



# SKILLPILLS

## Skill Pill: Introduction to Git and Version Control

### Lecture 1: Git ready!

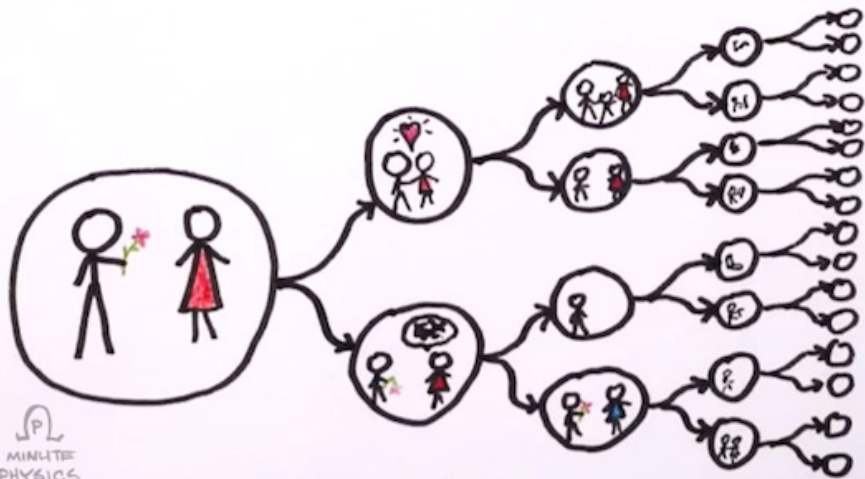
**James Schloss**

Okinawa Institute of Science and Technology  
[james.schloss@oist.jp](mailto:james.schloss@oist.jp)

June 1, 2016

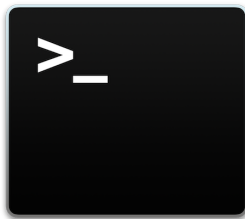


- 1 What is Version Control
- 2 Terminal Talk
- 3 Git basics
- 4 Working alone

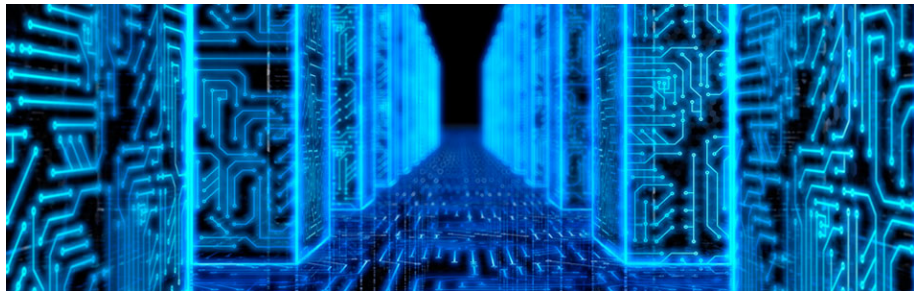


- Version control is a method that allows you to control different versions of things (Not necessarily universes).

- We had a terminal skill pill and I have included the cheatsheet from that.
- There is a GUI downloadable from GitHub called the **GitHub Desktop**. We will not be using this for religious reasons.
- Everything we do will be usable on Sango.



- A **repository** is a place to store code.
  - There are many sites to host your repository on (github, bitbucket), including your own local machine.
  - All of the essential parts of your repository can be found in **.git** file



Let's **git** started.

- To initialize a git repository, simply type **git init** in a directory (preferably empty for now)
- This creates a folder **.git/**, where all your git information is held.
- Git tracks **commits**. Check these commits with **git log**.
- **git status** checks any changes since the last commit.
- **git add** adds new files.
- **git commit** commits anything git status shows in **green**.



## EXERCISE

- 1 Open a terminal
- 2 Create a new directory and run **git init**
- 3 Create a file and run **git status**
- 4 Use a combination of **git add** and **git commit** to add a new file to the git repository.
- 5 Check the **git log**.

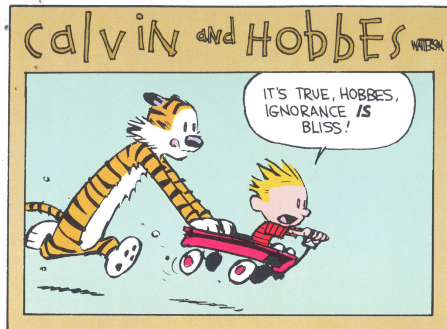
- Keep your repository clean! Do your best to commit as few images and data files as possible!
- You can do this by ignoring certain file extensions in a **.gitignore** file.

