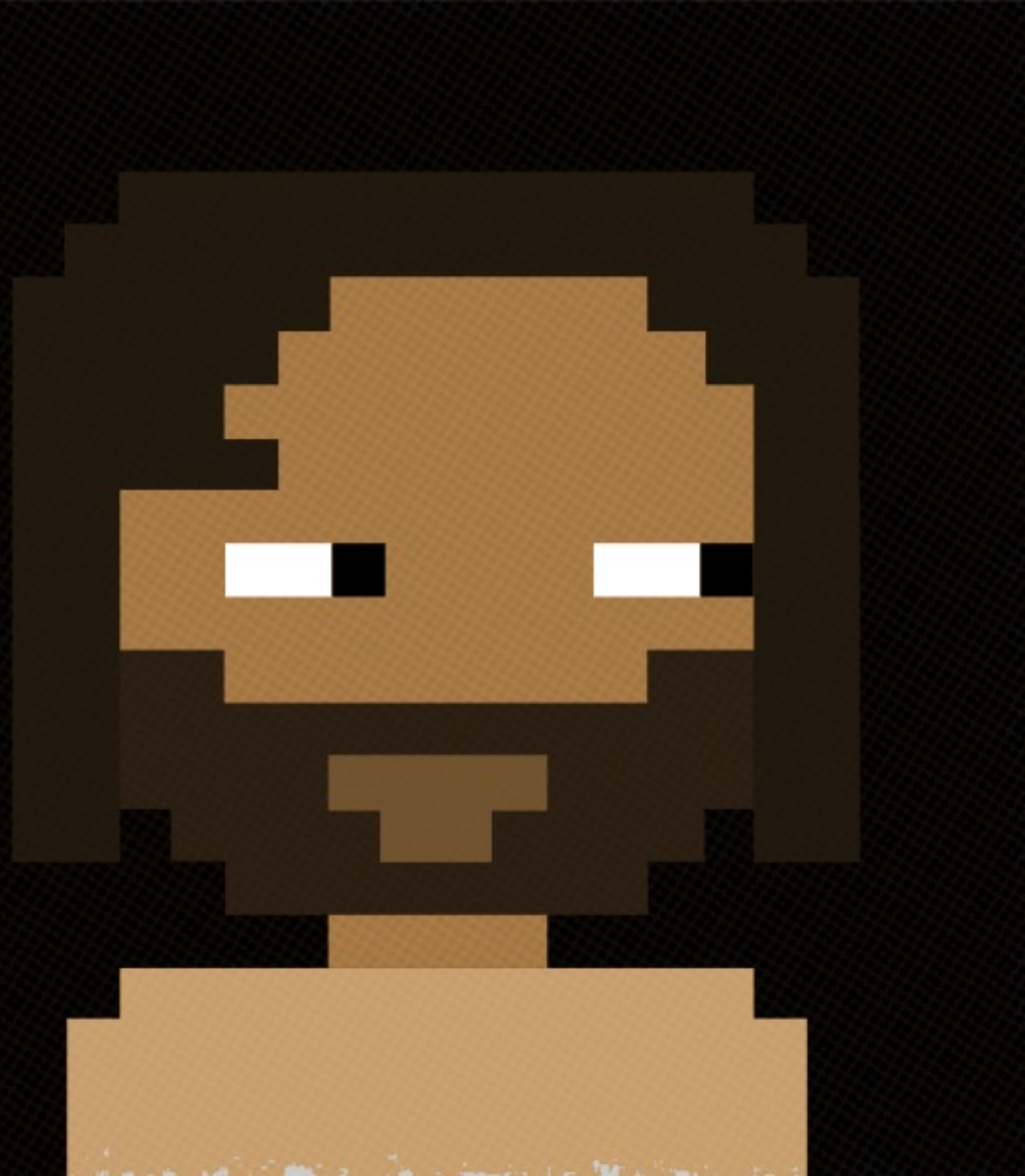
event	the black is tech conference 2023
session	command line for control
presenter	jack a. lester
date	10 august 2023
time	16:30 - 17:45
location	software engineering stage g ballroom 2 georgia world congress center



[./welcomemsg.sh]

the computer is an anatomy.

your terminal is the communication,
modification, exploration,
&& creation interface.

using a terminal will affect your
machine's structure && expression.

[whoami]

/// [users] also works here.
/// [whoami] is a command which displays your username.
/// try [who] to get a more verbose output containing the date.

vs;wr (very short; will read)

```
echo "
    [+] into custom keyboard caps.
    [+] a human religious about design.
    [+] prototyping experiences at YOO.
    [+] my first tech talk was on the iphone3g.
    [+] intro to the terminal? password resets.
    [+] currently building a query engine at neo&&.
    [+] core team member at the development lab, W3BBIE."
```

command + spacebar
then type "terminal"

/// on linux: terminal is an icon in top bar.
/// on a mac? try the above on your keyboard to launch a spotlight search.
/// if on windows: look for PowerShell. windows is non-UNIX, so commands will be different.

