Create a new commit containing the current contents of the index and the given log message describing the changes. The new commit is a direct child of HEAD, usually the tip of the current branch, and the branch is updated to point to it (unless no branch is associated with the working tree, in which case HEAD is "detached" as described in [git-checkout[1]](https://git-scm.com/docs/git-checkout)).

The content to be committed can be specified in several ways:

1. by using [git-add[1]](https://git-scm.com/docs/git-add) to incrementally "add" changes to the index before using the **commit** command (Note: even modified files must be "added");
2. by using [git-rm[1]](https://git-scm.com/docs/git-rm) to remove files from the working tree and the index, again before using the **commit** command;
3. by listing files as arguments to the **commit** command (without --interactive or --patch switch), in which case the commit will ignore changes staged in the index, and instead record the current content of the listed files (which must already be known to Git);
4. by using the -a switch with the **commit** command to automatically "add" changes from all known files (i.e. all files that are already listed in the index) and to automatically "rm" files in the index that have been removed from the working tree, and then perform the actual commit;
5. by using the --interactive or --patch switches with the **commit** command to decide one by one which files or hunks should be part of the commit in addition to contents in the index, before finalizing the operation. See the “Interactive Mode” section of [git-add[1]](https://git-scm.com/docs/git-add) to learn how to operate these modes.