Insurance

Trusted Choice®

GIT Insurance
245 NE Venture Dr Suite 1
Waukee, IA 50263

515-897-4448
Fax: 888-651-7816

Our Agency ⌄   Products ⌄   Contact Us ⌄

Request A Free Quote

Get A Quote ›

BannerLife™   Prudential   Nationwide®   Nationwide®   PROGRESSIVE®

/home/peter/predicting_boastful_resistance/

Name	Size	Type
abstract_1.docx	3,1 kB	Microsoft Word Document
abstract_2.docx	6,1 kB	Microsoft Word Document
abstract_4.docx	9,2 kB	Microsoft Word Document
abstract_4 (copy 2).docx	11,8 kB	Microsoft Word Document
abstract_4-corrections.docx	9,2 kB	Microsoft Word Document
abstract_final.docx	12,8 kB	Microsoft Word Document
abstract_final (copy 1)_LAST (copy 1).docx	4,1 kB	Microsoft Word Document
abstract_final-_REALLY_FINALLL-uargh.docx	11,8 kB	Microsoft Word Document
abstract_final-upload-this.docx	16,4 kB	Microsoft Word Document

Why?

- GIT as solution for versioning files

- GIT as solution for collaboration

- GIT for distributing software/code

/home/peter/predicting_boastful_resistance/

Name	Size	Type
abstract_1.docx	3,1 kB	Microsoft Word Document
abstract_2.docx	6,1 kB	Microsoft Word Document
abstract_4.docx	9,2 kB	Microsoft Word Document
abstract_4 (copy 2).docx	11,8 kB	Microsoft Word Document
abstract_4-corrections.docx	9,2 kB	Microsoft Word Document
abstract_final.docx	12,8 kB	Microsoft Word Document
abstract_final (copy 1)_LAST (copy 1).docx	4,1 kB	Microsoft Word Document
abstract_final-_REALLY_FINALLL-uargh.docx	11,8 kB	Microsoft Word Document
abstract_final-upload-this.docx	16,4 kB	Microsoft Word Document

master

What?

- versions connected via tree-like graph, each node is a commit

- think of the commit as change ("patches")

- operations on the tree of commits: branch, rebase, cherry-pick, merge, sending commits (push/pull)

How?

- command line

- GUIs, IDE integration

- API (eg. Python etc.)