how i would describe a terminal to pre-terminal me?

```
echo "
    [+] also called a shell, command-line.
    [+] you can do all computing from here.
    [+] the web 1.0 of computer interfaces.
    [+] a design product of bell labs → at&t.
    [+] seen in the matrix, films depicting hacking."
```

my reasons for using a terminal:

```
echo "
    [+] rapid self-testing.
    [+] less overall overhead.
    [+] things get done faster.
    [+] filesystem hygiene improves.
    [+] first-party content can be cultivated.
    [+] provides deeper access to computer as a utility."
```

after this demo, i want us to:

```
echo "
    [+] grasp that everything is a file.
    [+] feel more confident using the terminal.
    [+] be familiar with several basic commands.
    [+] build momentum for expanding today's content."
```

conventions used throughout demo:

```
echo "
    # indicates a comment
    /// used for sub-text
    [] placeholder for a commands
    <> placeholder for command-line arguments
    > indicates that we need to type something "
```

(enter && project)

[mkdir] <commandlinedemo>

/// we will call [mkdir] a command, but its technically a program.

```
# our working directory is where scripts will exist.
# remember: open a terminal before proceeding.
# below is the pseudocode to what we'll be doing.
# ----------------------------------------------
> change into root directory (home)
> print a few environment variables
> clear the screen
> create a new directory with a sub-directory
> change into created directory
> verify our location
> update configuration file
> view our directory tree
```

```
bash-5.2$ echo "
User: ${USER}
Home: ${HOME}
Shell: ${SHELL}
Text editor: ${EDITOR}"

User: xyz
Home: /Users/xyz
Shell: /bin/zsh
Text Editor: /usr/bin/nano
bash-5.2$ clear
```

/// writing echo as, 'echo " ' allows multi-line typing.
/// try the [printenv] command to view all of your environment variable.
/// also: we'll constantly use variables; they hold data, make writing/reading code easier.

```
bash-5.2$ OG=~/blackistech2023; cd ${OG}
bash-5.2$ DEMO_DIR=commandlinedemo; SUB_DIR=scripts
bash-5.2$ mkdir -p ${DEMO_DIR}/${SUB_DIR}
bash-5.2$ pwd
/Users/xyz/blackistech2023/commandlinedemo
bash-5.2$ ls -hrtla
total 0
drwxr-xr-x    2 xyz    staff        64B Aug 7 11:03 scripts
bash-5.2$ nano ~/.bashrc
bash-5.2$ exec $SHELL
```

```
/// ls -l lists files in long format.
/// ls -r reverses the order of the sort.
/// ls -h reduces number of digits in size of file.
/// ls -t sorts in descending order (most recently modified first).
/// ls -a lists hidden files
/// mkdir -p creates intermediate directories as required
```

```
GNU nano 7.2
1 export PATH_TO_DEMO=~/blackistech2023/commandlinedemo
2 export PATH_TO_DEMO_SCRIPTS=~/blackistech2023/commandlinedemo/scripts
3 export CONFIG=~/.bashrc
4
5 # ls alias
6 alias ls="ls -hrtla"
7
8 # Welcome message
9 alias welcomemsg=${PATH_TO_DEMO_SCRIPTS}/welcomemsg.sh
```

/// the welcomemsg is a default ☺
/// to save: ctrl + o; to exit: ctrl + x
/// exporting the values from the config allow us to use them on the terminal.
/// your config file might be named any of these: .zprofile, .profile, .bashrc, .config

```
bash-5.2$ tree

.
|--- scripts
     |--- welcomemsg.sh
|--- welcomemsg_Cellos.aiff
2 directories, 2 files
```

/// our tree is like finder's gui, possibly better ☺
/// the root of the tree (.) is being counted

VAR=	variable name
rm	removes a file
cd	change directory
ctrl + x	exit nano editor
${}	variable construct
echo	write argument to output
ctrl + o	save file in nano editor
tree	displays a directory tree
&&	AND operator, allows command chaining
exec	execute commands, replace current shell
~	tilde. takes you to root directory (home)
pwd	present working directory(current location)
-p	allows creation of intermediate directories
;	allows for multiple commands on a single line
mkdir -p	create a directory and its intermediate directories
ls -hrtla	list reduced digits, reverse sort, descending order, longformat, hidden files

[+] do you see how (over time)
this is much faster than
manually creating a folder?

[+] do you see the structured natured
of working from the terminal
as a limitation or a strength?

[voicenote] <note> <voice>

```
# we will build a spartan text-to-voice-note script.
# ---------------------------------------------------
> from commandlinedemo, change into scripts
> create a temporary variable
> create a new file
> open a text editor
> write the script in the text editor
> create an alias
> test out script
> view tree
```

```
bash-5.2$ cd ${PATH_TO_DEMO} && mkdir voice-notes
bash-5.2$ cd ${PATH_TO_DEMO_SCRIPTS}
bash-5.2$ SCRIPT=voicenote.sh
bash-5.2$ NOTE="do harmony && innovation."
bash-5.2$ VOICE="Cellos"
bash-5.2$ touch ${SCRIPT}; chmod u+x ${SCRIPT}
bash-5.2$ nano ${CONFIG}
bash-5.2$ source ${CONFIG}
bash-5.2$ nano ${SCRIPT}
```

