H2S

Hello Terminal! Hello 42!

Kai kai@42.us.org

Ruchen ru@42.us.org

Summary: *This is the stuff to do on the first day!*

C⬤DE
for FUN

42

Contents

Chapter I

Preamble

42 is about face to face, self-directed, peer-to-peer learning.

1. Face to face: humans are all social creatures, we all like to meet and know each other. Even if you like to work alone, being part of group can be a really powerful source of mental health and life satisfaction. Be generous, help other people and be appreciative when other people help you.

2. Self-directed: you will meet some older mentors here, who are 42 cadets that have volunteered to help you learn. They are not instructors though. In fact, they may all be learning the same topics as you by your side. We learn by reading official documentation, tutorials, articles, and Q&A forums and then teaching each other! Don't be afraid to write some clumsy code, run it and then act on the error messages. That's why this is free: it's like studying on your own, but with supportive friends all around.

3. Peer-to-peer: The more we share information and knowledge with each other, the stronger pool of talent we have to build useful applications. At 42 you will grade each others' projects face to face. Be courteous and rigorous when you do this! Give constructive feedback so that your peers do not get mad at you, but do learn to code better they did before.

Chapter II

Intro to MacOS for Windows Users

Are you comfortable using these iMacs yet?

If not, here are some resources to get accustomed to the operating system:

Guide from MakeUseOf

Guide from Apple Support

Chapter III

View the Projects on Intra

Intra is the main student website of 42. Here you will find profiles of your fellow students, and details about all the projects we offer.

- Go to intra.42.fr in your web browser (Safari). Sign in with the username and password that came to your email, the same one that you logged into the computer with.

- The first page is your profile page. A few things to notice:

  - Your level meter, starting at 0.0%. As you turn in projects and have them approved by your peers you will level up!

  - Your "Correction Points" in the top left. At 42, there is a correction point economy. You must spend a correction point when a peer grades your work and you earn a correction point when you grade someone else's.

  - The "Evaluations" section. It is empty right now, but if you have an appointment to grade someone's project or vice versa that will appear here.

  - The "Projects" section. You will find links to the projects you are currently working on right here.

- In the Evaluations section, click on the "Manage Slots" button. It will show you a calendar of the current week. Find the current date and time. Then, click and drag to open an availability slot for the last hour that you plan to be here. (If you will be here until 4, click and drag from 3-4 pm). You should open a slot like this each day that you come to 42.

- Now, notice grey icons on the left side of the page.

  - Profile: The top, 42, and the head-and-shoulders icon will both take you to your profile home page.

  - Projects: The graph icon. Click here and see the views in "All Projects" and "List projects". In the "All projects" map, you will start at the bottom with the First Day project, by default start with the Parseltongue Piscine challenges,

and then pick any branch of the tree that you would like to explore. If you already know how to code, you can use "List projects" to see which other projects are available. You can register to anything that you think is the best one for you.

○ E-Learning: The movie strip icon links to a few videos and PDFs that are produced by 42 and available for reference. There should be a few that are visible to you.

○ Forum: The speech bubble icon links to a forum. It is currently mostly used by French students and not the Americans, but you are welcome to create your own HackHighSchool community there.

○ The last three, Companies, Meta, and Shop will not be very relevant for you at this time.

Chapter IV

Our Chatroom is Slack

Slack is messaging platform designed for companies and communities. It's useful because we can...

- Direct message people.

- Add Slack to smartphones so messages show up like text messages.

- Choose times of day when you are "offline" and messages are muted.

- Create public and private groups for your projects and teams.

Create an account on h2s42.slack.com using the @student.42.us.org email address that displays on your Intra page. It is an email alias that will forward to the email you signed up with. It's helpful to add Slack to your phone too and enable notifications.

> It's best if your display name is your full name, and your Slack username is the same your 42 intra username.

> Add a profile photo, whether your face or a picture you identify with!

