~/bin/ there, done that!

A gentle guide to working with Bash

By: Jason Franklin

@lifecrisis on GitHub
@lifecrisis on GitLab

Slides at https://bit.ly/31pTFmB

Roadmap

1. Introducing myself and this talk
2. Tools
3. An example
4. Learning materials and resources
5. Closing comments

Part I: Introductions

Who am I?

- BS in math from UGA
- 1 year of Apache Spark and Python
- 2.5 years of LAMP among other things
- https://github.com/scrooloose/nerdtree
- 21+ patches to the Vim core
- I love bash

This talk IS NOT …

- … about POSIX
- … a how-to lesson
- … a list of “bash tricks to blow your mind!”
- … about testing your scripts :(

This talk IS …

Slides at https://bit.ly/31pTFmB

You have a problem.

Part II: Tools

Tools Overview

- Git (never forget!)
- Make
- Snippets
  - https://github.com/SirVer/ultisnips
- Shellcheck
  - https://www.shellcheck.net/
- Your prompt!

Git and Make

- When you learn something cool, never forget
- Automate the installation of your scripts
- Fork this repository to get started:
  - https://github.com/lifecrisis/scripts
- The README.md file explains it all!

demo-snippets

demo-shellcheck

https://raw.githubusercontent.com/git/git/master/contrib/git-jump/git-jump

… or locate git-jump

demo of prompt

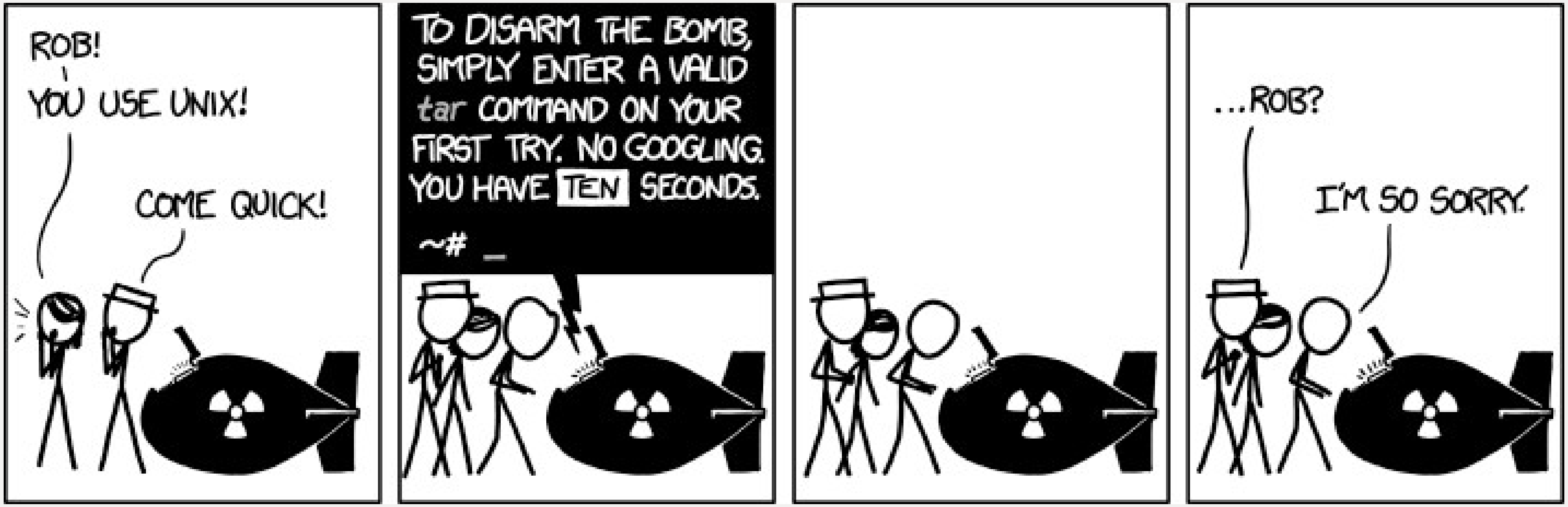
Bonus: demo of Vim!

