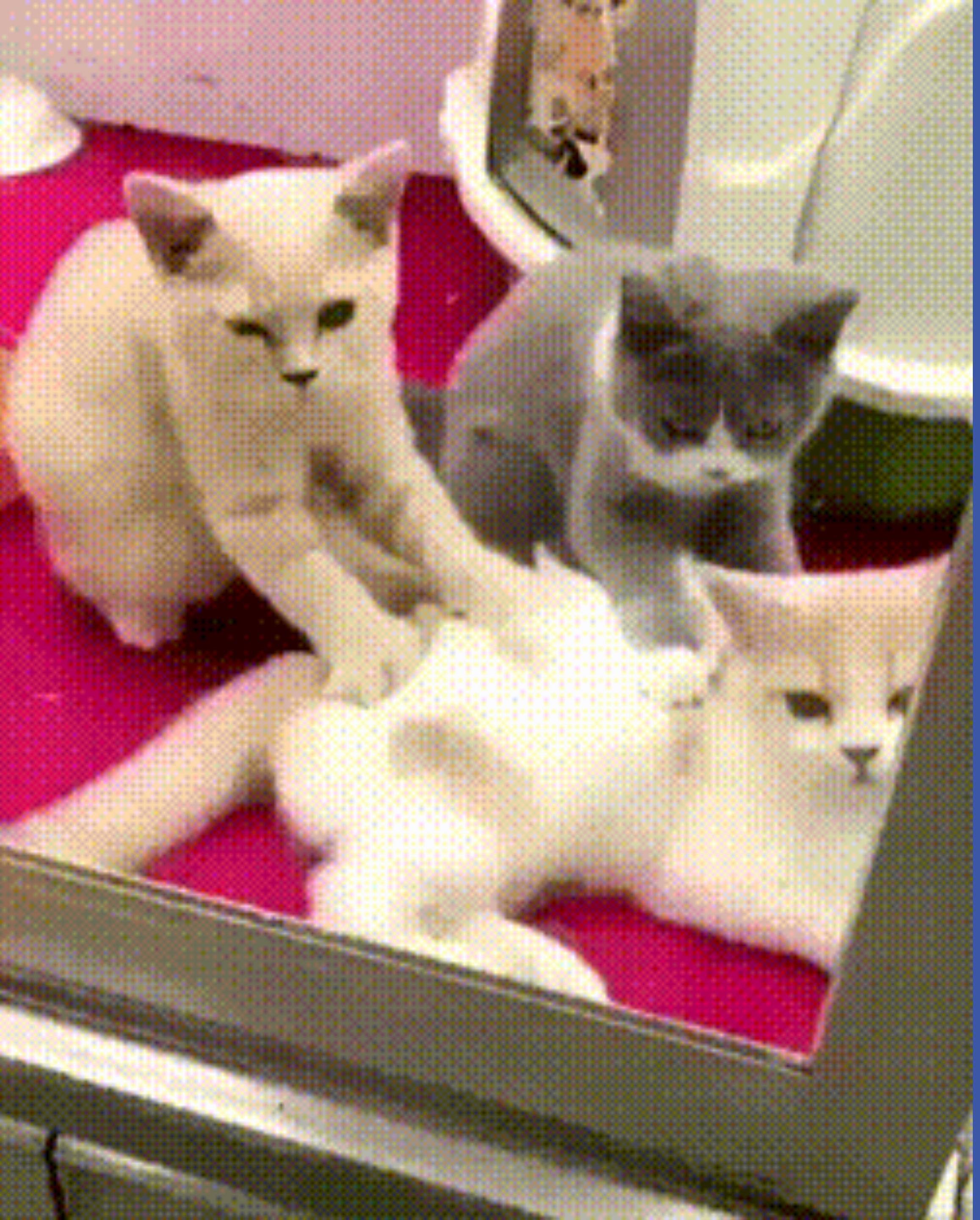


DATA SCIENCE FUNDAMENTALS

WEEK 1, LESSON 2

Hay Kranen
Wednesday September 26th 2018



TODAY'S PROGRAMME

Recap

Break

Git + the terminal

Break

Comparisons and the if statement

String methods

Exercise

Lunch break

RECAP

```
amount = input("How many oreo's did you eat?")
```

```
amount = input("How many oreo's did you eat?")
^
```

```
SyntaxError: invalid character in identifier
```

Comments

```
# Brinky Nutritional Value #
kcal = 115.2
carbs = 16.6
fat = 4.8
eggwhite = 1.3
```

```
In [1]: fat = float(16.9)
```

```
In [2]: carbs = int(72)
```

```
In [3]: protein = float(3.8)
```

Type	Example	Conversion function
Integer	42, -5903, 0	int()
Float	3.14, -55.24, 2.0	float()
String	"Hello, World"	str()
Boolean	True, False	bool()

