Git auto-fetch

Automatically fetches all changes from all remotes while you are working in a git-initialized directory.

To use it, add `git-auto-fetch` to the plugins array in your zshrc file:

```
plugins=(... git-auto-fetch)
```

Usage

Every time the command prompt is shown all remotes will be fetched in the background. By default, `git-auto-fetch` will be triggered only if the last auto-fetch was done at least 60 seconds ago. You can change the fetch interval in your .zshrc:

```
GIT_AUTO_FETCH_INTERVAL=1200 # in seconds
```

A log of `git fetch --all` will be saved in `.git/FETCH_LOG`.

Toggle auto-fetch per folder

If you are using a mobile connection or for any other reason you can disable git-auto-fetch for any folder:

```
$ cd to/your/project
$ git-auto-fetch
disabled
$ git-auto-fetch
enabled
```

Caveats

Automatically fetching all changes defeats the purpose of `git push --force-with-lease`, and makes it behave like `git push --force` in some cases. For example:

Consider that you made some changes and possibly rebased some stuff, which means you'll need to use `--force-with-lease` to overwrite the remote history of a branch. Between the time when you make the changes (maybe do a `git log`) and the time when you `git push`, it's possible that someone else updates the branch you're working on.

If `git-auto-fetch` triggers then, you'll have fetched the remote changes without knowing it, and even though you're running the push with `--force-with-lease`, git will overwrite the recent changes because you already have them in your local repository. The `git push --force-with-lease` docs talk about possible solutions to this problem.