/// heads up: your config file might be something else; .profile, .zshrc, .bash_profile, …

```
 GNU nano 7.2
 1 export PATH_TO_DEMO=~/blackistech2023/commandlinedemo
 2 export PATH_TO_DEMO_SCRIPTS=~/blackistech2023/commandlinedemo/scripts
 3 export CONFIG=~/.bashrc
 4
 5 # ls alias
 6 alias ls="ls -hrtla"
 7
 8 # Welcome message
 9 alias welcomemsg=${PATH_TO_DEMO_SCRIPTS}/welcomemsg.sh
 10
 11 # Voicenote
 12 alias voicenote=${PATH_TO_DEMO_SCRIPTS}/voicenote.sh
```

/// to save: ctrl + o
/// to exit: ctrl + x

GNU nano 7.2

```
1  #!/usr/bin/env bash
2  # @usage voicenote.sh <note> <voice-name> <alias>
3  clear
4  NOTE=${1}; VOICE=${2}; ALIAS=${3}
5  DATE=`date "+%Y%m%d%H%M%S"`
6  FILE_NAME="voicenote_${DATE}_${ALIAS}.aiff"
7  VOICE_NOTES_FOLDER=~/blackistech2023/commandlinedemo/voice-notes
8  echo "*** New Note: ${1} ***"
9  if [[ ! -d ${VOICE_NOTES_FOLDER} ]];then
10     echo "${VOICE_NOTES_FOLDER} does not exist. Exiting.";exit 1
11 else
12     say -o "${VOICE_NOTES_FOLDER}/${DATE}_voicenote_${ALIAS}.aiff"
           -v ${VOICE} ${NOTE};exit 0
13 fi
14 # backticks `` allow embedding a command into a string. see line 5
```

```
bash-5.2$ voicenote ${NOTE} ${VOICE} "testnote"
bash-5.2$ *** new note: do harmony && innovation ***
bash-5.2$ cd ${PATH_TO_DEMO} && tree
```

```
/// harmony && innovation t-shirt is available ☺
/// try the command [ say -v "?" ] for a list of all the system's available voices.
```

```
bash-5.2$ tree
.
|--- scripts
     |--- voicenote.sh
     |--- welcomemsg.sh
|--- voice-notes
     |--- 20230809231806_voicenote_testnote.aiff
|--- welcomemsg_Cellos.aiff

3 directories, 4 files
```

./${SCRIPT}	run the script
VAR=	variable creation
chmod u+x	execute permission
${}	variable construct
cd ~/<path>	change into directory
nano	opens nano text editor
open .	opens directory in system gui
say -o -v	say to output file in a voice
OUTPUT_FILE	variable for output audio file
date "+%Y%m%d%H%M%S"	current system's date, formatted
#!/usr/bin/env bash	commands in script executable(actionable)
tree	tree structure of directory and sub-layers

[+] could i input a .txt file
as an argument instead
of a manual string?

[+] would you use different voices
for various types of notes?

[deskon/deskoff]

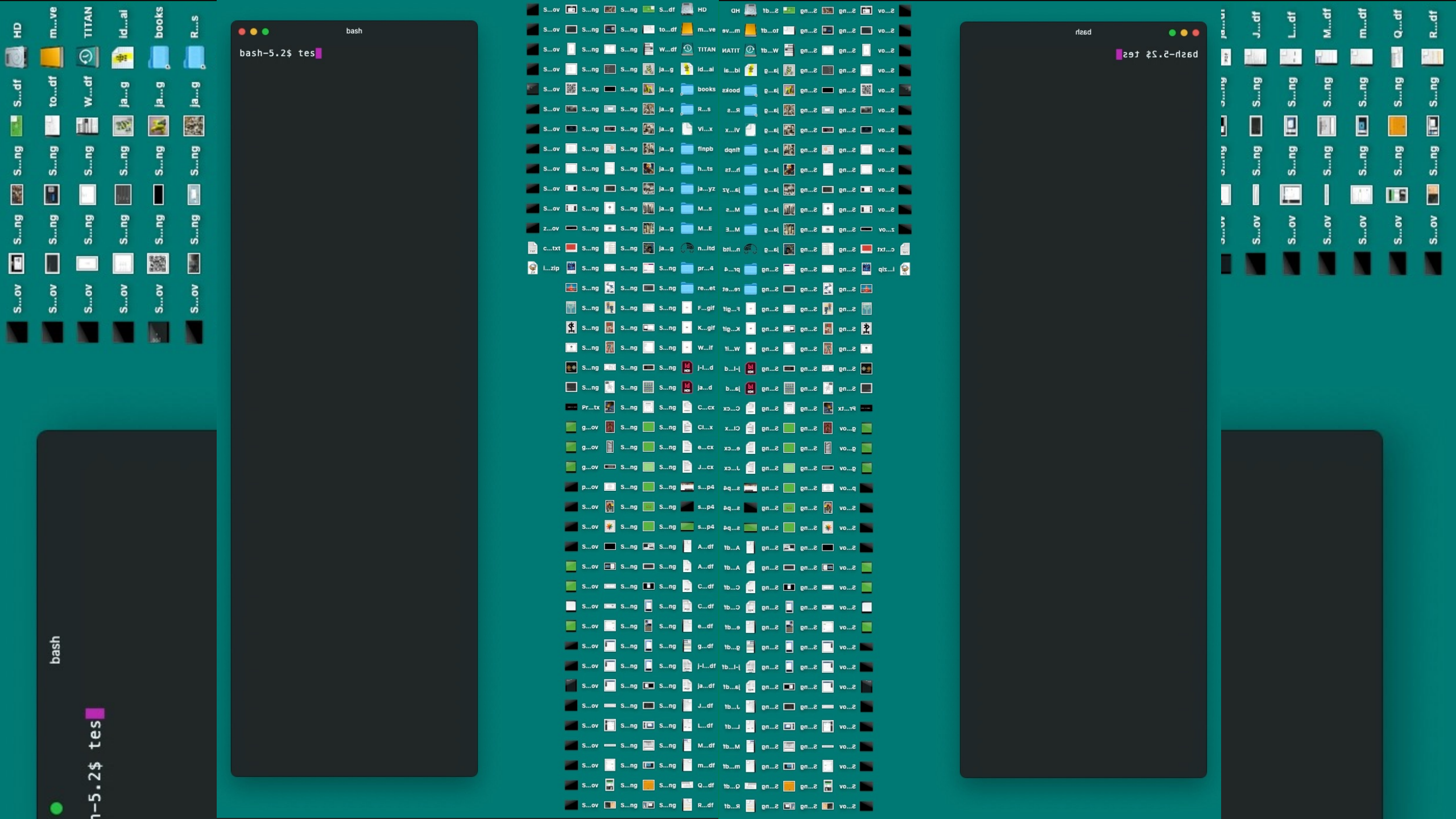
```
# we will create an alias to toggle our desktop.
# i use this before screen sharing to appear neat
# also: great in public places to detract shoulder surfing.
# disclaimer: limited to OSX environments.
# ----------------------------------------------------------
> change into commandlinedemo
> create a temporary variable
> add aliases to configuration file, source config.
> use our new aliases
> view tree
```

```
GNU nano 7.2

12 # Lines 1...11 redacted for display purposes.
13
14 # Toggle Desk
15 alias deskon="defaults write com.apple.finder CreateDesktop
   true;killall Finder"
16 alias deskoff="defaults write com.apple.finder CreateDesktop
   false;killall Finder"
```

```
bash-5.2$ cd ${PATH_TO_DEMO}
bash-5.2$ nano ${CONFIG}
bash-5.2$ source ${CONFIG}
bash-5.2$ deskon; deskoff
```

