

Intro to Psychtoolbox

This wiki page will hopefully be a useful guide to the basics of Psychtoolbox (PTB). For specific questions, please take a look at our [Psychtoolbox Q&A](#). If your question isn't in there, feel free to add it!

If you are coming in with little or no MATLAB experience, it may be useful to get an overview of how the language works first, as this guide will assume basic MATLAB understanding. For a quick review of MATLAB and an introduction to PTB, it may be useful to look over the first few weeks of slides provided by Dr. Jonas Kaplan from USC: [Intro to Psychtoolbox in Matlab](#).

Below we outline a few basics for working with PTB scripts. Then, we recommend you use the example scripts and demos noted in second section to see contextual examples of more specific functions.

1. BASICS

Screen function

The Screen command is the basis of PTB scripts, as it controls the way information is displayed to the participant. Because of its fundamental nature, we decided to begin with a brief overview of its functionality so that you can get started on the framework of an experiment. Screen can have many different functions, and many functions work with Screen to create a working program.

Before you can present things to the subject, you must first initialize a "window" that will contain the visual material you want to present. You create a new window with the Screen('OpenWindow') function.

A simple (but not perfect) way to think about this window is as a double-sided dry erase board. Each side of the board can be referred to as a buffer. At any given time, you present one side to the subject. This is the front buffer. While the subject views the material on the front buffer, you prepare the next screen you want them to see on the back buffer. For example, you can prepare text using the DrawFormattedText() function or images using the Screen('DrawTexture') function. When you are ready to present the next screen, use the Screen('Flip') function to flip the buffers: the back buffer switches to the front and you get a new back buffer to prepare for the next screen and so on.

To end your program, you must close the window and anything drawn on it so that you can access your monitor again. To do this, you simply use Screen('CloseAll').

We provide a brief example with a detailed description of each function in the [Screen Function](#) wiki page.

try-catch structure

We recommend using the try-catch structure for your PTB scripts to make them more robust to errors. Section 2 contains examples of this.

The idea here is to embed the bulk of your script (the real "functional" portion) in the try block. MATLAB will attempt to execute everything in the try block per usual. If no errors occur, the catch block will be skipped entirely. If any errors do occur within the try block, MATLAB will immediately jump to the catch block and proceed through whatever commands you have included there to deal with the error situation. This way, an error is much less likely to crash your task in an especially unceremonious way (just freezing, leaving keyboard output disabled, etc.) or go unnoticed until after it is done. Rather, the catch block can be used to properly close files and restore settings as well as report info on the error.

2. LEARN THROUGH EXAMPLES

One of the best ways to get started is to look through one or more existing PTB experiments and learn how they work. You can then use relevant parts from these as a model for your own PTB script. We are linking two previous, well-commented experiments from the lab below. We believe it's easier to learn a lot of these functions in context. So, instead of going through all the common functions one by one, we commented these scripts to try to explain what the functions and their components are doing.

Experiment 1 - [script for training session, main task script](#)

This experiment compares different emotion regulation techniques in response to aversive pictures. It involves cuing the subject for a viewing strategy, presenting an image, and presenting several screens asking the subject to make a keyboard response. The full set of files and scripts associated with this experiment can be found in [ccn-keoki.win.duke.edu/experiments2/John/EmoReg1](#)

These scripts cover:

- using arguments (input you give your script when you call it, such as the subject #)
- constructing and using pseudo-randomized stimuli and condition orders
- defining and presenting screens with text or images
- working with keyboard responses
- sending signals to BIOPAC
- writing keyboard responses and trial information to an output data file
- playing audio files (for startle)
- triggering screen changes with subject responses vs. defined durations

Experiment 2 - [script for running Monte Carlo pseudo-randomization](#)

This experiment is a three day memory reconsolidation study, including encoding, reactivation with reconsolidation, and retrieval. The goal is to see whether participants can use emotion regulation as a form of reconsolidation. The task itself appears in the form of paired associate learning, where participants are presented with pairs of images, reminded of one, and tested on their memory of the associated picture. This paradigm requires displaying images, cuing regulation strategies, reading in user button presses, and keeping track of old and new images for a memory test. The full set of scripts and stimuli can be found in [ccn-keoki.win.duke.edu/experiments2/Natasha/EmoRegCon](#)

In addition to most of the items covered in Experiment 1, this experiment covers:

- using mouse clicks to advance screens (practice/training files)
- displaying multiple images on a single screen (*Session1.m*)

- online pseudo-randomization in a systematic manner (*organizeStims.m*) or using a Monte Carlo simulation (*randomStimsMC.m*, linked to above)
- storage and usage of stimuli across multiple scripts (all Session files)
- batch scripting (*stimCreationBatch.m*).

PTB Demo Materials

There are also a few simple demos that come with the PTB software download itself. On Windows, they are stored in *[main disk drive]/toolbox/Psychtoolbox/PsychDemos/PsychExampleExperiments*, and on Mac they are in *Applications/Psychtoolbox/PsychDemos/PsychExampleExperiments*. In fact, the PsychDemos folder has a lot of useful starting information, including demos of how to include video and audio, although sometimes they call functions that are a little dated.