Manipulating Filenames

The following functions can be used
to manipulate filenames in a portable way
over multiple platforms.

paths.filep(path)

Return a boolean indicating whether `path`
refers to an existing file.

paths.dirp(path)

Return a boolean indicating whether `path`
refers to an existing directory.

paths.basename(path,[suffix])

Return the last path component of `path`
and optionally strip the suffix `suffix`.
This is similar to the well know shell command `"basename"`.

paths.dirname(path)

Return the name of directory containing file `path`.
This is similar to the well known shell command `"dirname"`.

paths.extname(path)

Return the extension of the `path` or nil if none is found.

paths.concat([path1,….,pathn])

Concatenates relative filenames.

First this function computes the full filename
of `path1` relative to the current directory.
Then it successively computes the full filenames
of arguments `path2` to `pathn` relative to
the filename returned for the previous argument.
Finally the last result is returned.

Calling this function without argument returns the
full name of the current directory.

paths.cwd()

Return the full path of the current working directory.

paths.execdir()

Return the name of the directory containing the
current Lua executable.
When the module `paths` is first loaded,
this information is used to relocate
the variables indicating
the location of the various Torch components.

paths.tmpname()

Return the name of a temporary file.
All the temporaty files whose name was obtained in this way
are removed when Lua exits.

This function should be preferred over `os.tmpname()`
because it makes sure that the files are removed on exit.
In addition, `os.tmpname()` under windows often returns filenames
for which the user has no permission to write.