/// sourcing our config updates the file.
/// we have also accessed a hidden file, .bashrc, our configuration file.
/// we now have two custom aliases to use versus typing the long command each time.
/// we have also changed permissions for all .sh files by using a wildcard operator.


bash
bash-5.2$ tes

```
bash-5.2$ tree
.
|--- scripts
     |--- voicenote.sh
     |--- welcomemsg.sh
|--- voice-notes
     |--- 20230809231806_voicenote_testnote.aiff

3 directories, 3 files
```

touch	creates a file
./${SCRIPT}	runs the script
VAR=	create a variable
${}	variable construct
chmod u+x	execute permission
~./bashrc	configuration file
cd	change into directory
nano	opens nano text editor
open .	opens directory in gui
<aliasname>	runs the named shortcut
mv	moves a file (or renames)
source	runs commands listed in a file
#!/usr/bin/env bash	commands in script executable(actionable)

[+] do you think we should make
a script to organize our desktop?

[byeds, stopds]

```
# desktop services is a hidden file in the OSX world.
# it stores info on folders, icons, and other visual info.
# .DS_Store creates like to create itself a lot ☹
# it [.DS_Store] could expose info about your application;
# here, we will create a python script to remove instances.
# ----------------------------------------------------------
> find .DS_Store files in a given path.
> remove each found path.
> add alias to configuration file.
> create an alias to stop desktop services creation.
```

```
bash-5.2$ cd ${PATH_TO_DEMO}
bash-5.2$ SCRIPT=byeds.py
bash-5.2$ touch ${SCRIPT};chmod u+x ${SCRIPT}
bash-5.2$ mv ${SCRIPT} ${PATH_TO_DEMO_SCRIPTS}
bash-5.2$ nano ${SCRIPT_FOLDER}/${SCRIPT}
```

GNU nano 7.2

```
1 #!/usr/bin/env python3
2 # @usage python byeds.py <path>
3 import os, sys
4 print("Locating .DS_Store files.")
5 try:
6     for root, directories, files in os.walk(sys.argv[1]):
7     for file in files:
8         if file.endswith(".DS_Store") or file==".DS_Store":
9             joinedPath = os.path.join(root, file)
10            print("Deleting ==>", joinedPath)
11            if os.remove(joinedPath):
12                print("Cannot delete a directory in this context.")
13            else:
14                print("Deleted.")
15    print(".DS_Store files deleted:", counter)
16 except IndexError:
17     print("Check your arguments.")
18     print("Scripts expects you to pass in a location on the file system.")
```

```
GNU nano 7.2

1 # Lines 1...16 redacted for display purposes.
17
18  # Stop .DS_Store
19  alias byeds="python3 ${PATH_TO_DEMO_SCRIPTS}/byeds.sh"
20  alias stopds="defaults write.com.apple.desktopservices
    DSDontWriteNetworkStores true"
```

```
bash-5.2$ nano ${CONFIG}
bash-5.2$ source ${CONFIG}
bash-5.2$ stopds
bash-5.2$ byeds <path-to-somewhere>
```

```
bash-5.2$ tree
.
|--- scripts
     |--- byeds.py
     |--- voicenote.sh
     |--- welcomemsg.sh
|--- voice-notes
     |--- 20230809231806_voicenote_testnote.aiff
|--- welcomemsg_Cellos.aiff

3 directories, 5 files
```