Python dynamic typing

```
name = "Oreo cookie"      # string
calories = 115            # int
carbs = 16.6              # float
is_snack = True           # bool
```

Java static typing

```
String name = "Oreo cookie";
int calories = 115;
float carbs = 16.6;
boolean is_snack = true;
```

```
calories = input("hoeveel calorieën heeft het?")  
fat = input("hoeveel vet heeft het?")  
carbs = input("hoeveel carbs heeft het?")  
sugar = input("hoeveel suiker heeft het?")  
  
kitkat = calories + fat + carbs + sugar  
  
kitkat = input("hoeveel kitkats heb je gegeten?")
```

Cal = 160

Fat = 1.5

Carb = 37

Sugar = 30

```
1 calories = 50
2 fat = 2
3 carbs = 8
4 sugar = 4
5
6 cookies = input("How many cookies did you eat?")
7
8 print("calories = " + str((cookies * calories)) + " fat = " +
      str((cookies * fat)) + " carbs = " + str((cookies *
carbs)) + " sugar = " + str((cookies * sugar)))
9 .
```

```
print("You've eaten " + str(int(avocados) * calories) + " Calories" )
print("You've eaten " + str(int(avocados) * carbohydrate) + " carbohydrates" )
print("You've eaten " + str(int(avocados) * fat) + " fat" )
```

```
1 calories = 50
2 fat = 2
3 carbs = 8
4 sugar = 4
5
6 cookies = input("How many cookies did you eat?")
7
8 print("calories = " + str(cookies * calories)) + " fat = " +
      str(cookies * fat)) + " carbs = " + str(cookies *
      carbs)) + " sugar = " + str(cookies * sugar))
9 .   cal_value = str(cal_value)
      fat_value = str(fat_value)
      carb_value = str(carb_value)
      sugar_value = str(sugar_value)
      portions = str(portions)

      print('You ate ' + cal_value + ' calories.')
      print('You ate ' + fat_value + ' grams of fat.')
      print('You ate ' + carb_value + ' grams of carbs.')
      print('You ate ' + sugar_value + ' grams of sugar.'
```

STYLE

python.org

Python PSF Docs PyPI Jobs Community

TM

python

About Downloads Documentation Community Success Stories News Events

Tweets by @ThePSF

Python Software @ThePSF

SoCal Python Meetup Sep 25 - Marina del Rey, CA. Talks include: Python and IOTA cryptoledger and Faster Autocomplete: From Elasticsearch to pure Python! [meetup.com/socalpython/ev...](https://meetup.com/socalpython/events/)

September SoCal Python me... This September meetup will be... meetup.com

PEP 8 -- Style Guide for Python Code

PEP: 8

Title: Style Guide for Python Code

Author: Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>

Status: Active

Type: Process

Created: 05-Jul-2001

Post-: 05-Jul-2001, 01-Aug-2013

History:

Search GO Socialize

Python »» Python Developer's Guide »» PEP Index »» PEP 8 -- Style Guide for Python Code

PEP 8 -- Style Guide for Python Code

PEP: 8

Title: Style Guide for Python Code

Author: Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>

Status: Active

Type: Process

Created: 05-Jul-2001

Post-: 05-Jul-2001, 01-Aug-2013

History:

1. Limit lines to 80 characters

```
1 # Define the nutritional values here
2 calories = 50
3 fat = 2.3
4 carbs = 8.3
5 sugar = 4.7
6
7 # Ask how many cookies the user had
8 nr_of_cookies = input("How many Oreo cookies did you eat? ")
9
10 # Feedback to the user
11 print("You had " + nr_of_cookies + " cookies")
12
13 nr_of_cookies = float(nr_of_cookies)
14 user_calories = calories * nr_of_cookies
15 user_fat = fat * nr_of_cookies
16 user_carbs = carbs * nr_of_cookies
17 user_sugar = sugar * nr_of_cookies
18
19 print("Calories: " + str(user_calories))
20 print("Fat: " + str(user_fat) + " grams")
21 print("Carbs: " + str(user_carbs) + " grams")
22 print("Sugar: " + str(user_sugar) + " grams")
23
```

2. Use lowercase_with_underscores for variable names

```
1 # Define the nutritional values here
2 calories = 50
3 fat = 2.3
4 carbs = 8.3
5 sugar = 4.7
6
7 # Ask how many cookies the user had
8 nr_of_cookies = input("How many Oreo cookies did you eat? ")
9
10 # Feedback to the user
11 print("You had " + nr_of_cookies + " cookies")
12
13 nr_of_cookies = float(nr_of_cookies)
14 user_calories = calories * nr_of_cookies
15 user_fat = fat * nr_of_cookies
16 user_carbs = carbs * nr_of_cookies
17 user_sugar = sugar * nr_of_cookies
18
19 print("Calories: " + str(user_calories))
20 print("Fat: " + str(user_fat) + " grams")
21 print("Carbs: " + str(user_carbs) + " grams")
22 print("Sugar: " + str(user_sugar) + " grams")
23
```

