するより，こうしたフレームワークの機能を利用した方が，手間が省けて楽であるし，プログラミングのミスを減らせるという利点もある。とくにプログラミング未経験者や初心者にとっては，最低限習得すべき知識量が減り，学習コストが低減することは大きいといえよう。

jsPsych を利用してプログラムを作成するには，予め PC にインストールされている必要がある。はじめて利用する場合には，まず公式サイト（https://www.jspsych.org）からファイルをダウンロードし，PC 上の任意の場所に展開すればよい。2020 年 8 月時点では，最新版は 6.1.0 である。ファイルが展開されると，css，examples，plugins といったフォルダに加え，jspsych.js，license.txt といったファイルが作成される。この jspsych.js が jsPsych の本体にあたり，このファイルを読み込むことで，jsPsych の機能が利用可能になる

jsPsych によるプログラミングについては，本稿ではあまり深く踏み込まず，簡単な解説に留めることにする。公式サイトにチュートリアルがあるほか，インストール後につくられる examples フォルダに複数のプログラムが例示されている。インターネット上にも解説がいくつも公開されているので，適宜関心のある方は参照されたい。

jsPsych による実験制御プログラムの例

ここでは，学部生向けの心理学実験実習でも取り上げられることの多い，選択反応時間測定の実験を例に，jsPsych の機能とプログラム記述について見ていこう（図 1 参照）。

実験の流れとしては，①まず最初に教示説明が表示され，キーボードで何かキーを押すとそれが消え，②2 秒間のブランク（空白）画面が表示される，③注視点（＋）が中央部に呈示され，400，600，800，1000，1200，1400 ミリ秒のいずれかランダムに選ばれた値の間，表示される。④■，◆，★，▲のいずれか一つがターゲット刺激として中央部に呈示され，f か j のキーを押すことでターゲットが消える。押されたキーの種類とその正誤，そのときの反応時間の情報を取得した後，⑤2 秒のブランクを挟んで，③から⑤の手順を計 120 回（120 試行分）繰り返す。⑥事後説明が 2 秒間表示されると，取得した情報を chice_rt_result.csv というファイル名で保存して，実験が終了する。

この実験では，各ターゲット刺激が呈示される順番はランダム化されており，ターゲットが■，◆であればキーボードの f キーを押し，★，▲であれば j キーを押すように求めている。正しくキーが押されれば，結果ファイルの correct 変数に TRUE の値が保存されるが，もし間違ったキーが押された場合には誤答となり，correct 変数に FALSE の値が保存される。各種の結果のデータはカンマで区切られたテキスト形式（CSV 形式）で保存され，Excel 等のソフトウェアで開くことができる。ファイル名と保存先は，任意に変更することが可能である。ただしファイルの文字コードが UTF-8 となっているため，環境によっては，文字が適切に表示されない（文字化けが生じる）かもしれない。この対処法はいろいろ考えられるが，一つには BOM（byte order mark）を付して保存す