---

# Example gitignore  
configuration

```
*.log  
*.tar  
*.gz  
*.exe  
*.dat
```

---





## EXERCISE

- 1 Touch multiple files with various extensions, one of which should be **.dat**.
- 2 Ignore the **.dat** file, but commit all the others.
- 3 Be sure to write a clear message describing what you did.
- 4 Check the **git log**

Now we move to the fun\* stuff: working with **online repositories**.

- For this, we will be using **github**.
- To use an online repository, we need to synchronize our local machine with the master repository held elsewhere. This is done with the **clone** command.
- From here, you can do the following:
  - **git push** to push any changes you may have to the online repository.
  - **git pull** to take any changes from the

\*Here, the word *fun* is subject to interpretation.



Z

## EXERCISE

- 1 Clone our skillpill repository:

---

```
git clone git@github.com:oist/skillpill-git.git
```

---

or

---

```
git clone https://github.com/oist/skillpill-git.git
```

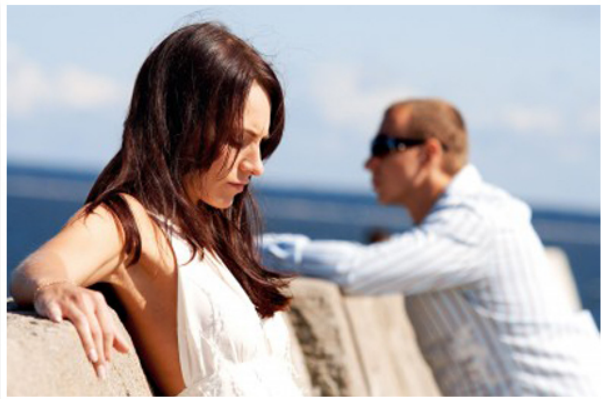
---

- 2 Working with a small group, make commits and push and pull stuff from that repo.

- git is not intuitive to start with, but it's the best way to work collaboratively with other people.
- The more you use it, the more you will like it. Think Stockholm syndrome.



# Coders and Relationships: 5 Signs He'll Never Commit



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFT	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

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

We now know how to work with both local and online repositories, but what about using different versions?

- **git checkout** allows you to view the repository at any old commit (found with **git log**).
- You may also checkout specific files like so:

---

```
git checkout a1e8fb5 hello.py
```

---

- Note that the most recent commit is **HEAD** and the one just before that is **HEAD~1**
- This command will be used later, so keep it in mind!

PRODUTOR DE GANHADORES DE MAIS DE US\$ 7.000.000,00

# CHECK HIM OUT

HARVEST X CHECK HER TWICE FIRST DOWN DASH

Produziu ganhadores de mais de US\$ 1.000.000,00 nos últimos 6 anos seguidos: 2012/2011/2010/2009

Harvest	Shore Call	Shore Bird
	Island Kitty	Twilight
		Harvest
		1 & 4 Kings
Check Her Twice	First Down Dash	Dash For Cash
	Future Favorite	Real Power House
		Disparo O' Da
		Digital To Go

**NÃO BRAS, PRODUZIU UOMO VERDE:**  
Última semana o star na final dos GP's Torneio Início, Brazilian Futurity e Ponto do Futuro ABQ 2013.

**CONVITADO 2013**  
10 parcerias  
por 700,00

Informações:  
(11) 3226.8952 | 3868.5257

Finally, what is actually happening with your commits under the hood?

- Git has a staging area before commits that can be checked with **git status**. Anything in **green** is staged.
- If you wish to unstage the commit, simply type **git reset**.
- **git reset** will work for individual files and you may go back to any commit in the history.

---

```
git reset HEAD~1
```

---

- If you wish to undo a commit entirely, use the **git revert** command.
- **git clean** will remove any untracked files.





## EXERCISE

- 1 Stage a commit
- 2 Unstage the commit
- 3 Make a commit
- 4 Undo the commit

- git is weird. It's not intuitive, but it's the best way to collaborate with people on open projects.
- Whenever you are using git, think about other people and how they will perceive your comments. **Would you be able to understand your own cryptic commit messages?**
- You will make mistakes. Don't worry about it. Your entire history is backed up already. Learn from your mistakes and don't make them again!
- Listen to git. It's smarter than you.