3. Consistently use spaces and newlines

```
1 # Define the nutritional values here
2 calories = 50
3 fat = 2.3
4 carbs = 8.3
5 sugar = 4.7
6
7 # Ask how many cookies the user had
8 nr_of_cookies = input("How many Oreo cookies did you eat? ")
9
10 # Feedback to the user
11 print("You had " + nr_of_cookies + " cookies")
12
13 nr_of_cookies = float(nr_of_cookies)
14 user_calories = calories * nr_of_cookies
15 user_fat = fat * nr_of_cookies
16 user_carbs = carbs * nr_of_cookies
17 user_sugar = sugar * nr_of_cookies
18
19 print("Calories: " + str(user_calories))
20 print("Fat: " + str(user_fat) + " grams")
21 print("Carbs: " + str(user_carbs) + " grams")
22 print("Sugar: " + str(user_sugar) + " grams")
23
```

4. Use comments to describe what you're doing

```
1 # Define the nutritional values here
2 calories = 350
3 fat = 2.3
4 carbs = 8.3
5 sugar = 4.7
6
7 # Ask how many cookies the user had
8 nr_of_cookies = input("How many cookies did you eat? ")
9
10 # Feedback to the user
11 print("You ate " + str(nr_of_cookies) + " cookies")
12
13 nr_of_cookies = float(nr_of_cookies)
14 user_calories = calories * nr_of_cookies
15 user_fat = fat * nr_of_cookies
16 user_carbs = carbs * nr_of_cookies
17 user_sugar = sugar * nr_of_cookies
18
19 print("Calories: " + str(user_calories))
20 print("Fat: " + str(user_fat) + " grams")
21 print("Carbs: " + str(user_carbs) + " grams")
22 print("Sugar: " + str(user_sugar) + " grams")
23
```

Comments

```
# Brinky Nutritional Value #
kcal = 115.2
carbs = 16.6
fat = 4.8
eggwhite = 1.3
```

4. Use comments to describe what you're doing

```
1 # Define the nutritional values here
2 calories = 50
3 fat = 2.3
4 carbs = 8.3
5 sugar = 4.7
6
7 # Ask how many cookies the user had
8 nr_of_cookies = input("How many Oreo cookies did you eat? ")
9
10 # Feedback to the user
11 print("You had " + nr_of_cookies + " cookies")
12
13 nr_of_cookies = float(nr_of_cookies)
14 user_calories = calories * nr_of_cookies
15 user_fat = fat * nr_of_cookies
16 user_carbs = carbs * nr_of_cookies
17 user_sugar = sugar * nr_of_cookies
18
19 print("Calories: " + str(user_calories))
20 print("Fat: " + str(user_fat) + " grams")
21 print("Carbs: " + str(user_carbs) + " grams")
22 print("Sugar: " + str(user_sugar) + " grams")
23
```

```
# Nutritional values for one Oreo cookie
# Source: https://www.fitbit.com/foods/Oreo+Cookie/16421
calories = 50
fat = 2.3
carbs = 8.3
sugar = 4.7
nr_of_cookies = input("How many Oreo cookies did you eat? ")
print("You had " + nr_of_cookies + " cookies")
# Convert to float, because the user might have had part of a cookie
nr_of_cookies = float(nr_of_cookies)
# And calculate the user values
user_calories = calories * nr_of_cookies
user_fat = fat * nr_of_cookies
user_carbs = carbs * nr_of_cookies
user_sugar = sugar * nr_of_cookies
print("Calories: " + str(user_calories))
print("Fat: " + str(user_fat) + " grams")
print("Carbs: " + str(user_carbs) + " grams")
print("Sugar: " + str(user_sugar) + " grams")
```



TABLE OF CONTENTS

1. Introduction
2. Main objective
3. Learning objectives
4. Assignments
 - 4.1. Registration
 - 4.2. Storing
 - 4.3. API
 - 4.4. Dataset
 - 4.5. Scraping
5. Contents and schedule
6. What you need for this course
7. The book
- 8. Programming tips and references**
9. Assessment
10. Installing Anaconda

Student's manual - READ THIS FIRST



8. Programming tips and references

References

- Extensive Python cheatsheet with examples
- A more minimal cheatsheet
- Hay's cheatsheet
- Datacamp Python basics
- A summary of the Python coding style
- String methods and functions
- Datacamp introduction to lists
- Working with JSON web API's
- Installing Git on Mac, Windows and Linux