Part III: An Example

Our example: tip

Repository: https://github.com/lifecrisis/bash-tip

See also: https://github.com/cheat/cheat


ROB!
YOU USE UNIX!
COME QUICK!
TO DISARM THE BOMB, SIMPLY ENTER A VALID tar COMMAND ON YOUR FIRST TRY. NO GOOGLING. YOU HAVE TEN SECONDS.
~# _
...ROB?
I'M SO SORRY.

Slides at https://bit.ly/31pTFmB

demo of “tip”

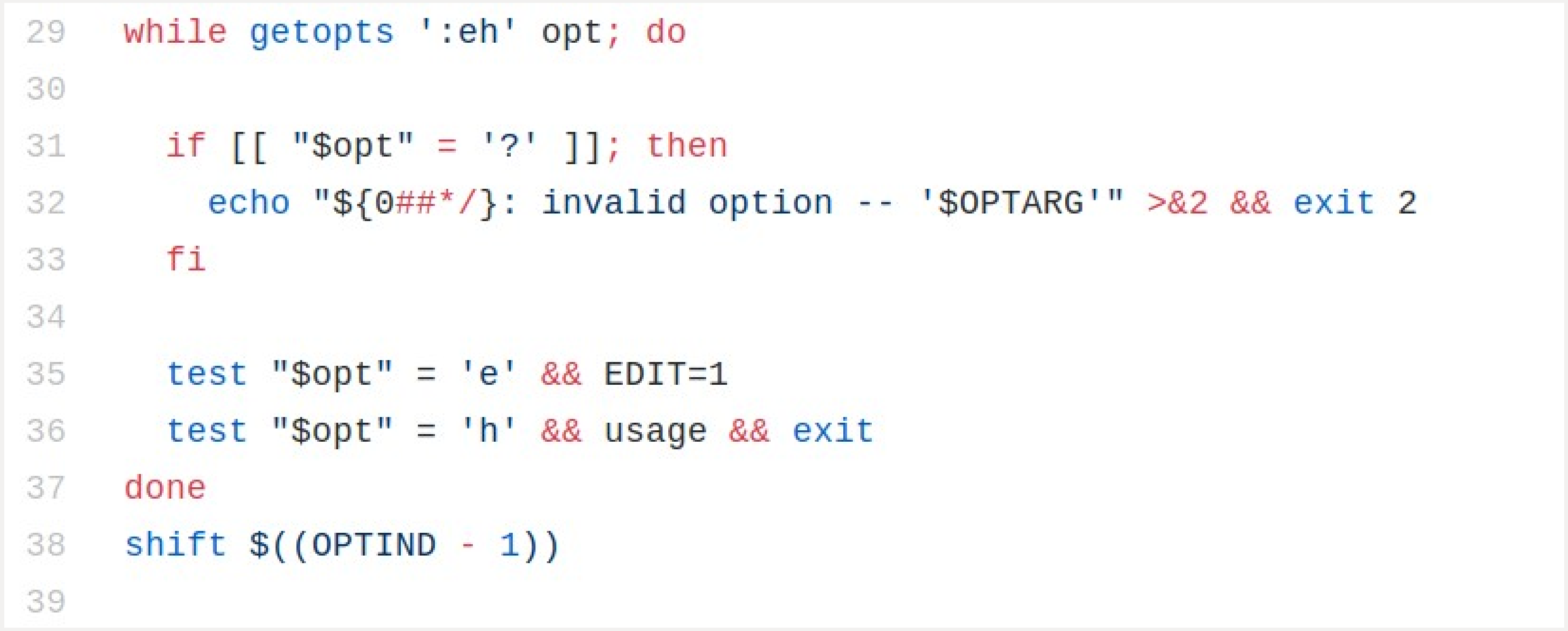
Initialization

```bash
# The "TIP_DIR" environment variable tells us where tips are kept.
if [[ -z "${TIP_DIR+x}" ]]; then
  echo 'tip: not configured' >&2 && exit 2
fi

# Remove trailing slash.
TIP_DIR="${TIP_DIR%/}"

# Default behavior is to print the tip.
EDIT=0