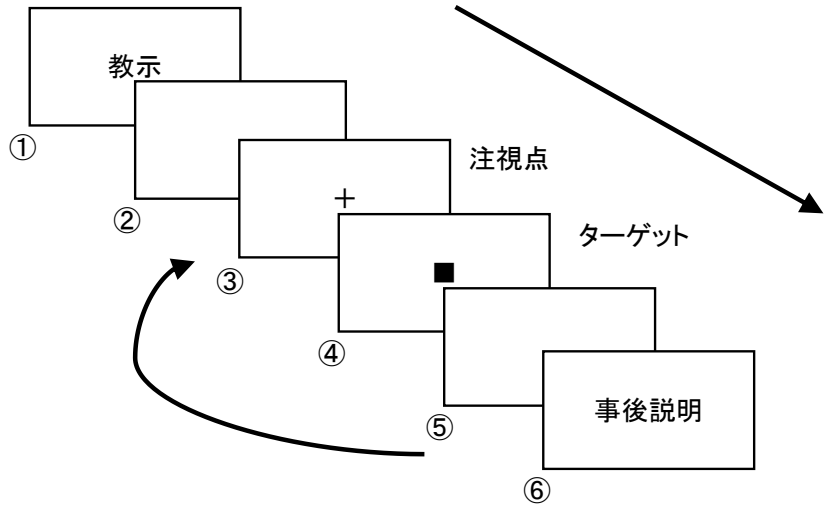

図 1　選択反応時間を測定する実験の流れ

る方法がある。ファイルの先頭部に BOM を付けることで，文字コードが UTF-8 であることが明示されるため，ソフトウェアで開いたときに文字化けが起こるのを回避できる。具体的なやり方として，例えば，Windows に標準搭載されているメモ帳で結果ファイルを開いた後，文字コードのオプションで「UTF-8（BOM 付き）」を選択して保存（名前を付けて保存）を行えばよい。[4]

choice_rt.htm（添付資料参照）は，選択反応時間の測定を行う実験プログラムを収めた Web ページのファイルである。冒頭の<!DOCTYPE html>で HTML5 で記述されていることを宣言し，<html>と</html>で挟まれた部分が HTML 文章であることを明示している。

<head>と</head>で挟まれた部分が，HTML 文章のヘッダ領域と呼ばれるところで，文章のタイトルであったり，作者名であったり，使用する文字コードやスタイルシートを指定したりと，HTML 文章の特徴や設定に関する情報（メタデータ）を記述している。ヘッダ領域に記載した情報の多くは，一部の例外を除いて，Web ブラウザには表示されない。例では，<meta charset>で HTML 文章の文字コードを UTF-8 と指定しており，Web ページで日本語の文字を正しく表示するための設定である。続いて<script>と</script>で挟まれた部分で，JavaScript プログラム（スクリプト）の呼び出しを行っており，src 属性でそのファイルのアドレスを指定している。具体的には，jspsych.js と jspsych-html-keyboard-response.js という，JavaScript のファイルを読み込んでいる。<link>と</link>で挟まれた部分では，Web ページのスタイルシート（例では jspsych.css）を外部ファイルとして読み込み，jsPsych で表示される文字のフォントや大きさ，表示位置等の設定を記述している。

続いて<body>と</body>で挟まれた部分が，HTML 文章の内容に関する領域で，Web ブラウザに表示される文章の本体にあたる。今回の例では何も記載されていないが，一般的な Web ページでは，閲覧できる文章の文面がこの body 領域に記載される。

body 領域の後に，再度<script>と</script>で挟まれた部分があるが，例ではここに jsPsych のプログラムを直接記述している。jsPsych を利用してプログラムを作成する場合，この部分を自分の都合

[4] Excel を利用するのであれば，「外部データの取り込み」からテキスト・データとして，「Unicode（UTF-8）」を選んでインポートする方法もある。その際，区切り文字としてカンマにチェック（✔）が入っているかを確認するとよい。インポートが完了したら，保存するのを忘れないこと。

にあうように記述すればよく，<script></script>以外の部分は，とくに手を加えなくても，そのまま使い回せることが多いと考えられる。ただしプログラムの記述が長くなる場合は，独立したファイルとして保存し，外部ファイルとして読み込ませた方がよいかもしれない。その方が，プログラムの可読性や保守性の点から好ましいからである。実験制御プログラムは，実施する課題の性質に応じ，既存のプログラムを改変して作成する（使い回す）ことが多いと思われる。その際に少しでも読みやすい，わかりやすい記述になっていることは，実用上重要である。

例にある選択反応時間の実験制御プログラムでは，①教示文→②ブランク（空白画面）→③注視点呈示→④ターゲット刺激呈示→⑤ブランク→⑥事後説明という流れ（タイムライン）で進行する（図 1 参照)。③から⑤については，③注視点呈示→④ターゲット刺激呈示→⑤ブランクを計 120 回（120 試行分）繰り返した後で，⑥事後説明に移行するように記述している。プログラム上は，instructions，fixation，showStimulus，debrief というオブジェクトをそれぞれ生成し，それらをタイムライン変数に代入し，jsPsych.init 関数を呼び出すことで実験が開始されることになる。