- Google and [Stack Overflow](#) are your friends. It's not a shame to Google even really basic concepts. I have been programming for more than twenty years and i still Google really basic stuff every single day.
- Your code should be properly commented (use #). Good commenting means you explain *why* you do something, not *what* you're doing.
- Keep your code DRY: Do not Repeat Yourself. If you copy-paste code, you probably could use a function instead.

hay / hu-dsf

Watch 3 Star 4 Fork 2

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master → hu-dsf / cheatsheet.md Find file Copy path

hay Adding cheatsheet 14e4177 27 seconds ago
1 contributor

34 lines (31 sloc) | 1.46 KB Raw Blame History

Data science fundamentals cheatsheet

Terminal commands

command	use	example
ls	List files in current directory	ls
	List files as a list	ls -l
cd	Change to directory	cd Desktop
	Up one level in directory structure	cd ..
mkdir	Make a directory	md pythoncode
pwd	Print working directory	pwd
touch	Create a new empty file	touch test.py

GIT AND THE TERMINAL

**** COMMODORE 64 BASIC V2 ****
64K RAM SYSTEM 38911 BASIC BYTES FREE
READY.





A terminal window titled "hay — hay@haybook: ~ — -bash — 112x33". The window shows a black terminal screen with a white border. At the top left are three colored window control buttons (red, yellow, green). The title bar contains the text "hay — hay@haybook: ~ — -bash — 112x33". Below the title bar, the terminal displays the text "Last login: Tue Sep 25 12:52:35 on ttys003" in green, followed by "hay@haybook:~ \$". The rest of the terminal screen is blank black space.



Source: Peter Hamer / CC-BY-SA / [https://commons.wikimedia.org/wiki/File:Ken_Thompson_\(sitting\)_and_Dennis_Ritchie_at_PDP-11_\(2876612463\).jpg](https://commons.wikimedia.org/wiki/File:Ken_Thompson_(sitting)_and_Dennis_Ritchie_at_PDP-11_(2876612463).jpg)

hay — hay@haybook: ~ — -bash — 72x23

Last login: Tue Sep 25 16:04:07 on ttys004

hay@haybook:~ \$ Why do we still use a terminal?█

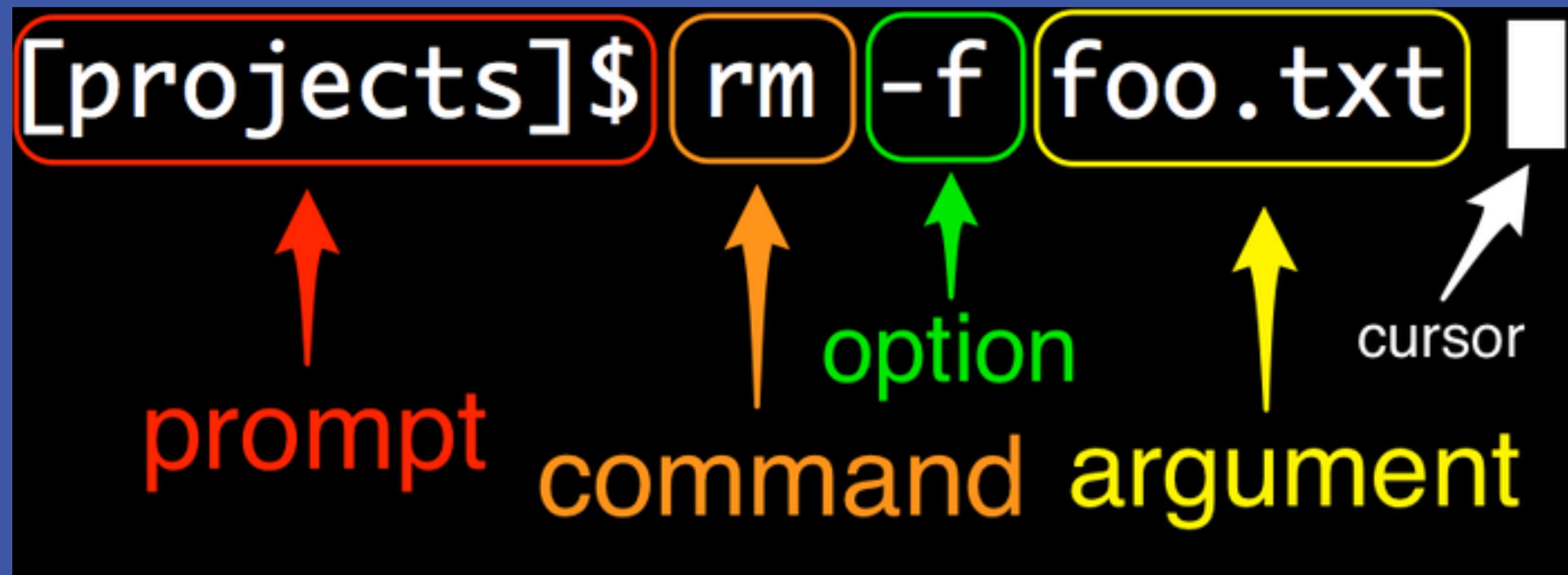
Some practical examples

- * Resize a 1000 jpeg files
- * Download YouTube videos
- * Backup a server every night
- * Convert between data formats



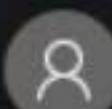
VS.