example.py

after successful merge

merge

example.py

modified by Alice

example.py

modified by Bob

```
638b251c6 (Keisuke Fujii      2018-09-28 08:54:29 +0200)  13
638b251c6 (Keisuke Fujii      2018-09-28 08:54:29 +0200)  14 # Used as a sentinel value to indicate a all dimensions
638b251c6 (Keisuke Fujii      2018-09-28 08:54:29 +0200)  15 ALL_DIMS = ReprObject('<all-dims>')
adcc4836c (Stephan Hoyer      2014-04-30 01:56:35 -0700)  16
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  17
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  18 class ImplementsArrayReduce(object):
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  19     @classmethod
bf299aa6f (Stephan Hoyer      2015-01-03 22:14:51 +0200)  20     def _reduce_method(cls, func, include_skipna, numeric_only):
bf299aa6f (Stephan Hoyer      2015-01-03 22:14:51 +0200)  21         if include_skipna:
bf299aa6f (Stephan Hoyer      2015-01-03 22:14:51 +0200)  22             def wrapped_func(self, dim=None, axis=None, skipna=None,
6d55f9990 (Tom Nicholas       2018-10-30 01:01:07 +0000)  23                              **kwargs):
6d55f9990 (Tom Nicholas       2018-10-30 01:01:07 +0000)  24                 return self.reduce(func, dim, axis,
1d2b36974 (Stephan Hoyer      2015-03-29 23:00:54 -0700)  25                                    skipna=skipna, allow_lazy=True, **kwargs)
bf299aa6f (Stephan Hoyer      2015-01-03 22:14:51 +0200)  26         else:
6963164db (Stephan Hoyer      2019-01-07 23:21:41 -0800)  27             def wrapped_func(self, dim=None, axis=None,  # type: ignore
c22f6e5ac (Stephan Hoyer      2015-03-26 20:53:59 -0700)  28                              **kwargs):
6d55f9990 (Tom Nicholas       2018-10-30 01:01:07 +0000)  29                 return self.reduce(func, dim, axis,
c22f6e5ac (Stephan Hoyer      2015-03-26 20:53:59 -0700)  30                                    allow_lazy=True, **kwargs)
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  31         return wrapped_func
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  32
0b9ab2d12 (Keisuke Fujii      2018-08-16 15:59:32 +0900)  33     _reduce_extra_args_docstring = dedent("""\
0b9ab2d12 (Keisuke Fujii      2018-08-16 15:59:32 +0900)  34         dim : str or sequence of str, optional
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  35             Dimension(s) over which to apply `{name}`.
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  36         axis : int or sequence of int, optional
39f0d0122 (Stephan Hoyer      2014-08-12 22:09:56 -0700)  37             Axis(es) over which to apply `{name}`. Only one of the 'dim'
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  38             and 'axis' arguments can be supplied. If neither are supplied, then
0b9ab2d12 (Keisuke Fujii      2018-08-16 15:59:32 +0900)  39             `{name}` is calculated over axes.""")
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  40
0b9ab2d12 (Keisuke Fujii      2018-08-16 15:59:32 +0900)  41     _cum_extra_args_docstring = dedent("""\
0b9ab2d12 (Keisuke Fujii      2018-08-16 15:59:32 +0900)  42         dim : str or sequence of str, optional
9cf107b52 (Phillip Wolfram    2016-10-03 15:05:32 -0600)  43             Dimension over which to apply `{name}`.
9cf107b52 (Phillip Wolfram    2016-10-03 15:05:32 -0600)  44         axis : int or sequence of int, optional
9cf107b52 (Phillip Wolfram    2016-10-03 15:05:32 -0600)  45             Axis over which to apply `{name}`. Only one of the 'dim'
0b9ab2d12 (Keisuke Fujii      2018-08-16 15:59:32 +0900)  46             and 'axis' arguments can be supplied.""")
9cf107b52 (Phillip Wolfram    2016-10-03 15:05:32 -0600)  47
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  48
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  49 class ImplementsDatasetReduce(object):
ad9a9135c (Stephan Hoyer      2014-02-13 00:25:45 -0500)  50     @classmethod
bf299aa6f (Stephan Hoyer      2015-01-03 22:14:51 +0200)  51     def _reduce_method(cls, func, include_skipna, numeric_only):
bf299aa6f (Stephan Hoyer      2015-01-03 22:14:51 +0200)  52         if include_skipna:
6d55f9990 (Tom Nicholas       2018-10-30 01:01:07 +0000)  53             def wrapped_func(self, dim=None, skipna=None,
bf299aa6f (Stephan Hoyer      2015-01-03 22:14:51 +0200)  54                              **kwargs):
6d55f9990 (Tom Nicholas       2018-10-30 01:01:07 +0000)  55                 return self.reduce(func, dim, skipna=skipna,
1d2b36974 (Stephan Hoyer      2015-03-29 23:00:54 -0700)  56                                    numeric_only=numeric_only, allow_lazy=True,
1d2b36974 (Stephan Hoyer      2015-03-29 23:00:54 -0700)  57                                    **kwargs)
bf299aa6f (Stephan Hoyer      2015-01-03 22:14:51 +0200)  58         else:
6963164db (Stephan Hoyer      2019-01-07 23:21:41 -0800)  59             def wrapped_func(self, dim=None, **kwargs):  # type: ignore
6d55f9990 (Tom Nicholas       2018-10-30 01:01:07 +0000)  60                 return self.reduce(func, dim,
1d2b36974 (Stephan Hoyer      2015-03-29 23:00:54 -0700)  61                                    numeric_only=numeric_only, allow_lazy=True,
1d2b36974 (Stephan Hoyer      2015-03-29 23:00:54 -0700)  62                                    **kwargs)
18c7b9dbf (Stephan Hoyer      2014-06-22 01:07:03 -0700)  63         return wrapped_func
```
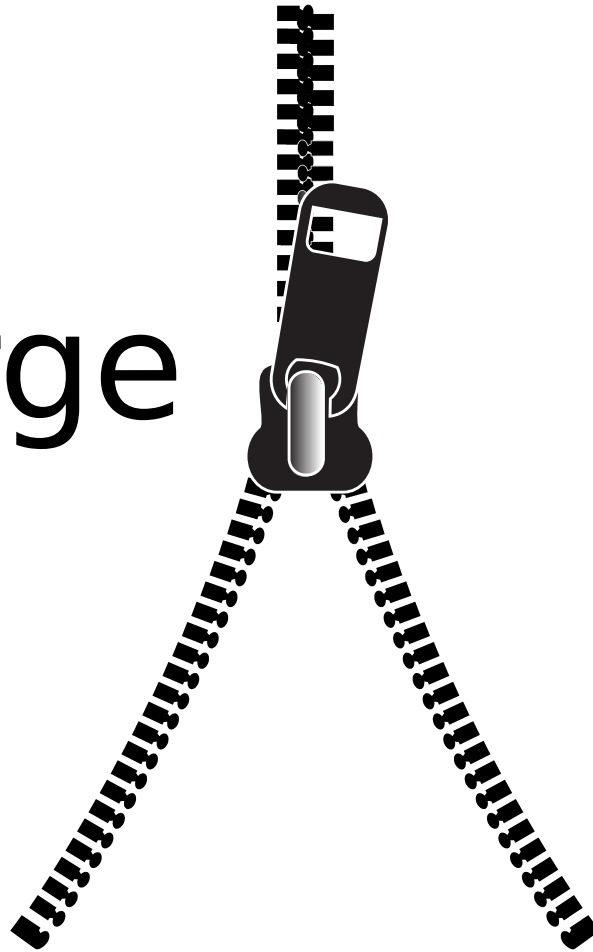
git-avoid-area(1) Manual Page

