



TALKS

The Art of Lucid Knowledge Sharing

Why?



beer & rashomon



git

A Developer's
Version Control Swiss Knife

Sam

<http://samof76.me>

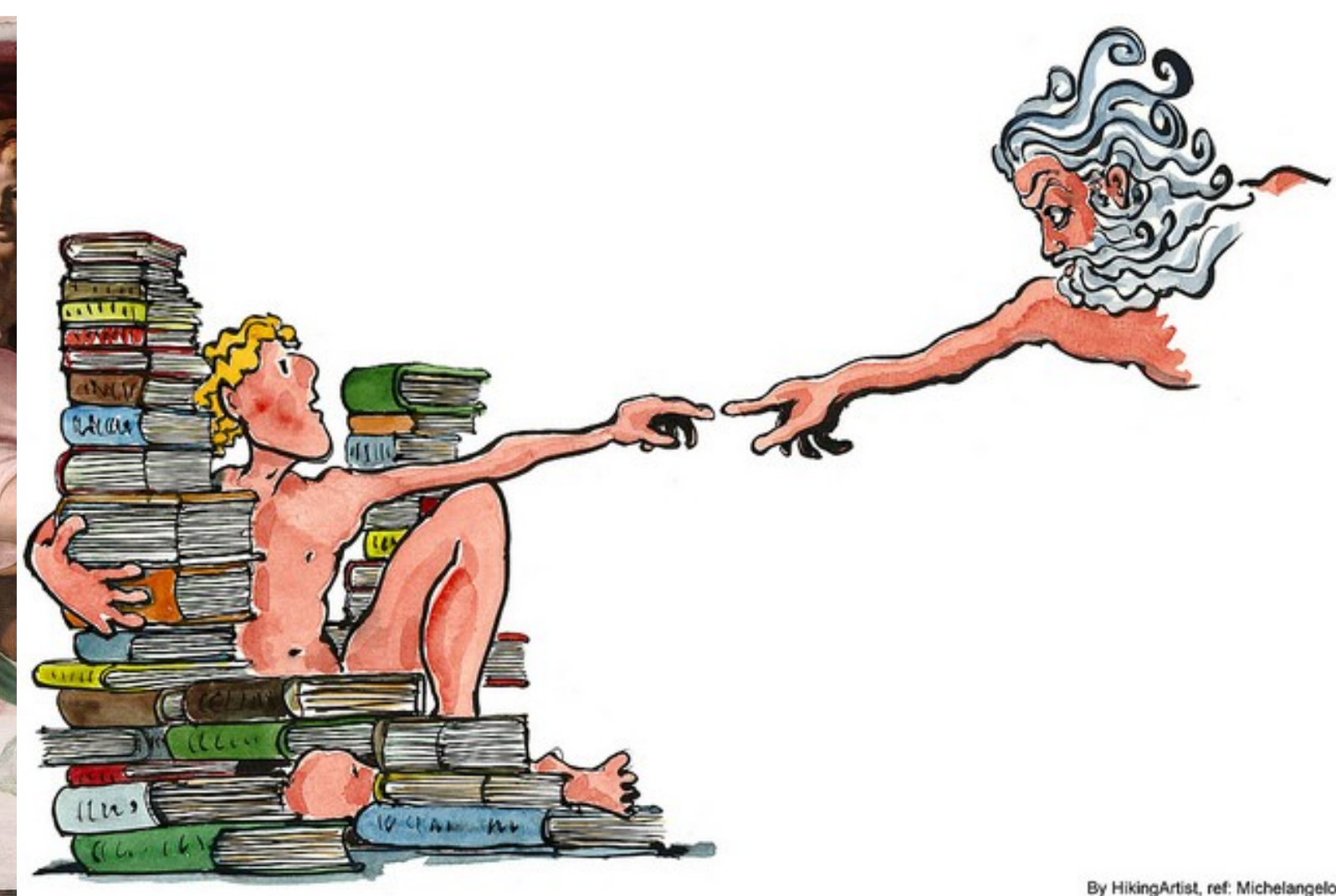


Developer

/di'veləpər/

Noun:

- A person or organization that develops something.
- A person who grows or matures at a specified time or rate.