type（対応するプラグイン名)，stimulus（表示する HTML 文字列)，post_trial_gap（次の試行に移行するまでの時間)，choices（反応として求めるキーの種類)，trial_duration（当該試行が持続する時間)，on_finish（試行終了時に実行する処理）は，オブジェクトのプロパティ名（キー）である。これらは jspsych.js，jspsych-html-keyboard-response.js のなかで定義されているものであり，これらに適切な値を代入してタイムライン変数に渡すことで，実験制御を行っている。jsPsych では，課題や処理の性質に応じて使い分ける沢山のプラグインが用意されており，実験の目的と合致したものを読み込む必要がある。例では，HTML 文字列を表示し，キーボードを用いて反応を取得する，jspsych-html-keyboard-response というプラグインを利用している。jsPsych の各プラグインの性質と，それぞれどのようなプロパティが用意されているかは，公式サイトにわかりやすくまとめられているので参照されたい。

timeline には，実験を構成するために作成されたオブジェクトが代入され，最終的に jsPsych.init 関数の引数として jsPsych に渡されることになる。timeline_variables は，その際に参照される配列を指定しており，この場合では stimList として定義した配列から，ターゲット刺激を読み込んでいる。その際，sample 中で fixed-repetitions を指定し，size で示した回数分 stimList を読み込み，刺激呈示の順番をランダム化している。例では，ターゲット刺激は■，◆，★，▲の 4 種類あり，それらを 30 回ずつそれぞれ読み込んで，順番をランダムに並び替えて呈示するようにしている。なお今回は用いていないが，repetitions と randomize_order を指定する方法もある。stimList を指定した回数分読み込んでランダム化する点は同じだが，repetitions と randomize_order を指定した場合は，stimList から 1 回分読み込むごとにランダム化されるので，基本的に同じ刺激（例えば■）が連続して呈示されないという違いがある。目的に応じて使い分けるとよいだろう。

以上，選択反応時間を測定する実験を例に，jsPsych の記述についてみてきたが，他にも様々な実験課題が作成可能である。本稿では取り上げなかったものの，視覚探索課題のように複数の画像を円状に散りばめて配置したり，音声や動画のデータを呈示したり，リッカート尺度による質問紙実

験を構成したりすることも，jsPsych では比較的容易に実施できる。興味のある方は試されたい。

実施時の注意点

jsPsych は JavaScript で動作するため，Web ブラウザが利用可能であれば，ファイルを開くことで Windows や macOS，Linux の PC だけでなく，Android や iOS のスマートフォン，タブレット端末でも動作する（マルチプラットフォーム）。しかしスマートフォンやタブレットでは，画面が小さく解像度が低いことによって，レイアウトが崩れて表示される可能性や，キー入力に対応していない等の問題がある。そのため，実際に遠隔実験を行う場合は，ある程度大きな画面が使え，キーボードを備えた PC に限定した方が無難だろう。

それに加えて JavaScript は Web ブラウザによって異なる環境（エンジン）で動作するので，遠隔実験で使用する Web ブラウザを指定した方がよい。とくに Internet Explorer は新しい規格の JavaScript に対応していないため, jsPsych が正常に動作しない危険性があり, 避けるべきであろう。Microsoft Edge，Google Chrome，Mozilla Firefox，Safari は新しい規格の JavaScript に対応しているので，これらの最新版を使用するのが望ましいが，予め jsPsych プログラムが正しく動作することが確認できたものを，受講生（実験参加者）に利用するよう促すべきである。

今回は簡潔を期すために記述を省いたが，諸々の不具合が生じる危険性を防止するため，jsPsych プログラムは，フルスクリーン（全画面表示）環境で実行した方がよいだろう。jsPsych では，jspsych-fullscreen プラグインを読み込み，fullscreen_mode プロパティに TRUE 値を代入することで実現できる。逆にフルスクリーン環境を解除するには，fullscreen_mode プロパティに FALSE 値を代入すればよい。またプログラム実行時には，ミリ秒単位の高い時間精度を確保・維持するために，なるべく Web ブラウザ以外の余計なソフトウェアを終了させておくべきである。Windows や macOS, Linux などのマルチタスク OS では, 複数の処理を短時間で切り替えながら同時処理を実現しているため，同時に行う処理が多ければ多いほど，プログラムの実行に遅延が生じる可能性が高くなるからである。