> Fill in your bio with info about what programming language you like to use and what high school you go to.

Chapter V

Choose your first topic to study

When you come in next week, we want to match you with a mentor who is studying the same topics as you.

Please fill out this survey to tell us which topic you want to study at first!

Curriculum Placement Survey

You can swap later, but you should always sit with the group that matches the topic you're working on, so that you can all ask each other for help and share successes.

Here are the topics we offer this year:

1. `AP CS Principles`: This will be more like a school class with group discussions and assignments. We'll use p5.js, a Javascript-based graphics library, for the full class. AP CS Principles means we will cover "Intro to Programming" topics, create models of how the Internet works, build creative projects, complete a current events research assignment, and study for the multiple choice AP exam.

2. `Intro to Python`: On Intra this is called the Parseltongue Piscine, and then you can try Algorithmic Puzzles, OOP, or Game Design 1.

3. `Object Oriented Programming in Python`: Work through the "Object Oriented Python Programming" group of projects, and then "Reverse Polish Notation". Consider building a game using Python-style OOP.

4. `Data Science in Python`: Try out the four projects under "Data Mining the 49ers".

5. `Web development`: Do the Runestone class "Fundmentals of Web Programming," then the "Web Development" projects, and then the PHP Piscine. The Runestone class "Javascript for Python Programmers" may also be useful.

6. `C Programming`: Work through the C Piscine, and then consider trying the CPP Piscine. Ask if you would like access to other C projects from 42. We also have

a `Unity` piscine which can be set up on your own laptop but not on the iMac computers.

7. `AP Java`: You can self-study for the exam using the Runestone classes "Java for Python Programmers" and "Java Review for the AP CS A Exam".

Chapter VI

Recess: FTZZLE!

Eat dessert first!

Play Ftzzle before you start your chores.

Open the website ftzzle.42.us.org.

There are no instructions... click around and use logic to figure out how this works!

You can definitely ask your friends (but not the mentors, in this case) for brainstorming help.

Chapter VII

Coding Tools

Set up your programming environment by picking a program for each of three essential functions. Our favorites are listed below. If you want to install one which is not already on your computer, simply download the program and then drag its icon to your desktop or to a folder of your choice (just not Applications).

1. Web browser

   - Built-in: Safari

   - Recommended: Firefox

   - Recommended: Google Chrome

2. Terminal

   - Built-in: Terminal

   - Recommended: iTerm2

3. Text Editor

   - Built-in inside the terminal: Emacs or Vim (recommended that you learn one of these.)

   - Built-in on Mac: Xcode

   - Recommended: VSCode

   - Recommended: Atom

   - Recommended: Sublime Text

Chapter VIII

Command Line

No matter what language or topic you study, you will need to know how to use Terminal (AKA the Command Line) in order to have a really good conversation with your computer.

Read a few articles about it and then complete the exercises below!

Although you may use online editors such as repl.it, or codepad.org, you will periodically need to access the terminal to turn in your code to our system or to test other people's code. So brush up on your knowledge and refer back to cheat sheets when you need to.

VIII.1   Learn Online

Here are some great tutorials for the Command Line!

- Team TreeHouse Blog

- Django Girls

- Launch School

- Lifehacker

- Learn Enough to Be Dangerous (a more in-depth tutorial)

VIII.2   Ru's Cheat Sheet

use your tab button to auto-complete filenames! Trust me, it's super useful once you get the hang of it.

