nvm Node version manager

CONCEPTS

> nvm is a shell function, not a program: it only exists in shells that source nvm.sh, and it changes PATH of that shell only.

.nvmrc version in project
↓ `nvm use`
nvm shell function
↓ `sets PATH`
~/.nvm/versions installed Node
↓ `first on PATH`
node, npm this shell only

> Global npm packages belong to one Node version; a new version starts without them.

INSTALL NVM

`curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.8/install.sh | bash`
Install nvm with curl

`nvm --version`
Show the nvm version

`command -v nvm`
Check that nvm is loaded

INSTALL AND USE

`nvm install <version>`
Install a Node version

`nvm install --lts`
Install the newest LTS

`nvm install node`
Install the latest release

`nvm install`
Install the version from .nvmrc

`nvm use <version>`
Use a version in this shell

`nvm use --lts`
Use the newest LTS

INSTALL AND USE (cont.)

`nvm use`
Use the version from .nvmrc

`nvm install <version> --reinstall-packages-from=<old>`
Install and copy global packages

INSPECT VERSIONS

`nvm ls`
List installed versions

`nvm ls-remote`
List all installable versions

`nvm ls-remote --lts`
List LTS versions only

`nvm current`
Show the active version

`nvm uninstall <version>`
Remove an installed version

ALIASES AND DEFAULTS

`nvm alias default <version>`
Set the version new shells start with

`nvm alias default lts/*`
Default to the newest LTS

RUN WITHOUT SWITCHING

`nvm run <version> <script.js>`
Run node under that version

`nvm exec <version> <command>`
Run any command under that version

.NVMRC AND PACKAGES

`echo "lts/*" > .nvmrc`
Pin the project to the newest LTS

`node -v > .nvmrc`
Pin the project to the running version

`nvm install && nvm use`
Install then use the pinned version