Structure of the course material for the IWR Dune course

```
course
├── dune
│   ├── dune-common
│   ├── dune-grid
│   ├── ...
│   └── dune-pdelab-tutorials
│       ├── tutorial00
│       │   ├── src
│       │   ├── doc
│       │   └── exercise
│       │       ├── task
│       │       ├── doc
│       │       └── solution
│       ├── tutorial01
│       │   └── ...
│       └── ...
├── release-build
│   ├── dune-common
│   ├── dune-grid
│   ├── ...
│   └── dune-pdelab-tutorials
│       ├── src_dir
│       ├── tutorial00
│       │   ├── src_dir
│       │   ├── src
│       │   │   └── src_dir
│       │   ├── doc
│       │   │   └── src_dir
│       │   └── exercise
│       │       ├── src_dir
│       │       ├── task
│       │       │   └── src_dir
│       │       ├── doc
│       │       │   └── src_dir
│       │       └── solution
│       │           └── src_dir
│       ├── tutorial01
│       │   └── ...
│       └── ...
├── install.sh
├── external.sh
├── update.sh
└── release.opts
```

The course material for the dune-pdelab course at IWR is distributed in form of a virtual machine that is running a Fedora system as the guest system. The username is `courseuser`, the password `dunecourse`. In this virtual machine, you have a folder `course` in your home directory, which has the structure shown to the left.

The `dune` subfolder contains the sources of all the Dune modules needed for the course. The course examples are located in the module `dune-pdelab-tutorials`.

Dune uses a `CMake` based build system. In `CMake`, there is a clean separation between the *source directory*, which usually is under version control (here: the `dune` subdirectory) and the *build directory*, where built executables, program output and such are placed. In this course the build directory is `release-build` where all programs are compiled with all optimizations.

The build directory mirrors the structure of the source directory. You should by default navigate to the `release-build` subdirectory of the current exercise and work there. If you need to access the sources of a given example, you can follow the symlink `src_dir` in any build subdirectory to switch to the corresponding subdirectory of the source tree.

As already mentioned the course examples are in the `dune-pdelab-tutorials` module. Each subdirectory of this module corresponds to one tutorial (`tutorialxy`). They all have the same internal structure: The `src` subdirectory contains the example code which was shown in the lecture. The `doc` subdirectory contains the latex sources of a detailed explanation of the tutorial. The `exercise` subdirectory, which is relevant for this course, is subdivided even further: `task` contains the code skeleton to work on during the exercise, `doc` contains the sources of the exercise sheet and `solution` contains what you expect it to.

The shell scripts in `course` set up the course material. This has already been done, so please do not call any of these scripts unless the course team tells you to. The two `opts` files do configure the two separate builds, there is also no need to touch these.

In case you are not familiar with UNIX system in general, here is a small cheat sheet of commands that we think are necessary for the exercises:

- `cd <dir>` changes the current working directory to `dir` (to the home directory, if omitted).
- `ls` `lis`ts the contents of the current working directory
- `pwd` prints the current working directory
- `g++ <options> <sources>` compiles C++ sources (only needed in the C++ exercise)
- `make <executablename>` (re)builds executables in the current build directory. If the executable name is omitted all executables in the current directory are built.
- `paraview` is a visualization program for VTK files