

Setting up the software used in this course

This is a guide to setting up the software we will be using in this course. Please follow it in sequence!

The steps applicable to most users are:

1. Download the course materials zip file from Moodle. If files need to change in the future, you may need to do some manual back-and-forth with versions. Please keep an eye on the announcements forum for any update of material.
2. Install **PsychoPy standalone**
3. Open the PsychoPy App. Open the file `test_psychopy_working.py`, located in the `experiments/week01_tools_setup` directory. Click the green “run” arrow. If a screen opens and displays “seems to be working!”, then your PsychoPy installation is (probably mostly) working.
4. Install a Python scientific computing environment like **the Anaconda Distribution**. If you already have one (e.g. what you used in Grundlagen) that will probably be fine.

More detailed descriptions are below. Linux or Chromebook users may need to take a different approach.

You’re done! You should have everything that you need for the course now.

Directory structure on your computer

The unzipped files on your computer should now have the following directory structure (in this example I assume you put them in a directory called `wahrnehmen_seminar_materials`):

(Note that some of the files below may be missing at the start of the semester, since they won’t all be ready).

```
./wahrnehmen_seminar_materials
  LICENSE
  README.md
  data
  experiments
    week01_tools_setup
      test_psychopy_working.py
    week03_know_your_stimulus
      create_monitor_profile.py
      determine_fullscreen_pixels.py
      simple_luminance_match.py
    helper_functions.py
    kanizsa_experiment.py
    kanizsa_params_spatial.yml
    kanizsa_params_temporal.yml
    test_experiments.py
  notebooks
    demo_data
      demo_2ifc_data.csv
      demo_yes_no_data.csv
    helpers
      notebook_helper_functions.py
      prepare_demo_data.py
      psychometric_function.py
      sdt_helper_functions.py
  img
    accuracy_bias.png
    criterion_and_bias_2.png
    criterion_and_bias.png
    morpheus.jpg
    yes_no_2afc.png
  notebook_tests
    week_08_tests.py
```

```
week_10_tests.py
week_12_tests.py
week_13_tests.py
week_03_know_your_stimulus.ipynb
week_07_python_refresher_pandas_basics.ipynb
week_08_experiment_analysis_1.ipynb
week_09_experiment_analysis_2.ipynb
week_10_computing_SDT_measures.ipynb
week_11_roc_curve.ipynb
week_12_optimal_bias_accuracy.ipynb
week_13_forced_choice_performance.ipynb
screenshots
kanizsa_experiment_Illusory_fat_-2.5.png
kanizsa_experiment_Illusory_thin_2.5.png
kanizsa_experiment_L0_control_fat_-2.5.png
kanizsa_experiment_L0_control_thin_2.5.png
tools_setup.md
tools_setup.pdf
```

Note how the directory contains a number of subdirectories, some of which contain other files and some are currently empty (data).

Command line users

Users who are not able to install PsychoPy standalone or the Anaconda GUI interface, or prefer not to use those interfaces, will need to install and use them via a command line interface. For example, PsychoPy standalone and the Anaconda GUI interface may not be available for Linux systems.

To install PsychoPy, see the [PsychoPy download instructions](#), and options like `pip` install or `conda` environments.

To install Python environments for scientific computing needed for the notebooks, follow any of the many available guides for setting up such an environment.

Below we provide additional information / guides for using a command line interface.

Optional: Learn how to use your computer like a developer :-)

To make use of the command prompt (also called a “terminal”, “shell” or “command line”) on your computer implies knowing about the file system. This is how many software developers often interact with a computer, and it allows you to do many things that are not supported by graphical interfaces. This may feel scary at first, but learning to do this will be really useful for your future cognitive science studies. Therefore, I encourage you to take some time outside of the seminar to get familiar with the concepts below.

File systems If you are not sure how to use the file system on your machine (or what a file system is), please [read this article](#), then [see here](#), or [here](#) and generally using your favourite internet search engine to learn about how the file system of your operating system works. Keep going with the further setup steps below during the seminar, but please spend some extra time to understand this really important concept.

Command prompt We will be using the command line quite a bit in this course. If you are not sure how to use the command line to navigate around the file system and run commands, please see [here](#) for Mac or Linux users and [here](#) for Windows users.

For those of you not familiar with using a command line, here are some handy commands you might make use of below:

See your current directory: Mac/Linux: `pwd` (“print working directory”); Windows: `cd` or `chdir`.