pwd: "Present Working Directory" For all the "Where am I?" moments. When you first open a terminal, you start in your Home directory, which is labeled with your username.

```
$ pwd
```

cd: aka "Change Directory". "cd .." means go up a directory. "cd" by itself means go to your Home directory. Below we are moving into a directory called Desktop.

```
$ cd Desktop/
```

mkdir: Short for "Make Directory". Creates a new empty directory.

```
$ mkdir helloTerminal
```

ls or ls -a: "list", list all the folders and files inside the current directory you are in.

```
$ ls
```

touch: create files (files will usually end with the language you are using for example ruby is 42.rb, python is 42.py, text files are 42.txt)

```
$ cd helloTerminal
$ touch 42.txt
```

vim: opens an in terminal text editor to edit your files. Once inside vim, press "i" to edit and press ":wq" to save and exit the file. Try writing "Hello World" on the top and then exiting.

```
$ vim 42.txt
```

cat: to preview the contents of the file in Terminal. It should show "Hello World".

```
$ cat 42.txt
```

cp: "copy", copies files or directories. Here, we copy the file 42.txt and place it in the HackHighSchool/ directory. Don't forget to make the HackHighSchool directory first with mkdir.

```
$ cp 42.txt HackHighSchool/
```

mv: "move", move a file into a directory, use mv with the source file as the first argument and the destination directory as the second argument. Here we move 42.txt into HackHighSchool/

```
$ mv 42.txt HackHighSchool/
```

rm or rm -rf: "rm" to delete files and "rm -rf" to delete directories.

```
$ rm -rf HackHighSchool
```

ascii banner: create an ascii art banner on your terminal

```
$ banner -w 35 COOL
```

say: Make your mac say whatever you want it to say.

```
$ say "HackHighSchool is super fun"
```

> ⚠ Be careful with `rm` and `rm -rf`, you might accidentally delete something you need!

VIII.3   The Vim Text Editor

Vim is a powerful text editor from 1991 that comes built into every terminal.

Users of `Vim` are party to a long-running fandom war against those who prefer an even older and more powerful built-in text editor, called `Emacs`.

If you learn and prefer `Emacs` here, you must do it in secret, and never reveal your alleigance to any of your mentors, your parents, or loved ones beyond a few fellow-minded peers. Enjoy your rebellion. `You have been warned.`
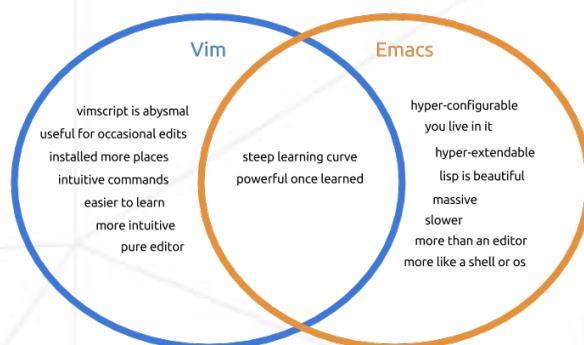
Vim basics:

- `vim <filename>` opens a file for editing, or creates one if that file doesn't exist.

- `i` toggles the editor to `insert` mode so you can write in the file.

- `ESC button` toggles the editor out of insert mode, so you can type commands instead of writing in the file.

- `:w` is a command that you can type, when not in insert mode, to `write to file` aka save your work.

- `:q` is a command that you can type, when not in insert mode, to `quit` the text editor.

- `:q!` will let you quit without saving.

- `:wq` lets you save and quit at the same time.

You can learn a lot more about how to use `Vim` by entering the command "vimtutor" in Terminal or visiting OpenVim.com.

Chapter IX

Git your files to Vogsphere

At 42 we use a version of the `git protocol` to turn in assignments online.

What is `git`?

- [Video: Codemy School](#)

- [Video: The Coding Train](#)

- [Learn Enough to be Dangerous](#)

- [Git, the Simple Guide](#)

Git is a set of terminal commands that makes up a powerful system for version control - i.e. the Google Docs of programming. It was invented to track changes so that programmers can edit their code experimentally before making those changes permanent.

We use git to send files to and from the Vogsphere, which is actually just a server (a computer) that is physically present in our server room. It is named after the planet of the Vogons, the repulsive bureaucratic overlords of Hitchhiker's Guide to the Galaxy.