他にも遠隔実験では，事前に想定していなかった様々な不具合が生じる可能性があり，それらへの対応も必要になる。遠隔実験では，実験者がその場に居合わせないため，どのような状況下で問題が発生したのかが，よくわからない場合が多い。受講生（実験参加者）の報告が要領を得ないこともあるため，エラー画面を写真に撮り，それが生じた詳しい経緯を記録して送信するように求めるなど，不具合が発生したときにどうするかを，事前に詳しく説明しておくことは重要であろう。

さいごに

コロナ後の時代において，従来の方法で心理学実験実習を実施することが難しくなっていることを踏まえると，遠隔実験の重要性は今後さらに高まっていくものと考えられる。その方法として，本稿では jsPsych によって遠隔実験を実施するものについて述べた。遠隔実験を行う手法としては，他にも無料で利用できる PsychoPy（Peirce et al., 2019; 十河，2019）や lab.js（Henninger, Shevchenko,

Mertens, Kieslich, & Hilbig, 2019），OpenSesame（Mathôt, Schreij, & Theeuwes, 2012）等を使うというもの，有料ではあるが Pavlovia や Inquisit，Gorilla 等のサービスを利用するものなど，選択肢は様々なものがある。各々の関心や利用環境に応じて，適切なものを選択すればよいだろう。

心理学実験実習を実施する上で，Moodle や manaba，Google Classroom などの学習管理システムを利用するのが便利である。実験テーマに関して，解説資料の掲示，実験プログラムの導入，素データやレポートの提出，受講生からの質問，さらには活動への取り組み状況の把握といった一連の機能を，一つの Web プラットフォーム（Web 上で展開される共通基盤）で実現できるからである。例えば，反応時間の測定に関する資料を受講生に読ませ，課題と実験手順の説明動画を視聴させた後，jsPsych プログラムによる実験を実施する。そこで得られた結果のデータを分析させ，必要に応じて素データや分析途中のファイルの提出を求め，受講生からの質問があれば，掲示板や電子メールで受け付ける。結果から受講生が考察し，作成したレポートを Web フォームで提出させる。これらの活動を一つの学習管理システム下で実行できるのである。Web 上で学習管理システムにログインして，心理学実験実習のコース（サイト）にアクセスすれば，心理学実験実習で行われるすべての活動を済ませることができるので，受講生にとっての利便性は大きいといえよう。

今後，遠隔授業に関するノウハウが蓄積されるにつれ，心理学実験実習を遠隔授業で実施するケースは増えていくことが予想される。従来の対面授業以上に，高い教育効果を見込める遠隔授業の方法も開発されるかもしれない。そのためには多くの実践的試みがなされ，それらが広く共有されることが重要となる。本稿がその一助になれば幸いである。

引用文献

de Leeuw, J. R. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a web browser. *Behavior Research Methods*, *47*, 1–12. https://doi.org/10.3758/s13428-014-0458-y

Grootswagers, T. (2020). A primer on running human behavioural experiments online. *Behavior Research Methods, 52*, 2283–2286. https://doi.org/10.3758/s13428-020-01395-3

Henninger, F., Shevchenko, Y., Mertens, U., Kieslich, P. J., & Hilbig, B. E. (2019). lab.js: A free, open, online experiment builder. Zenodo. https://doi.org/10.5281/zenodo.2775942

Hilbig, B. E. (2016). Reaction time effects in lab- versus Web-based research: Experimental evidence. *Behavior Research Methods*, *48*, 1718–1724. https://doi.org/10.3758/s13428-015-0678-9

黒木 大一朗 (2020). ウェブ実験の長所と短所，およびプログラム作成に必要となる知識 基礎心理学研究, *38,* 250–257. https://doi.org/10.14947/psychono.38.37

Mathôt, S., Schreij, D., & Theeuwes, J. (2012). OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, *44*, 314–324. https://doi.org/10.3758/s13428-011-0168-7