Show the contents of the current directory: Mac/Linux: `ls` and hit return (`ls -l` to see output as a list); Windows `dir` and hit return to see a print out of all the files and folders in your current directory.

Enter a subdirectory:

- If you're in the directory `/Users/me/Documents` and you want to go into a directory `materials` that's located in `Documents`, type `cd materials` and hit return.
- If you wanted to go into a subdirectory of `materials` called `tests`, type `cd materials/tests`.
- You can press the `Tab` key on your keyboard to “complete” the path – type `cd mat` and hit `Tab` (will complete to `materials`).

Go up one directory: If you're currently located in `/Users/me/Documents/materials` and you want to go back to `Documents`, you can type `cd ..` and hit return.

Running PsychoPy experiments from the command line

You can run the experiment scripts from the course with the command line. To see interface options, use `python <script_name.py> --help`.

Notes:

- when using PsychoPy from the command line, you almost certainly want to disable the PsychoPy GUI by passing the `--no_gui` flag to the script.
- you can quit the program by pressing either `'q'` or `escape`.
- if PsychoPy ever crashes and you can't see your screen, you should be able to kill it in the command prompt by (1) switch applications to see or activate the command prompt window and (2) press `ctrl-c` a few times (Windows users – may be different).

Other tools

You may consider installing and working with a code editing environment like **VS Code** (free) or **PyCharm**.

VS Code allows you to work with Python code (along with lots of other languages) and with Jupyter Notebooks as well. I am using VS Code for almost everything I do now, and I think it's an excellent tool. One advantage of this is that once you're familiar with how it generally works, you can use it for not only Python development but also languages used in your other courses (e.g. Java or JavaScript).

VS Code is developed by Microsoft, and contains trackers that report user data back to Microsoft to allow them to guide development and improve the product. It also contains the excellent Python language support extension **Pylance**.

If you'd prefer not to send your user data back to Microsoft (to e.g. improve user experience, add your data to a megacorporation's understanding of the world, etc), you can use the free alternative **VS Codium**, which is the same source code but not compiled by Microsoft. Unfortunately it does not seem to contain Pylance, so the Python language support is not as good.

PyCharm is another good tool for python development. Only the Professional edition of PyCharm allows you to edit Jupyter notebooks directly, but as a student you can get it free from the [education discount page](#).

If you choose to use one of these tools, there are plenty of getting started guides available online.

Note that if you want to use one of these IDEs to run the command line programs and notebooks we will be using for this course and you are using an environment (e.g. a **conda** environment), you will need to make sure that your IDE is working from within the correct **conda** environment for the course. Otherwise you will receive errors when you try to run things (usually about “can't find package X”). You can find guides online to set up these IDEs to work with virtual environments. For VS Code, this is usually as simple as selecting the right Python environment in the blue sidebar the first time you open a `.py` file in that project directory.

Report writing tools

You may prepare your report using whatever document preparation system you like. The final report will be submitted as a PDF. You can use Word, LibreOffice or similar.

The standard way to write scientific documents in many technical fields is LaTeX. In case you'd like to use LaTeX, I provide a LaTeX template to get started in the project materials. To use it, you need a LaTeX distribution, which lets you compile the `.tex` file into a PDF. You can do that by either installing a distribution on your system (several gigabytes), or working on the TU Darmstadt's ShareLatex web-based instance.

Note that the TU Darmstadt library offers regular training sessions on how to use tools like LaTeX, Zotero, etc.

I detail how to do these things below. **Note again, that both of these are optional.**

TU Darmstadt ShareLatex (optional)

You can work on the document online using a web-based collaborative LaTeX editor "Sharelatex", of which TU Darmstadt hosts an instance. You can find the info page [here](#)

1. Login [here](#)
2. Click "new project", and select "from zip file".
3. Select the files `report_template.tex` and `bibliography.bib` in the `report` subdirectory of the materials that I gave you, and turn them into a `.zip` file. You can likely use your file manager to do this (on Mac: right click then "Compress").
4. Drag these files into the web browser to upload to ShareLatex.

Install on your system (optional)

[Windows or Ubuntu](#)

[Mac](#)

Once installed, check that you can compile the lab report template `report_template.tex`. Requires the `apa6` document class to be installed, which it should be with one of the bulk LaTeX installations linked above. For more see [docs on this class](#).

You can compile the report from the command line, or from a GUI like [TexMaker](#)