VAR=	create a variable
${}	variable insertion
cd ~/<path>	change into directory
nano	opens nano text editor
<aliasname>	runs the named shortcut
mv	moves a file (or renames)
grep -q	searches input file quietly
chmod u+x	read/write/execute permission
locate	locates paths for a given file
alias	creates a named shortcut to run a command
#!/usr/bin/env bash	commands in script executable(actionable)

[+] are there any other files
you find as annoying as .DS_Store?

[+] it feels like we are doing less "coding".
is this ok?

(exhale && proceed)

[history -c]

/// your terminal is storing every thing you type into it ☹

```
# history is great for recalling commands.
# if someone were to physically breach your device,
# your terminal history could be the hacker's delight.
# also, you why not just do it for the clean up?
# ----------------------------------------------------
> show the first and last 3 lines of our terminal history.
> run the [history -c] command
```

/// history storage has been more of problem in zsh than bash

```
bash-5.2$ history | head -n 3
1  echo "User: ${USER} Home: ${HOME} Shell: ${SHELL} Text editor:
${EDITOR}
2  mkdir -p ${DEMO_DIR}/${SUB_DIR}
3  pwd
bash-5.2$ history | tail -n 3 && echo "History: `history|wc-l`"
217  history | head -n 3
218  man mkdir
219  history | tail -n 3
History: 220
bash-5.2$ history -c
```

/// i created a script, but then realized this is the simplest way.

```
history -c    deletes terminal history.
tail -n       show n lines from end of file
head -n       show n lines from beginning of file
wc -l         counts lines in a file
|             pipe, connect output of first command to next input
```

/// the more you use pipes, the less strange they become ☺

[+] no questions..let's move on ☺

[grep] <something> <somewhere>

/// let's get both specific && expansive

```
# grep is how we look for specific data within a file.
# regular expressions amplify grep by enabling pattern defining.
# we will create multiple patterns to account for data variations.
# ---------------------------------------------------------------
> create a subdirectory for our search results.
> create some test data.
> look for email addresses in test data.
> look for phone numbers in test data.
> look for ip addresses in test data.
> look for social security numbers in test data.
> look for btc/eth addresses in test data.
> create aliases for all of the above.
```

```
bash-5.2$ cd ${PATH_TO_DEMO}
bash-5.2$ GREP_DIR=grepping-for-data
bash-5.2$ TEST_DATA=test-data.txt; touch ${TEST_DATA}
bash-5.2$ mkdir -p ${PATH_TO_DEMO}/${GREP_DIR}
bash-5.2$ touch
              btcgrep.sh emailgrep.sh ethgrep.sh
              ipgrep.sh phonegrep.sh ssngrep.sh
bash-5.2$ chmod u+x *.sh; mv *.sh ${PATH_TO_DEMO_SCRIPTS}
bash-5.2$ nano ${CONFIG}
bash-5.2$ echo "ctrl+l to clear"; clear;
```

GNU nano 7.2

```
1 # Lines 1...20 redacted for display purposes.
21
22 # Grep Aliases
23 alias btcgrep=${PATH_TO_DEMO_SCRIPTS}/btcgrep.sh
24 alias emailgrep=${PATH_TO_DEMO_SCRIPTS}/emailgrep.sh
25 alias ethgrep=${PATH_TO_DEMO_SCRIPTS}/ethgrep.sh
26 alias ipgrep=${PATH_TO_DEMO_SCRIPTS}/ipgrep.sh
27 alias phonegrep=${PATH_TO_DEMO_SCRIPTS}/phonegrep.sh
28 alias ssngrep=${PATH_TO_DEMO_SCRIPTS}/ssngrep.sh
```

```
bash-5.2$ source ${CONFIG}
bash-5.2$ cat ${TEST_DATA}
```

```
bash-5.2$ cat ${TEST_DATA}
```