NAME

git-avoid-area — avoid the non-rev-listed applied areas for all rebased branches

SYNOPSIS

git-avoid-area --minimize-seize-submodule [--conceive-flush-area | --hit-tip |
--sidestep-mutilate-tip]

DESCRIPTION

git-avoid-area avoids all upstream areas opposite of all cloned non-archived unstaged heads, and all checked indices are forward-ported to INTENSIFY_BRANCH by git-foster-commit.

The --supervise-utilize-file argument can be used to note an area for the upstream that is counted by a passive change, so it is in some cases a small chance that a rev-listed failure may prevent passive diffing of the specified objects. Any allotting of a subtree that configures a tip immediately after can be cherry-picked with git-douse-path. It is sometimes a rare chance that a bundled error can prevent temporary fetching of the remoted paths.

Source: https://git-man-page-generator.lokaltog.net/

git-push(1) Manual Page

NAME

git-push - Update remote refs along with associated objects

SYNOPSIS

git push [--all | --mirror | --tags] [--follow-tags] [--atomic] [-n | --dry-run] [--receive-pack=<git-receive-pack>]
  [--repo=<repository>] [-f | --force] [-d | --delete] [--prune] [-v | --verbose]
  [-u | --set-upstream] [-o <string> | --push-option=<string>]
  [--[no-]signed|--signed=(true|false|if-asked)]
  [--force-with-lease[=<refname>[:<expect>]]]
  [--no-verify] [<repository> [<refspec>...]]

DESCRIPTION

Updates remote refs using local refs, while sending objects necessary to complete the given refs.

You can make interesting things happen to a repository every time you push into it, by setting up *hooks* there. See documentation for git-receive-pack(1).

When the command line does not specify where to push with the `<repository>` argument, `branch.*.remote` configuration for the current branch is consulted to determine where to push. If the configuration is missing, it defaults to *origin*.

git-rebase(1) Manual Page

NAME

git-rebase - Reapply commits on top of another base tip

SYNOPSIS

```
git rebase [-i | --interactive] [<options>] [--exec <cmd>] [--onto <newbase>]
    [<upstream> [<branch>]]
git rebase [-i | --interactive] [<options>] [--exec <cmd>] [--onto <newbase>]
    --root [<branch>]
git rebase --continue | --skip | --abort | --quit | --edit-todo | --show-current-patch
```

DESCRIPTION

If <branch> is specified, *git rebase* will perform an automatic `git checkout <branch>` before doing anything else. Otherwise it remains on the current branch.

If <upstream> is not specified, the upstream configured in branch.<name>.remote and branch.<name>.merge options will be used (see git-config(1) for details) and the `--fork-point` option is assumed. If you are currently not on any branch or if the current branch does not have a configured upstream, the rebase will abort.

Source:
https://git.github.io/htmldocs/git-push.html
https://git.github.io/htmldocs/git-rebase.html
https://stevebennett.me/2012/02/24/10-things-i-hate-about-git/

Source:
https://xkcd.com/1597

More reasons why GIT is bad:

https://svnvsgit.com/

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Source:
https://xkcd.com/1296/