



QUICK START

Getting started with Torch
Five simple examples

DOCUMENTATION

Tutorials, Demos, Examples
Package Documentation
Developer Documentation

Getting started with Torch

Installing Torch

We provide a simple installation process for Torch on Mac OS X and Ubuntu 12+:

Torch can be installed to your home folder in `~/torch` by running these three commands:

```
# in a terminal, run the commands WITHOUT sudo
git clone https://github.com/torch/distro.git ~/torch --recursive
cd ~/torch; bash install-deps;
./install.sh
```

The **first script** installs the basic package dependencies that LuaJIT and Torch require. The **second script** installs **LuaJIT**, **LuaRocks**, and then uses LuaRocks (the lua package manager) to install core packages like **torch**, **nn** and **paths**, as well as a few other packages.

The script adds torch to your PATH variable. You just have to source it once to refresh your env variables. The installation script will detect what is your current shell and modify the path in the correct configuration file.

```
# On Linux with bash
source ~/.bashrc
# On Linux with zsh
source ~/.zshrc
# On OSX or in Linux with none of the above.
source ~/.profile
```

If you ever need to uninstall torch, simply run the command:

```
rm -rf ~/torch
```

If you want to install torch with Lua 5.2 instead of LuaJIT, simply run:

```
git clone https://github.com/torch/distro.git ~/torch --recursive
cd ~/torch

# clean old torch installation
./clean.sh
# optional clean command (for older torch versions)
# curl -s https://raw.githubusercontent.com/torch/ezinstall/master/clean-old.sh | bash

# https://github.com/torch/distro : set env to use lua
TORCH_LUA_VERSION=LUA52 ./install.sh
```

New packages can be installed using Luarocks from the command-line:

```
# run luarocks WITHOUT sudo
$ luarocks install image
$ luarocks list
```

Once installed you can run torch with the command `th` from you prompt!

The easiest way to learn and experiment with Torch is by starting an interactive session (also known as the torch read-eval-print loop or **TREPL**):

```

$ th

      _____ | Torch7
     /  _  /  _  /  / | Scientific computing for Lua.
    /  /  /  \  \  \  \ |
   /  /  \  \  \  \  \ | https://github.com/torch
   /  /  \  \  \  \  \ | http://torch.ch

th> torch.Tensor{1,2,3}
  1
  2
  3
[torch.DoubleTensor of dimension 3]

th>

```

To exit the interactive session, type `^c` twice — the control key together with the `c` key, twice, or type `os.exit()`. Once the user has entered a complete expression, such as `1 + 2`, and hits enter, the interactive session evaluates the expression and shows its value.

To evaluate expressions written in a source file `file.lua`, write `dofile "file.lua"`.

To run code in a file non-interactively, you can give it as the first argument to the `th` command::

```
$ th file.lua
```

There are various ways to run Lua code and provide options, similar to those available for the `perl` and `ruby` programs:

```

$ th -h
Usage: th [options] [script.lua [arguments]]

Options:
  -l name           load library name
  -e statement      execute statement
  -h,--help         print this help
  -a,--async        preload async (libuv) and start async repl (BETA)
  -g,--globals      monitor global variables (print a warning on creation/access)
  -gg,--gglobals    monitor global variables (throw an error on creation/access)
  -x,--gfx          start gfx server and load gfx env
  -i,--interactive  enter the REPL after executing a script

```

TREPL is full of convenient features likes:

- Tab-completion on nested namespaces
- Tab-completion on disk files (when opening a string)
- History (preserved between sessions)
- Pretty print (table introspection and coloring)
- Auto-print after eval (can be stopped with `;`)
- Each command is profiled, timing is reported
- No need for `'='` to print
- Easy help with: `? funcname`
- Self help: `?`
- Shell commands with: `$ cmd` (example: `$ ls`)

Next steps

In addition to this manual, there are various other resources that may help new users get started with torch, all summarized in this **Cheatsheet**

The cheatsheet provides links to tutorials, demos, package summaries and a lot of useful information.

If you have a question, please come join the **Torch users mailing list**

Next →

Torch7 maintained by Ronan, Clément, Koray and Soumith.