```

Usage function

```
usage() {
  cat <<EOF
Usage: tip [-e] [-h] [--] COMMAND
Print tips on the usage of the given COMMAND.  Use the "-e" option to edit
the named tip file.
EOF
}

```

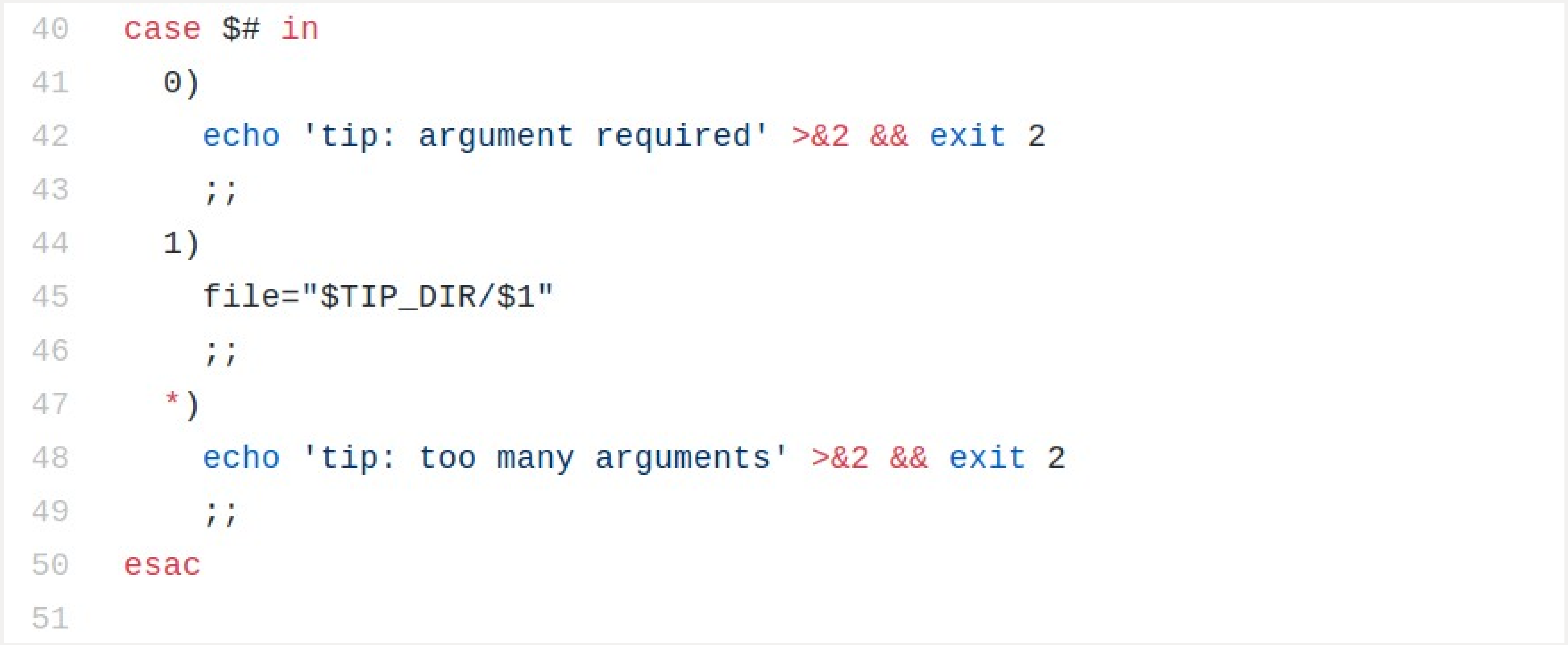
Option processing



```
while getopts ':eh' opt; do

  if [[ "$opt" = '?' ]]; then
    echo "${0##*/}: invalid option -- '$OPTARG'" >&2 && exit 2
  fi

  test "$opt" = 'e' && EDIT=1
  test "$opt" = 'h' && usage && exit
done
shift $((OPTIND - 1))

