How to include EyeLink commands into PsychoPy program

* copy the following files/folders into the experiment folder
  + EyeLinkCoreGraphicsPsychoPy.py
  + EyelinkHelperFunctions.py
  + EyelinkSetup.py
  + **sounds** folder
  + **videos** folder
* adding components to your PsychoPy experiment
  + el\_Setup
    - *setupScript*, a component loads eye tracking related packages and functions and set a short name for your study
    - *parameters* (optional), a component loads experiment **SPECIFIC** function or parameters, which need to be set up differently according to experiment
  + welcomeVideo (optional)
    - a video component playing cartoon animation to get participants' attention
  + el\_Calibration
    - a component to manage calibration and validation processes
  + el\_DriftCorrection (optional)
    - a component to activate drift correction process. Ideally, drift correction should be done for each trial.
  + trial
    - this component is the main stimuli presentation Routine
    - *el\_ControlScript*
      * starts eye tracker recording in Begin Routine tab;
      * ends eye tracker recording and save trial variables to eye tracking data in End Routine tab
    - *el\_SendMarkersScript*
      * send markers to indicate experiment events (e.g., onset of a stimulus). Markers should be sent in Each Frame tab. The marker should be sent once when the status of a component (e.g., image) becomes STARTED.
    - *el\_LandmarkScript*
      * creates landmarks (e.g., the location of a image) on Host PC and in eye tracking data
* for Gaze Contingent design
  + add scripts to *Each Frame* tab to read eye movement data every frame. This requires additional code to tell the program how to interact with eye movement data.
    - There are two ways of reading eye movement data in real-time: Gaze sample data (getNewestSample) and Eye Movement event data (getNextData & getFloatData).
* sending Commands
  + el\_tracker.sendCommand: to create visual elements associated with the recording, such as a highlighted area to indicate the location of stimuli. This is usually helpful for data analysis.
  + el\_tracker.sendMessage: to mark events in the recording, such as onset of a trial, stimuli, etc.
* Check **pylink api userguide** for more details: [Getting Started with Python and PyLink (sr-research.com)](https://www.sr-research.com/support/thread-48.html)