Recycle Bin



BernardoTech

Life at a glance

Play and explore

Most used

Get Started

Get Skype

Maps

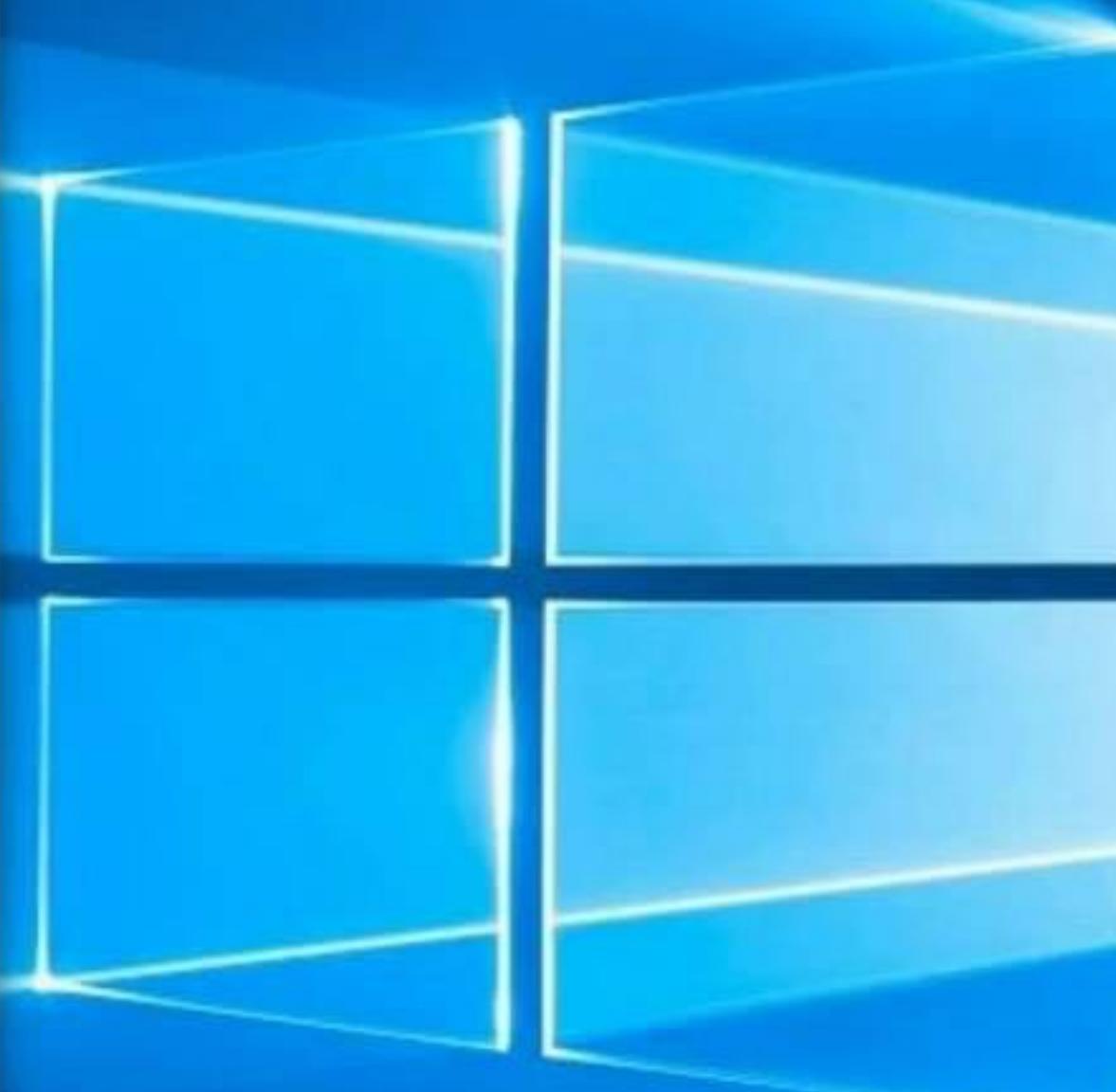
People

Calculator

Alarms & Clock

Recently added

OneDrive



File Explorer

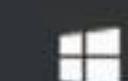


Settings

Power

All apps

New



Search the web and Windows



5:01 PM

3/3/2016



Terminal commands

command	use	example
<code>ls</code>	List files in current directory	<code>ls</code>
	List files as a list	<code>ls -l</code>
<code>cd</code>	Change to directory	<code>cd Desktop</code>
	Up one level in directory structure	<code>cd ..</code>
<code>mkdir</code>	Make a directory	<code>md pythoncode</code>
<code>pwd</code>	Print working directory	<code>pwd</code>
<code>touch</code>	Create a new empty file	<code>touch test.py</code>
<code>cat</code>	Print the contents of a file	<code>cat test.py</code>
<code>cp</code>	Copy one or more files	<code>cp test.py test2.py</code>
<code>mv</code>	Rename or move a file	<code>mv test2.py test3.py</code>
<code>rm</code>	Delete a file	<code>rm test3.py</code>