Version

/ˈvər-zhən, -shən/

Noun:

A particular form of something differing in certain respects from an earlier form or other forms of the same type of thing.

Verb:

Create a new version of: "it's the software for you if you need versioning and group editing".

Control

/kən'trōl/

Noun:

The power to influence or direct people's behavior or the course of events.

Verb:

Determine the behavior or supervise the running of.



Swiss Knife?

You might just need the knife all the time. But also helps when you need a screwdriver.

The quality of the utilities at engineered to last and so quality is an important criteria



Any Other words?

Any Other words?

One More!

Distributed

The future is
here – it's just
not widely
distributed.

William Gibson



Lets Dive in!

Tennessee State Fair - Diving Pig

This diving pig was from the "Randall's Racing Pigs" show held at the Tennessee State Fair in Nashville this past september.

One Advice



One Advice

Enjoy the fall. Its Free!!!



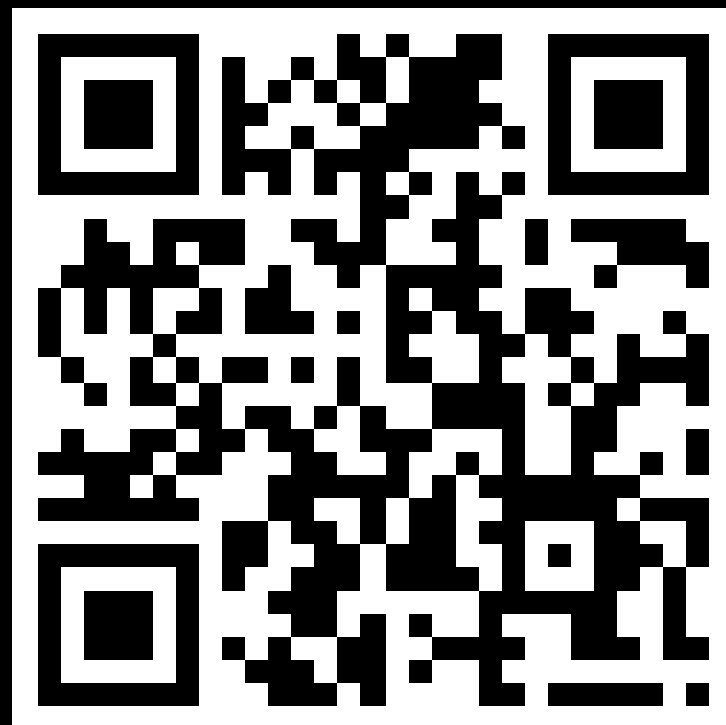
Windows

Mac

Linux

<http://17z.in/1R> <http://17z.in/1Q> <http://17z.in/1S>

Windows

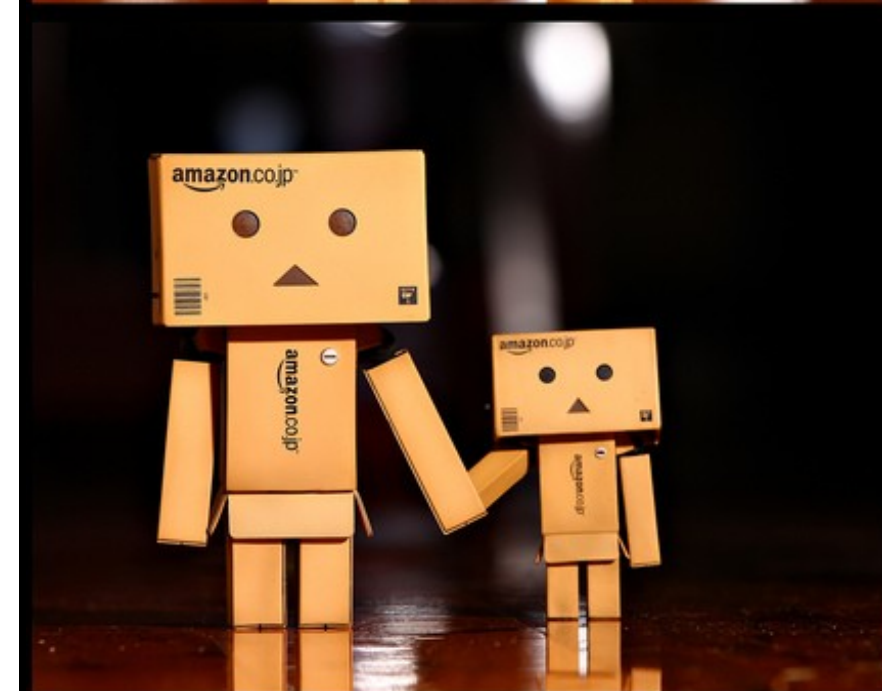


Mac



Linux





```
% git help
usage: git [--version] [--exec-path[=<path>]] [--html-path]
[-p|--paginate|--no-pager] [--no-replace-objects]
[--bare] [--git-dir=<path>] [--work-tree=<path>]
[-c name=value] [--help]
<command> [<args>]
```

The most commonly used git commands are:

add	Add file contents to the index
bisect	Find by binary search the change that introduced a bug
branch	List, create, or delete branches
checkout	Checkout a branch or paths to the working tree
clone	Clone a repository into a new directory
commit	Record changes to the repository
diff	Show changes between commits, commit and working tree, etc
fetch	Download objects and refs from another repository
grep	Print lines matching a pattern
init	Create an empty git repository or reinitialize an existing one
log	Show commit logs
merge	Join two or more development histories together
mv	Move or rename a file, a directory, or a symlink
pull	Fetch from and merge with another repository or a local branch
push	Update remote refs along with associated objects
rebase	Forward-port local commits to the updated upstream head