me@me.com	426-17-3059	U7H3dHgvaCZi	724.287.1647
them@them.com	426.17.3059	mnaEoJfD4tnrwFchDp5bbU	472-285-7657
us@us.com	376-36-9536	m23P3h65oaMU	314.426.8502
weirdnet.people@411.or	376364532	msU9NpWgtZXRK3yXPQisPk	4722125115
g	0.0.0.0	u1hEm6PxjzDr	505.925.7019
woah_what@mice.net	1.1.1.1	n1Wuwj6js8QXDQkfMUZdfX	(224) 878.7488
people-under-the-	11.1.1.11	aFQ9YeGYKSqo	505-525-3493
building@forever.com	256.34.23.06	0xCF79486A5BB20E265DE8	(505) 6463597
icouldgoonforever@movi	9.9.9.9	C7BE26937086E48A2A02	505-482-6064
e-quotes.com	243.203.230.150	0x45B47B5D51E00661E30C	5056461118
000-00-0001	249.11.116.204	1E4370A78EA712281A03	225-485-1636
004-00-0002	43.39.198.6	0x8686FCD10E365C603248	2255479218
005-00-0003	89.83.11.250	F305903064B396BE53E9	472-225-4548
005.00.0008	183.108.221.170	0x2B5938C52E7CE13C9AC8	505-527-4138
006-00-0004	77.221.149.157	DEBE925E89234F41CD25	505-644-6493
007-00-0005	28.63.187.243	0x1A12C452A2150E7F91B2	5056448090
470-03-5601	54.159.145.155	8604A5F4B93AD852CF12	472-286-1761
470033321	24.26.112.177	202-918-2132	472 252 0683
289-70-7913	226.191.13.96	505-468-1448	505-644-8616
289-70-7913	mwPA4CsmDZK8weBA7STAsc	472-287-9647	505 644 1835
510-64-1078	nTttHp6xgAX5	(307) 643-6083	505-579-4943
510.65.1078	n43kgUHMceCZdyr7u3fJNX	(427) 287-2647	472 294 6599

```
bash-5.2$ echo "Let's write our scripts!"
bash-5.2$ nano ${PATH_TO_DEMO_SCRIPTS}/btcgrep.sh
bash-5.2$ nano ${PATH_TO_DEMO_SCRIPTS}/ethgrep.sh
bash-5.2$ nano ${PATH_TO_DEMO_SCRIPTS}/emailgrep.sh
bash-5.2$ nano ${PATH_TO_DEMO_SCRIPTS}/ipgrep.sh
bash-5.2$ nano ${PATH_TO_DEMO_SCRIPTS}/phonegrep.sh
bash-5.2$ nano ${PATH_TO_DEMO_SCRIPTS}/ssngrep.sh
```

GNU nano 7.2

```
1  #!/usr/bin/env bash
2  # @usage btcgrep.sh ${FILE} {OUTPUT_DEST}
3  # BTC pattern can't begin with 0x, and is between 24-35 characters
4  FILE=${1};OUTPUT_DEST=${2}
5  BTC_PATTERN="^[a-wA-Wy-zY-Z1-9]{2}[a-zA-Z0-9]{22,33}"
6  if [[ ! -f ${FILE} ]];then
7    echo "${FILE} not found.";exit 1
8  else
9    grep -Eon ${BTC_PATTERN} ${FILE} > ${OUTPUT_DEST}/btc-grep-results.txt
10   exit 0
11 fi
```

```
bash-5.2$ btcgrep ${TEST_DATA} ${GREP_DIR}
40:mwPA4CsmDZK8weBA7STAscnTttHp6xgAX5
41:n43kgUHMceCZdyr7u3fJNXU7H3dHgvaCZi
42:mnaEoJfD4tnrwFchDp5bbUm23P3h65oaMU
43:msU9NpWgtZXRK3yXPQisPku1hEm6PxjzDr
44:n1Wuwj6js8QXDQkfMUZdfXaFQ9YeGYKSqo
```

GNU nano 7.2

```
1  #!/usr/bin/env bash
2  # @usage btcgrep.sh ${FILE} {OUTPUT_DEST}
3  # ETH addresses begin with 0x, are 42 characters in length
4  FILE=${1};OUTPUT_DEST=${2}
5  ETH_PATTERN="^[0x]{2}[a-zA-Z0-9]{40}"
6  # Commands to run.
7  if [[ ! -f ${FILE} ]];then
8    echo ${FILE} not found.;exit 1
9  else
10     grep -Eon ${ETH_PATTERN} ${FILE}
       > ${OUTPUT_DEST}/eth-grep-results.txt
       exit 0
11 fi
```

```
bash-5.2$ ethgrep ${TEST_DATA} ${GREP_DIR}
45:0xCF79486A5BB20E265DE8C7BE26937086E48A2A02
46:0x45B47B5D51E00661E30C1E4370A78EA712281A03
47:0x8686FCD10E365C603248F305903064B396BE53E9
48:0x2B5938C52E7CE13C9AC8DEBE925E89234F41CD25
49:0x1A12C452A2150E7F91B28604A5F4B93AD852CF12
```

GNU nano 7.2

```
1  #!/usr/bin/env bash
2  # @usage emailgrep.sh ${FILE} {OUTPUT_DEST}
3  FILE=${1};OUTPUT_DEST=${2}
4  AZazPUNC=[a-zA-Z0-9._-]
5  AZazPUNC2=[a-zA-Z0-9_-]
6  EMAIL_PATTERN="^${AZazPUNC}\+@${AZazPUNC2}\+\.[a-z]\{2,\}"
7  if [[ ! -f ${FILE} ]];then
8     echo ${FILE} not found.;exit 1
9  else
10    grep -win ${EMAIL_PATTERN} ${FILE}
      > ${OUTPUT_DEST}/email-grep-results.txt; exit 0
11 fi
```

```
bash-5.2$ email ${TEST_DATA} ${GREP_DIR}
1:me@me.com
2:them@them.com
3:us@us.com
4:weirdnet.people@411.org
5:woah_what@mice.net
6:people-under-the-building@forever.com
7:icouldgoonforever@movie-quotes.com
```