	Delete a directory	<code>rm -r pythoncode</code>

Terminal tips

- Try the TAB key to autocomplete your commands
- Use up and down arrow keys for previous / next commands
- Use CTRL-C to stop something

Basic

Convert your MyFitnessPal application to a .py textfile

Save it in a directory

Start a terminal and navigate to that directory

Try running the application using the python command

Advanced

Create a new directory and move to that directory

Create a new file called name.py

Use nano to open that file

Write a simple Python program that asks your name and prints it

Exit nano and run the program

Text editors

Windows: Notepad++, Programmer's Notepad, Atom

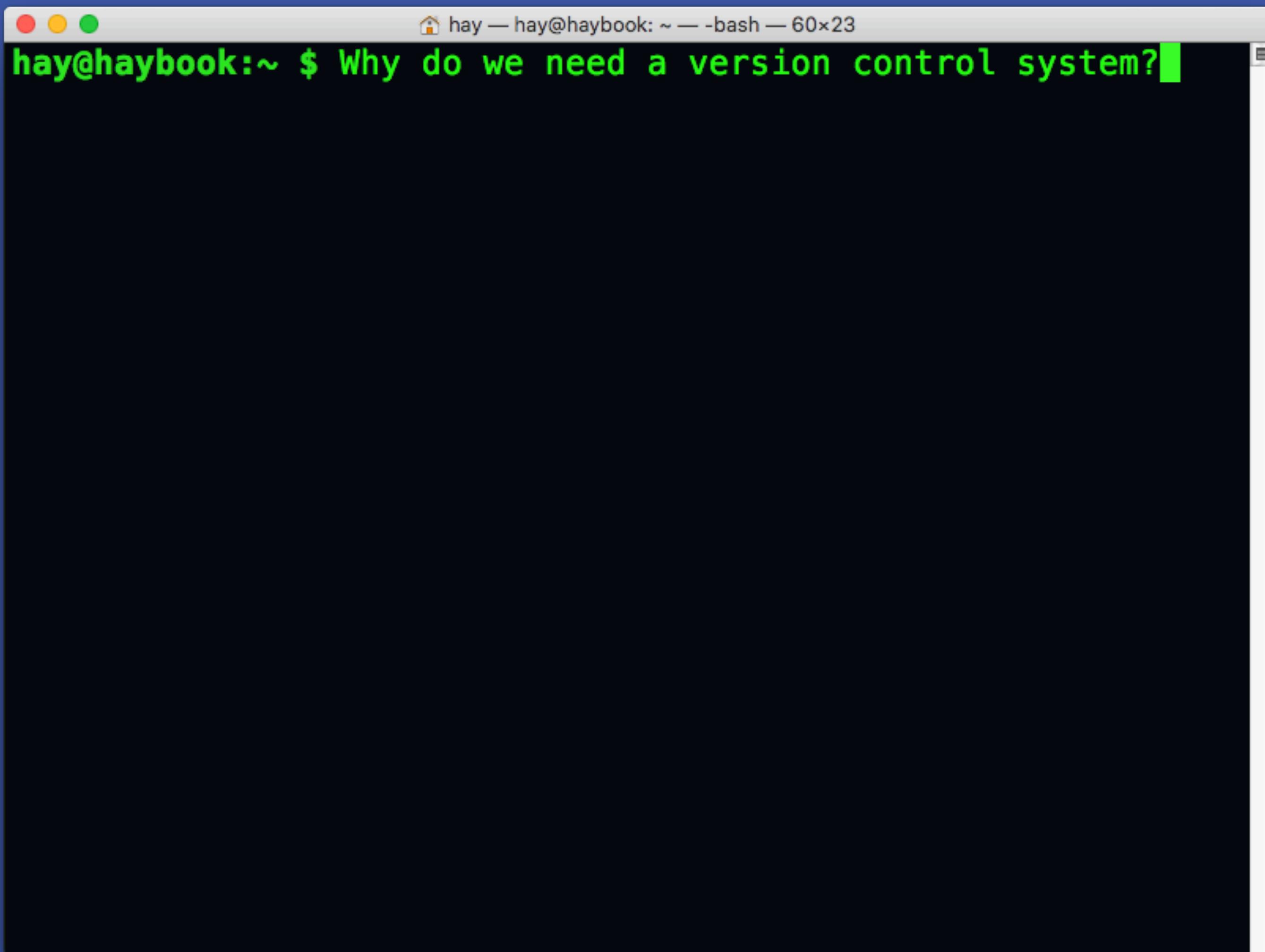
Mac: Sublime Text, Atom, BBedit



git

Version control system

Version control systems are a category of software tools that help a software team manage changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database.



hay — hay@haybook: ~ — -bash — 60x23

hay@haybook:~ \$ Why do we need a version control system?



