# Read access to home directories for group members

For all group members to be able to see each other's home directories all group members must belong to the same BioHPC group, in this example let it be `mylab`. The simplest procedure (to be followed by each group member) would be the following:

Change group of the home directory to `mylab`:

```
cd
chgrp mylab .
```

Make sure the home directory is readable and browsable by the group, but off-limits to 'others':

```
cd
chmod g+rx .
chmod o= .
```

After these operations, all `mylab` group members (and only them) will be able to `cd` to the user's home directory and list its content. However the subdirectories may still be blocked from access. Now let's say there is a subdirectory there called `some_dir` which is supposed to be visible to the group. To make sure it is, recursively change its permissions pertaining to the 'group' and 'others' users tiers:

```
cd
chmod -R go+rX some_dir
```

(note the capital '`X`' - this ensures that all subfolders of `some_dir` will be given the '`x`' bit and be therefore browsable by 'others'). Do this for all subfolders you want the group to have access to. On the other hand, if you want to make some other subfolder private (say, `some_private_dir`, not accessible to the group), remove permissions for 'group' and 'others':

```
cd
chmod go= some_private_dir
```

Do this for all subfolders of your home director you want to keep private. This procedure will set access to the stuff already existing in your home directory. In order for the new files and directories to also automatically have such access properties, you need to edit your `$HOME/.bashrc` file and add the following statement:

```
umask 0022
```

This will (try to) make sure that only the '`w`' permission is removed for 'group' and 'others' tiers for files and directories newly created by a user (*i.e.*, the new files will have permission `rw-r--r--` and new directories will have permissions `rwxr-xr-x`). Unfortunately, while setting `umask` is sufficient in most cases, the actual new file permissions often depend on the way the files arrived on the machine. For example, if some non-standard transfer tools are used, they may impose their own additional masks, which will change the end result. It is therefore always advisable to check permissions on the newly generated/transferred data and adjust them if needed.