GNU nano 7.2

```
1 #!/usr/bin/env bash
2 # @usage ipgrep.sh ${FILE}
3 {OUTPUT_DEST}FILE=${1};OUTPUT_DEST=${2}
4 IP_PATTERN="([0-9]{1,3}[\.]){3}[0-9]{1,3}"
5 if [[ ! -f ${FILE} ]];then
6    echo ${FILE} not found.;exit 1
7 else
8    grep -Eo ${IP_PATTERN} ${FILE} > ${OUTPUT_DEST}/ip-grep-
     results.txt;exit 0
9 fi
```

```
bash-5.2$ ipgrep ${TEST_DATA} ${GREP_DIR}
25:0.0.0.0
26:1.1.1.1
27:11.1.1.11
28:256.34.23.06
29:9.9.9.9
30:243.203.230.150
31:249.11.116.204
32:43.39.198.6
33:89.83.11.250
34:183.108.221.170
35:77.221.149.157
36:28.63.187.243
37:54.159.145.155
38:24.26.112.177
39:226.191.13.96
```

GNU nano 7.2

```
1  #!/usr/bin/env bash
2  # @usage phonegrep.sh ${FILE}
3  FILE=${1}; OUTPUT_DEST=${2}
4  AREA_CODE_GROUP="^((\([0-9]{3}\)\s)|([0-9]{3}\-)|([0-9]{3}\.)|[0-9]{3})"
5  MIDDLE_3_GROUP="([0-9]{3}\-|[0-9]{3}\.|[0-9]{3}\s|[0-9]{3})"
6  PHONE_PATTERN="${AREA_CODE_GROUP}${MIDDLE_3_GROUP}[0-9]{4}"
7  if [[ ! -f ${FILE} ]];then
8     echo ${FILE} not found.;exit 1
9  else
10    grep -Eon ${PHONE_PATTERN} ${FILE}
      > ${OUTPUT_DEST}/phone-grep-results.txt;exit 0
11 fi
```

```
bash-5.2$ phonegrep ${TEST_DATA} ${GREP_DIR}

50:202-918-2132
51:505-468-1448
52:472-287-9647
53:(307) 643-6083
54:(427) 287-2647
55:724.287.1647
56:472-285-7657
57:314.426.8502
58:4722125115
59:505.925.7019
60:(224) 878.7488
61:505-525-3493
62:(505) 6463597
63:505-482-6064
64:5056461118
65:225-485-1636
66:2255479218
67:472-225-4548
68:505-527-4138
69:505-644-6493
70:5056448090
71:472-286-1761
73:505-644-8616
75:505-579-4943
```

GNU nano 7.2

```
1 #!/usr/bin/env bash
2 # @usage ssngrep.sh ${FILE} {OUTPUT_DEST}
3 FILE=${1};OUTPUT_DEST=${2}
4 SSN_PATTERN="(([0-9]{3}[\-]|[0-9]{3}[\.]|[0-9]{3}[\s])([0-9]{2}[\-]|[0-
  9]{2}[\.]|[0-9]{2}[\s])([0-9]{4}))"
5 if [[ ! -f ${FILE} ]];then
6     echo ${FILE} not found.;exit 1
7 else
8     grep -Eon ${SSN_PATTERN} ${FILE} > ${OUTPUT_DEST}/ssn-grep-
      results.txt;exit 0
9 fi
```

```
bash-5.2$ ssngrep ${TEST_DATA} ${GREP_DIR}

50:202-918-2132                    63:505-482-6064
51:505-468-1448                    64:5056461118
52:472-287-9647                    65:225-485-1636
53:(307) 643-6083                  66:2255479218
54:(427) 287-2647                  67:472-225-4548
55:724.287.1647                    68:505-527-4138
56:472-285-7657                    69:505-644-6493
57:314.426.8502                    70:5056448090
58:4722125115                      71:472-286-1761
59:505.925.7019                    73:505-644-8616
60:(224) 878.7488                  75:505-579-4943
61:505-525-3493
62:(505) 6463597
```

```
bash-5.2$ tree
.
|--- grepping-for-data
     |--- btc-grep-results.txt
     |--- email-grep-results.txt
     |--- eth-grep-results.txt
     |--- ip-grep-results.txt
     |--- phone-grep-results.txt
     |--- ssn-grep-results.txt
|--- scripts
     |--- btcgrep.sh
     |--- byeds.py
     |--- emailgrep.sh
     |--- ethgrep.sh
     |--- ipgrep.sh
     |--- phonegrep.sh
     |--- ssngrep.sh
     |--- voicenote.sh
     |--- welcomemsg.sh
|--- test-data.txt
|--- voice-notes
     |--- 20230809231806_voicenote_testnote.aiff
|--- welcomemsg_Cellos.aiff

6 directories, 18 files
```

! -f	not a file
echo	write to output
VAR=	create a variable
${}	variable insertion
cd ~/<path>	change into directory
nano	opens nano text editor
<aliasname>	runs the named shortcut
mv	moves a file (or renames)
chmod u+x	read/write/execute permission
tail -n	show n lines from end of file
head -n	show n lines from beginning of file
grep -Eon	extended regex, exact-match, line number
alias	creates a named shortcut to run a command
#!/usr/bin/env bash	commands in script executable(actionable)
grep -win	ignore casing, treat match as word, line number

[+] does everything have a pattern?