Github



git

INTERNATIONAAL

It's official: Microsoft will spend a whopping \$7.5 billion to buy GitHub, a startup at the center of the software world

Matt Weinberger, Ben Gilbert ⌚ 04 Jun 2018 ↳ 170

Git commands

command	use	example
git	Prints most used git commands	git
git init	Create a new repository	git init
git add	Add a file to the repository	git add test.py
	Add all files in directory to repository	git add .
git commit	Commit changes to repository	git commit -m "Your message here"
git clone	Create a working copy of a repository	git clone git@github.com:hay/hu-dsf.git
git status	List changes in your working directory	git status
git log	List what happened in your repository	git log

```
git commit -m "Finally fixing the awful coffee machines at the HU"
```

git commit



Source: Peter Hamer / CC-BY-SA / [https://commons.wikimedia.org/wiki/File:Ken_Thompson_\(sitting\)_and_Dennis_Ritchie_at_PDP-11_\(2876612463\).jpg](https://commons.wikimedia.org/wiki/File:Ken_Thompson_(sitting)_and_Dennis_Ritchie_at_PDP-11_(2876612463).jpg)

Just memorize these fourteen contextually dependant instructions

Stack Overflow: Helping One Man Escape Vim



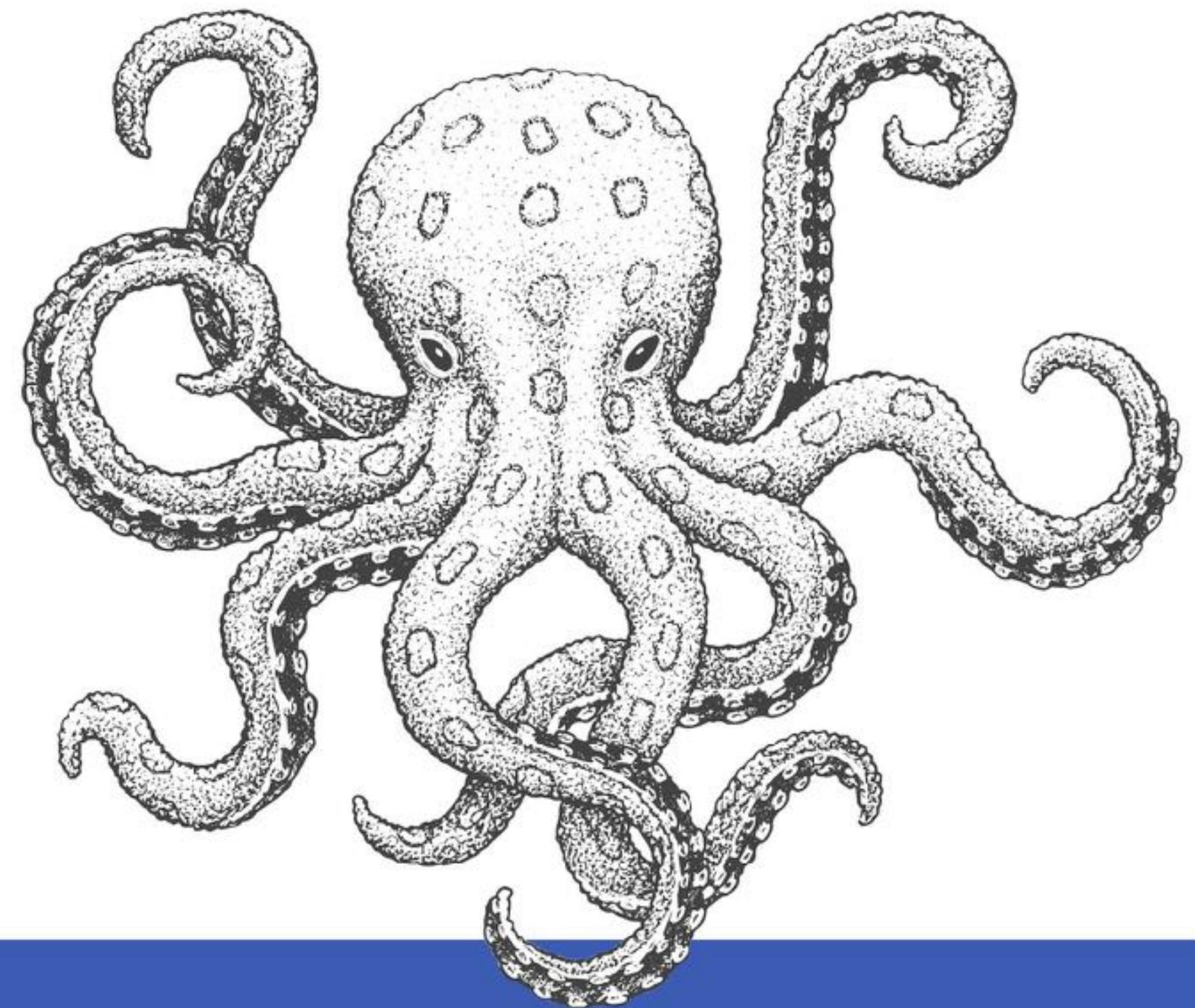
by David Robinson on May 23, 2017

How many people have been struggling to exit Vim?

In the last year, [How to exit the Vim editor](#) has made up about every 20,000 visits to Stack Overflow questions. That means there are about 80 people per hour that need help getting out of Vim.

I'm stuck and cannot escape. It says:
"type :quit<Enter> to quit VIM"
But when I type that it simply appears in the object body.

vim vi
362 share improve this question
asked Aug 6 '12 at 19:45
Peter Mortensen 11.1k ● 16 ● 76 ● 109
edited Nov 3 '16 at 20:26
jcclancy 6,784 ● 5 ● 10 ● 11



Exiting Vim

Eventually

O RLY?

@ThePracticalDev

Basic

Create a Github account

Go to the DSF repo and 'fork' the repository

Add your name and link to your account to the students.md file

Create a new pull request

Advanced

Create a new repository for your Python code

Add your MyFitnessPal example to the repository

Commit to your repo

COMPARISONS AND THE IF STATEMENT

Comparison: a statement that
returns a bool, so it's either True
or False

Operator	Description	Example
<code>==</code>	Is the same	<code>42 == 42</code>
<code>!=</code>	Is not the same	<code>"Tinus" != "Barrie"</code>
<code><</code>	Is lesser than	<code>30 < 40</code>
<code><=</code>	Is lesser or equals than	<code>30 <= 30</code>