水野 りか・松井 孝雄 (2014). ブラウザでできる基礎・認知心理学実験演習――JavaScript で書く実験プログラム―― ナカニシヤ出版

文部科学省 (2020). 令和２年度における大学等の授業の開始等について（通知）
https://www.mext.go.jp/content/20200527-mxt_kouhou01-000004520_3.pdf

Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., & Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, *51*, 195–203. https://doi.org/10.3758/s13428-018-01193-y

Reips, U. -D., & Krantz, J. H. (2010). Conducting true experiments on the Web. In S. D. Gosling & J. A. Johnson (Eds.), *Advanced methods for conducting online behavioral research* (pp. 193–216). American Psychological Association. https://doi.org/10.1037/12076-013

十河 宏行 (2019). PsychoPy の新機能――Python3 サポート・ブラウザベースの実験・実験の共有―― 基礎心理学研究, *38*, 154–160. https://doi.org/10.14947/psychono.38.24

添付資料

choice_rt.htm のプログラム・コード

HTML
CSS
JavaScript

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <script src="./jspsych.js"></script>
  <script src="./plugins/jspsych-html-keyboard-response.js"></script>
  <link rel="stylesheet" href="./css/jspsych.css"></link>
</head>
<body>
</body>
```

```
<script>
//注視点呈示時間の格納
let fduration;

//教示
const instructions = {
  type: "html-keyboard-response",
  stimulus: '<div style = "font-size: 20px;"><p>注視点「＋」が現れたらそれに視点をあわせてください。</p>'+
  '<p>次に表示される文字の形に応じて，キーボードのキーを押してください。</p>'+
  '<p>■，◆＝Ｆキー　　★，▲＝Ｊキー</p>'+
  '<p>キーはできるだけはやく，正確に押してください。</p>'+
  '<p><br>準備ができたら，何かキーを押すと実験がはじまります。</p></div>',
  post_trial_gap: 2000
};

//注視点　呈示時間は 400, 600, 800, 1000, 1200, 1400 ms のいずれかランダム
const fixation = {
  type: "html-keyboard-response",
  stimulus: '<div style = "font-size: 40px">＋</div>',
  choices: jsPsych.NO_KEYS,
  trial_duration: function() {
    fduration = jsPsych.randomization.sampleWithoutReplacement([400, 600, 800, 1000, 1200, 1400], 1)[0];
    return fduration;
  },
  on_finish: function(data) {
    data.fduration = fduration;
  }
};
```

```
//刺激リスト　各試行では ★, ▲, ■, ◆ のいずれかが表示される
const stimList = [
  {stimulus: '<p style = "font-size: 40px;">★</p>', data: {corect_response: 'j'}},
  {stimulus: '<p style = "font-size: 40px;">▲</p>', data: {corect_response: 'j'}},
  {stimulus: '<p style = "font-size: 40px;">■</p>', data: {corect_response: 'f'}},
  {stimulus: '<p style = "font-size: 40px;">◆</p>', data: {corect_response: 'f'}},
];

//刺激呈示・反応取得
const showStimulus = {
  type: "html-keyboard-response",
  stimulus: jsPsych.timelineVariable('stimulus'),
  choices: ['f', 'j'],
  data: jsPsych.timelineVariable('data'),
  post_trial_gap: 2000,
  on_finish: function(data) {
    data.response = jsPsych.pluginAPI.convertKeyCodeToKeyCharacter(data.key_press);
    data.correct = data.response === data.corect_response;
  }
};


//試行を構成　注視点 → 刺激呈示・反応取得
const main = {
  timeline: [fixation, showStimulus],
  timeline_variables: stimList,
  sample: {
    type: 'fixed-repetitions',
    size: 30
  }
};

//デブリーフィング
const debrief = {
  type: 'html-keyboard-response',
  stimulus: '<div style = "font-size: 20px;"><p>これで実験終了です。お疲れ様でした。</p></div>',
  trial_duration: 2000,
  choices: jsPsych.NO_KEYS
};

//実験全体の構成
const timeline = [];
timeline.push(instructions);
timeline.push(main)
timeline.push(debrief);

//実験開始
jsPsych.init( {
  timeline: timeline,
  on_finish: function() {
    jsPsych.data.get().localSave('csv','chice_rt_result.csv');
  }
} );

</script>

</html>
```