[+] how do you see yourself using grep?

[+] do you find the regular expression syntax weird?

[goreadme] <DIR> <PROJECT_NAME>

```
# READMEs provide context. let's generate one from prompts.
# the prompts are loosely development/project planning.
# please tweak to your needs though ☺
# ----------------------------------------------------------
> create a new script
> script expects an absolute path, project name as arguments.
> if directory exists: use this path
> else: exit
> prompt with README questions
> echo answers into README. use markdown formatting.
> display README
```

/// absolute paths are the full address to somewhere in the file system.
example: ~/mysecretfiles/moresecretfiles/othersecretfiles/this-file.txt

```
bash-5.2$ cd ${PATH_TO_DEMO}
bash-5.2$ SCRIPT=goreadme.sh; touch ${SCRIPT};
bash-5.2$ chmod u+x ${SCRIPT};mv ${SCRIPT_FOLDER}
bash-5.2$ TEST_DIR="test-project"; mkdir ${TEST_DIR}
bash-5.2$ ABS_PATH=${PATH_TO_DEMO}/{$TEST_DIR}
bash-5.2$ README_DESC="Test README"
bash-5.2$ nano ${CONFIG}
```

GNU nano 7.2

```
1 # Lines 1...28 redacted for display purposes.
29
30 # goreadme
31 alias goreadme=${PATH_TO_DEMO_SCRIPTS}/goreadme.sh
```

```
bash-5.2$ source ${CONFIG}
bash-5.2$ nano ${SCRIPT}
```

GNU nano 7.2

```
1 #!/usr/bin/env bash
2 # @usage ./goreadme.sh ${ABS_PATH} ${README_TITLE}
3 SCRIPT_NAME=${0}; PATH_TO_DIRECTORY=${1}; PROJECT_NAME=${2};
4 README_FILE="README.md"
# lines 11 through 105 omitted for demo. see full code in repo.
106 if [[ ! -d ${PATH_TO_DIRECTORY} ]];then
107     echo ${DIRECTORY_DOES_NOT_EXIST}; script_help
108     exit 1
109 else
110     echo ${DIRECTORY_EXISTS}; script_context
111     cd ${PROJECT_PATH}
112     create_README ${README_FILE} ${README_TITLE}
113     run_prompt ${PATH_TO_DIRECTORY} ${README_FILE}
114     cat ${README_FILE} | lessf
115 fi
```

```
bash-5.2$ ${PATH_TO_SCRIPTS}/goreadme.sh ${ABS_PATH} ${README_DESC}
bash-5.2$ *** Creating a README.md file for 'Test README' ***
bash-5.2$ Short description of project >>>
bash-5.2$ What is motivating this project? >>>
bash-5.2$ What obstacles or resistance might you encounter? >>>
bash-5.2$ Is this for profit or pure testing/research? Or both? >>>
bash-5.2$ (1/3) What technologies are you using? >>>
bash-5.2$ (1/3) List a feature of your project >>>
bash-5.2$ Creating sections to modify later.
bash-5.2$ Enter a contributor to your project >>>
```

/// experiencing bug somewhere, so running script directly instead of as alias.
/// update: bug was in my config, had the = sign in the path name 🤬

```
bash-5.2$ cat ${ABS_PATH}/README.md

Tue Aug  8 01:03:20 EDT 2023
# Test README
## ✍️ Description
## 💡 Motivation
## 🌋 Obstacles/Challenges
## 🤑 Profit
## 🛠️ Technology
## 🔥 Features
## ⚠️ Installation
## 🏁 Milestones
## 👽 Future Plans for Test README
## 🙏🏾 Credits
```

/// this is an excerpt of the README we just generated.
/// all of our answers to the prompts will appear in their assigned sections.

```
bash-5.2$ tree
.
|--- grepping-for-data
     |--- btc-grep-results.txt
     |--- email-grep-results.txt
     |--- eth-grep-results.txt
     |--- ip-grep-results.txt
     |--- phone-grep-results.txt
     |--- ssn-grep-results.txt
|--- scripts
     |--- btcgrep.sh
     |--- byeds.py
     |--- emailgrep.sh
     |--- ethgrep.sh
     |--- ipgrep.sh
     |--- phonegrep.sh
     |--- ssngrep.sh
     |--- welcomemsg.sh
|--- test-data.txt
|--- test-project
     |--- README.md
|--- voice-notes
     |--- 20230809231806_voicenote_testnote.aiff
|--- welcomemsg_Cellos.aiff

7 directories, 18 files
```

-p	prompt flag
cat	reads a file
>>	append output
! -d	not a directory
${}	variable construct
cd ~/<path>	change into directory
nano	opens nano text editor
mv	moves a file (or renames)
read	built-in command for input
>	redirect output of command
|	pipe connects output to input
chmod u+x	read/write/execute permission
..	go a level up in the directory
less	pulls up a pager view for a file
;	allows multiple commands on one line
#!/usr/bin/env bash	commands in script executable(actionable)
&&	and operator for running multiple commands

[+] are there any open-source projects which could use this?

[+] do you hate the README a little less now?

[+] what prompts would you add based on your project needs and workflow?

(eject && transfer)

linkedin

my website

discord

repo

q & a; gifts