# TUTORIAL 1:
# BRAIN IMAGING DATA STRUCTURE

BY: A. GRIGIS

❓ Overview

**Tutorial outline**

- What is BIDS?
- How a BIDS dataset can be made?
- How a BIDS dataset can be query?

**Objectives**

- Introduction to BIDS and its ecosystem of tools.

**30 minutes**

⚙ Prerequisites

For the best learning experience, it is important to have a basic understanding of the following topics:

- Unix operating system, command line, and shell.
- Python.

If not, we recommend that you follow Software Carpentry's lessons on the Unix shell and Python programming. Although not essential it is also helpful to have an understanding of:

- How brain imaging data are exported from scanners.

Any question: antoine.grigis@cea.fr - Generated: February 8, 2023

## Getting familiar with BIDS

The dataset that will be used in this tutorial is a fake one (empty files). It is a multimodal neuroimaging dataset that first needs to be downloaded from here and unzipped in you working directory. The *tutorials-main/BIDS* folder contains the **dsOO1** project. It follows the BIDS standard and provides raw structural MRI and functional data recorded for 16 participants.
You can find below an overview of a BIDS-compliant dataset content along with a short description of each file.

```
/
├── dataset_description.json ........ Describes briefly the dataset in JSON format
├── participants.tsv ........ Table file that stores demographical information of the participants
├── participants.json ........ Describes the columns of the participants.tsv table file
├── CHANGES ........ Describes changes between dataset versions
├── README ........ Describes the dataset in more details in Markdown format
├── sub-01
│   ├── anat
│   │   ├── sub-01_T1w.nii.gz ........ Raw T1w MRI stored in Nifti format
│   │   └── sub-01_T1w.json ........ Describes T1w MRI acquisition parameters
│   └── func
│       ├── sub-01_task-rest_bold.nii.gz ........ Raw bold MRI stored in Nifti format
│       ├── sub-01_task-rest_bold.json ........ Describes bold MRI acquisition parameters
│       ├── sub-01_task-faces_bold.nii.gz ........ Raw bold MRI stored in Nifti format
│       ├── sub-01_task-faces_bold.json ........ Describes bold MRI acquisition parameters
│       └── sub-01_taskfaces_events.tsv ........ Table file that describes each event
├── sub-02
├── ...
└── derivatives ........ Stores all the data generated using softwares
```

You can always check the BIDS Specification for a complete description of the standard.

### Exercise 1: create a BIDS-compliant dataset

In this exercise, you will work through creating a BIDS dataset from a non-BIDS compliant dataset.

1 Go to the directory **ds001-corrupted** that stores a modified version of the **ds001** dataset created for the purpose of this exercise.
2 Using the BIDS material provided above, rework the dataset to conform the BIDS standard (hints: use the *ls* and *tree* commands).
3 As you work through this exercise, you can use the BIDS validator to check your progress (must use Google Chrome or Firefox).

### Exercise 2: interact with BIDS dataset in Python

In this exercise, you will learn how to interact with a BIDS-compliant dataset in Python using the *pybids* library.

1 Open a terminal.
2 Install pybids: *python -m pip install pybids.*
3 Launch *ipython.*
4 Import the *BIDSLayout* class from the *pybids* library (hint: check the *pybids* tutorial).
5 Initialize a *BIDSLayout* object with the path of BIDS-compliant dataset.
6 Use the *get()* method on the created object to get the path of the T1w image of *sub-01.*
7 Similarly, use the *get()* method on the created object to get the path of the functional MRI images of *sub-01.* Optionally, provide the argument in a Python dictionary (hint: use the **** operator to pass a number of arguments in the form of a key/value dictionary to a function).
8 List the available functional MRI tasks (hint: use *get_tasks()*).
9 List the events files associated to the functional images retrieved in 7.