<code>></code>	Is greater than	<code>40 > 30</code>
<code>>=</code>	Is greater or equals than	<code>40 >= 40</code>
<code>in</code>	String is in other string	<code>"Jan" in "Jantje"</code>
<code>not in</code>	String is not in other string	<code>"Piet" not in "Jantje"</code>
<code>and</code>	Two comparisons are both true	<code>("Jan" in "Jantje") and ("Piet" in "Pietje")</code>
<code>or</code>	One of either is true	<code>("Jan" in "Jantje") or ("Piet" in "Jantje")</code>

What happens when you run this program?

```
age = 20

if age < 20:
    print("option 1")
elif age <= 20 and age > 20:
    print("option 2")
else:
    print("option 3")
```

- a** It prints "option 1"
- b** It prints "option 2"
- c** It prints "option 3"

Basic

Create a program that takes two numbers, compares them, and prints if the first number is greater, lesser or equal to the second.

Advanced

Change the program so that it takes two strings and prints if the first string is a part of the second string **or** has the same length as the second string.

STRING METHODS

Common string methods

Method	Description	Example
<code>lower()</code>	Lowercases the string	<code>"HELLO".lower()</code>
<code>upper()</code>	Uppercases the string	<code>"hello".upper()</code>
<code>replace()</code>	Searches for a string in a string and replaces that	<code>"I like pie".replace("pie", "cookies")</code>
<code>find()</code>	Gives the position of a string in another string	<code>"Jantje".find("je")</code>
<code>strip()</code>	Removes all whitespace before the beginning and end of a string	<code>" hello ".strip()</code>
<code>in</code>	Not a method, but an operator to check if a string is in another string	<code>"Jan" in "Jantje"</code>

What happens when you run this program?

```
name1 = "Tinus"  
name2 = "Barrie"  
friends = name1 + " " + name2  
friends.replace("Tinus", "Hans")  
print(friends)
```

- a** It prints “Tinus Barrie”
- b** It prints "Hans Barrie"
- c** It prints “Tinus Tinus”

Build your own love test



Basic

Write a program that asks for two names, then lowercases and strips those names and compares them. Have different outcomes when the names are equal (==), greater (>) or lesser (<).

Advanced

Calculate a 'similarity percentage' based on multiple criteria, such as:

- * Length of names
- * Similarity of characters in the names
- * If the combination is popular
- * Whatever you can think of yourself

When you're finished

Post your code to Slack or commit to Github and share the link