reset	Reset current HEAD to the specified state
rm	Remove files from the working tree and from the index
show	Show various types of objects
status	Show the working tree status
tag	Create, list, delete or verify a tag object signed with GPG

See 'git help <command>' for more information on a specific command.



```
$ git init
```

```
$ git add .
```

```
$ git commit -m "Hello world! My first git commit :)"
```

```
$ git reset --hard
```

```
$ git commit -a -m "Just another time to see what happens"
```



```
$ git add readme.txt Documentation
```

```
$ git rm service.py utils.py
```

```
$ git rm -r dist/packages/
```

```
$ git mv param.py params.py
```




```
$ git log
```

```
commit 25f6d3d663d74e5f65148dd667d9061bf3edf9de
```

```
Author: Samuel Vijaykumar, Madireddy <dhruv.samuel@gmail.com>
```

```
Date: Thu Jan 12 23:00:49 2012 +0530
```

```
Hello world! My first git commit :)
```

```
commit aead96adad5ad5ddfbcfbfa47ff000c72f036027d
```

```
Author: Samuel Vijaykumar, Madireddy <dhruv.samuel@gmail.com>
```

```
Date: Thu Jan 12 22:57:49 2012 +0530
```

```
Just another time to see what happens
```



```
$ git reset --hard aead96
```

```
$ git checkout aead96
```

```
$ git checkout master
```

```
$ git checkout aead96 zooper.py sooper.py
```

```
$ git checkout master
```

```
$ git checkout :/"Hello world"
```

```
$ git checkout master~2
```



```
$ git clone https://github.com/samof76/talks.git
```

```
$ git clone git://github.com/samof76/talks.git
```

```
$ git clone github:github.com:samof76/talks.git
```

```
$ git pull
```

```
253     'iframe'      : false,  
254     'orientation' : 'bottom',  
255     'width'       : '100%',  
256   });  
257   toolbar.render();  
  
258   </code></pre>  
259 </div>
```

```
253     'iframe'      : false,  
254     'orientation' : 'top',  
255     'width'       : '100%',  
256   });  
257   toolbar.render();  
258   toolbar.additem(new ToolbarItem({  
259     'content' : 'New Description',  
260     'label'   : 'New Label'  
261   }));  
262   </code></pre>  
263 </div>
```

\$ git diff

\$ git diff "@{yesterday}"

\$ git diff 1b6d "master~2"

\$ git whatchanged --since="2 weeks ago"



Remember!



have no fear of perfection.
you'll never reach it.
salvador dali

references



Thank you!

