

**git add** 命令将文件内容添加到索引(将修改添加到暂存区)。也就是将要提交的文件的信息添加到索引库中。

## 简介

```
git add [--verbose | -v] [--dry-run | -n] [--force | -f] [--interactive | -i] [--patch | -p]
        [--edit | -e] [--[no-]all | --[no-]ignore-removal | [--update | -u]]
        [--intent-to-add | -N] [--refresh] [--ignore-errors] [--ignore-missing]
        [--chmod=(+|-)x] [--] [<paths>...]
```

## 描述

此命令将要提交的文件的信息添加到索引库中(将修改添加到暂存区)，以准备为下一次提交分段的内容。它通常将现有路径的当前内容作为一个整体添加，但是通过一些选项，它也可以用于添加内容，只对所应用的工作树文件进行一些更改，或删除工作树中不存在的路径了。

“索引”保存工作树内容的快照，并且将该快照作为下一个提交的内容。因此，在对工作树进行任何更改之后，并且在运行 **git commit** 命令之前，必须使用 **git add** 命令将任何新的或修改的文件添加到索引。

该命令可以在提交之前多次执行。它只在运行 **git add** 命令时添加指定文件的内容；如果希望随后的更改包含在下一个提交中，那么必须再次运行 **git add** 将新的内容添加到索引。

**git status** 命令可用于获取哪些文件具有为下一次提交分段的更改的摘要。

默认情况下，**git add** 命令不会添加忽略的文件。如果在命令行上显式指定了任何忽略的文件，**git add** 命令都将失败，并显示一个忽略的文件列表。由 Git 执行的目录递归或文件名遍历所导致的忽略文件将被默认忽略。**git add** 命令可以用 **-f(force)** 选项添加被忽略的文件。

## 示例

以下是一些示例 -

添加 **documentation** 目录及其子目录下所有 **\*.txt** 文件的内容：

```
$ git add documentation/*.txt
```

注意，在这个例子中，星号\*是从 shell 引用的；这允许命令包含来自 **Documentation/** 目录和子目录的文件。

将所有 **git-\*.sh** 脚本内容添加：

```
$ git add git-*.sh
```

因为这个例子让 shell 扩展星号(即明确列出文件)，所以它不考虑子目录中的文件，如：**subdir/git-foo.sh** 这样的文件不会被添加。

## 基本用法

```
git add <path>
```

通常是通过 `git add <path>` 的形式把 `<path>` 添加到索引库中, `<path>` 可以是文件也可以是目录。

git 不仅能判断出 `<path>` 中, 修改(不包括已删除)的文件, 还能判断出新添的文件, 并把它们的信息添加到索引库中。

```
$ git add . # 将所有修改添加到暂存区
```

```
$ git add * # Ant 风格添加修改
```

```
$ git add *Controller # 将以 Controller 结尾的文件的所有修改添加到暂存区
```

```
$ git add Hello* # 将所有以 Hello 开头的文件的修改添加到暂存区 例  
如: HelloWorld.txt, Hello.java, HelloGit.txt ...
```

```
$ git add Hello? # 将以 Hello 开头后面只有一位的文件的修改提交到暂存区 例  
如: Hello1.txt, HelloA.java 如果是 HelloGit.txt 或者 Hello.java 是不会被添加的
```

Shell

`git add -u [<path>]`: 把 `<path>` 中所有跟踪文件中被修改过或已删除文件的信息添加到索引库。它不会处理那些不被跟踪的文件。省略 `<path>` 表示 `.`, 即当前目录。

`git add -A`: 表示把中所有跟踪文件中被修改过或已删除文件和所有未跟踪的文件信息添加到索引库。省略 `<path>` 表示 `.`, 即当前目录。

`git add -i`

我们可以通过 `git add -i [<path>]` 命令查看中被所有修改过或已删除文件但没有提交的文件, 并通过其 `revert` 子命令可以查看 `<path>` 中所有未跟踪的文件, 同时进入一个子命令系统。

比如:

```
$ git add -i
```

	staged	unstaged path
1:	+0/-0	nothing branch/t.txt
2:	+0/-0	nothing branch/t2.txt
3:	unchanged	+1/-0 readme.txt

\*\*\* Commands \*\*\*

1: [s]tatus	2: [u]pdate	3: [r]evert	4: [a]dd untracked
5: [p]atch	6: [d]iff	7: [q]uit	8: [h]elp

这里的 `t.txt` 和 `t2.txt` 表示已经被执行了 `git add`, 待提交。即已经添加到索引库中。  
`readme.txt` 表示已经处于 `tracked` 下, 它被修改了, 但是还没有执行 `git add`。即还没添加到索引库中。