You need to know the following git commands for projects at 42:

- `git clone <repo-url> <newfolder-name>`: Makes a copy of a folder from the Vogsphere (repo-url) onto your local computer, naming it <newfolder-name>.

- `git add <files>`: The files named will be included on the next commit.

- `git commit -m "message"`: Commit your changes on files that have been added. I.e., you say that you are happy with your changes on these files and are ready to save your progress. The message is for keeping notes on what was changed.

- `git push`: Pushes your files to the main server. Sometimes you add options to this command to be specific about where you are pushing to and from.

- `git pull`: If your teammate pushed changes to the server, this pulls the changes to update your own version of the files.

Chapter X

Today's Assignment

H2S	

Topics to study : `Terminal Commands, Git Commands, & Vim`
Files to turn in : `42.txt`
Notes : `n/a`

X.1 Pull from Vogsphere

1. First things first: Press "command + space" to bring up a search bar. Type "iterm2" and press "enter" to open the terminal.

2. From your Hello 42! Hello Terminal! project page on intra, copy the "Git Repository" link. (The one that looks like vogsphere@vgs.42.us.org:intra/2018...). If you do not have a link yet, make sure you are registered to the project, and wait about 5 minutes while refreshing the page.

3. Now, in the terminal type "git clone <copied link> (space) <newfoldername>". Replace <link> with your pasted link and <newfoldername> with Hello42, as a name for the project folder.

4. cd into the newly created folder. Everything placed inside here can be uploaded to Vogsphere.

5. Complete the project requirements (see next page). All files for the project should go in the folder we just created.

If you have an error during the git clone type "kinit <username>" and press enter. Then, type your intra password.

X.2   Use Terminal

- Inside your project folder (the one cloned from Vogsphere), create an empty directory called "ex00".

- Create a hidden text file (.shh42.txt) in that directory.

- Inside the hidden text file, write a list of your top 4 favorite websites.

- Create a text file named hello42.txt.

- Inside your text file, write notes on the terminal commands you learned and what they do.

- Create a new directory inside "ex00" called Hello42.

- Move your text file (hello42.txt) into the new directory.

- Go back to your home directory with the command `cd`.

- Find your way back to your "ex00" directory.

- Copy one of your text files onto your Desktop and watch it pop up on the desktop of your computer.

When done, use the next instructions to turn in your work.

X.3   Push to Vogsphere

To turn in your work, first navigate to the main level of your project folder.

If you type `ls -a`, you should see an item called ".git" which means this is your git-tracked folder.

Turn in your work by typing three commands in order:

- git add *

- git commit -m "Today I worked on ..<your comments here>.."

- git push

> 💡 Make sure you are in your project's git folder when you type the git add, git commit, git push commands.

> ⚠️ If you have an error during the git push, you may need to refresh your authentication ticket. Do this by typing "kinit <username>" and then typing your intra password.

Back on Intra, you can press the button "Set the Project as Finished" to indicate that you are ready for corrections.

X.4 Peer Correction

- On your project page, click the button "Subscribe to Defense." Choose a time slot from the screen that shows next.

- When the appointment time comes, look up your assigned corrector on Slack and send them a message. Find each other.

- Corrector, sit at the correctee's station and open a new web browser in Incognito mode. Log into your intra.

- Access the corrections page. Clone their Git repository into a new folder.

- Click "Begin correction" and follow the instructions on the correction page.

- Use this time to chat about the project and the rest of your life.

- Once the correction is finished, your corrector should remember to log out of Intra.

- After both the correction is complete, in order to finalize your score you must then provide feedback for the corrector on your project page. Go to your project page and click the "feedback" button for each completed correction.

Chapter XI

What's next?

Play more ftzzle.42.us.org.

Help the people around you finish, and then celebrate when all are done!

Learn how to use `Vim` by entering the command "vimtutor" in Terminal or visiting OpenVim.com.