```

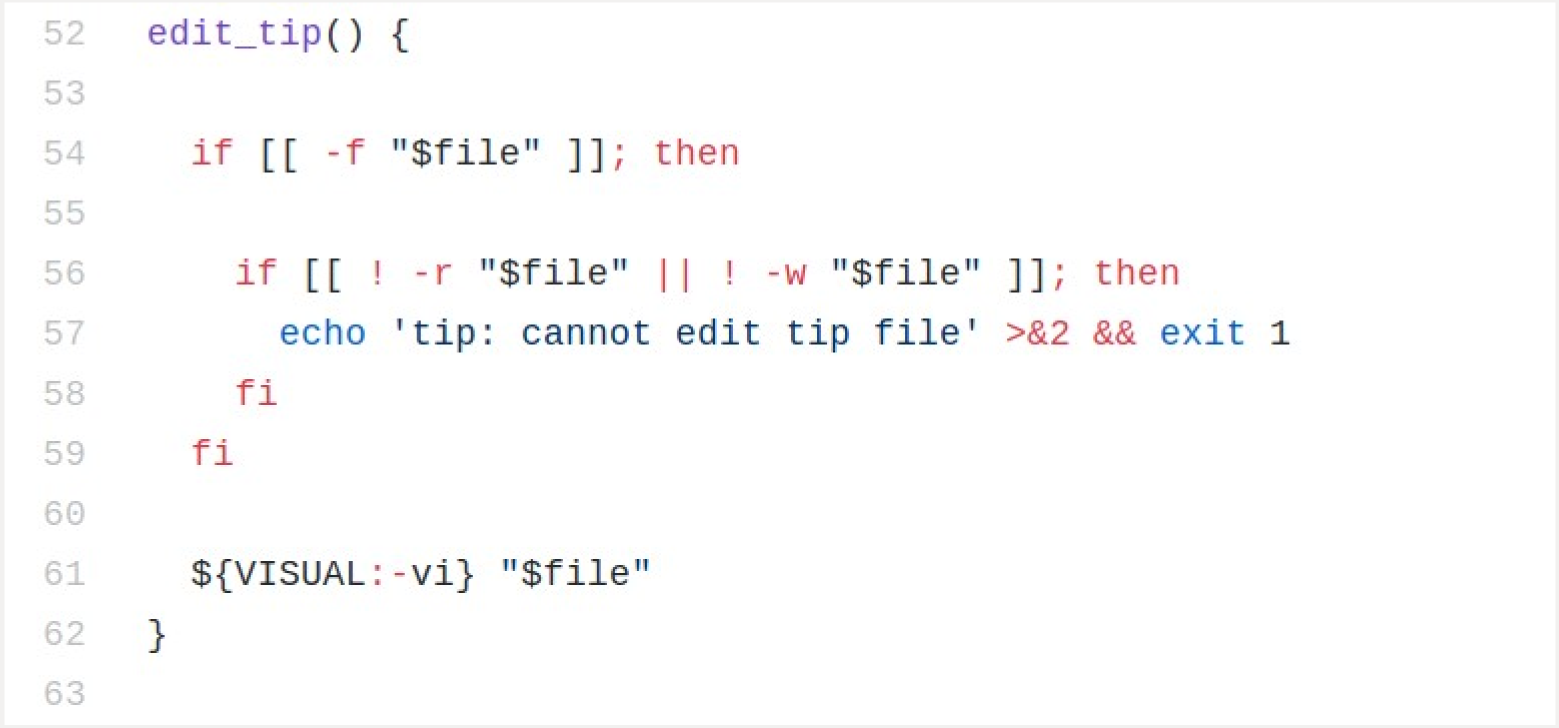
Slides at https://bit.ly/31pTFmB

Argument processing



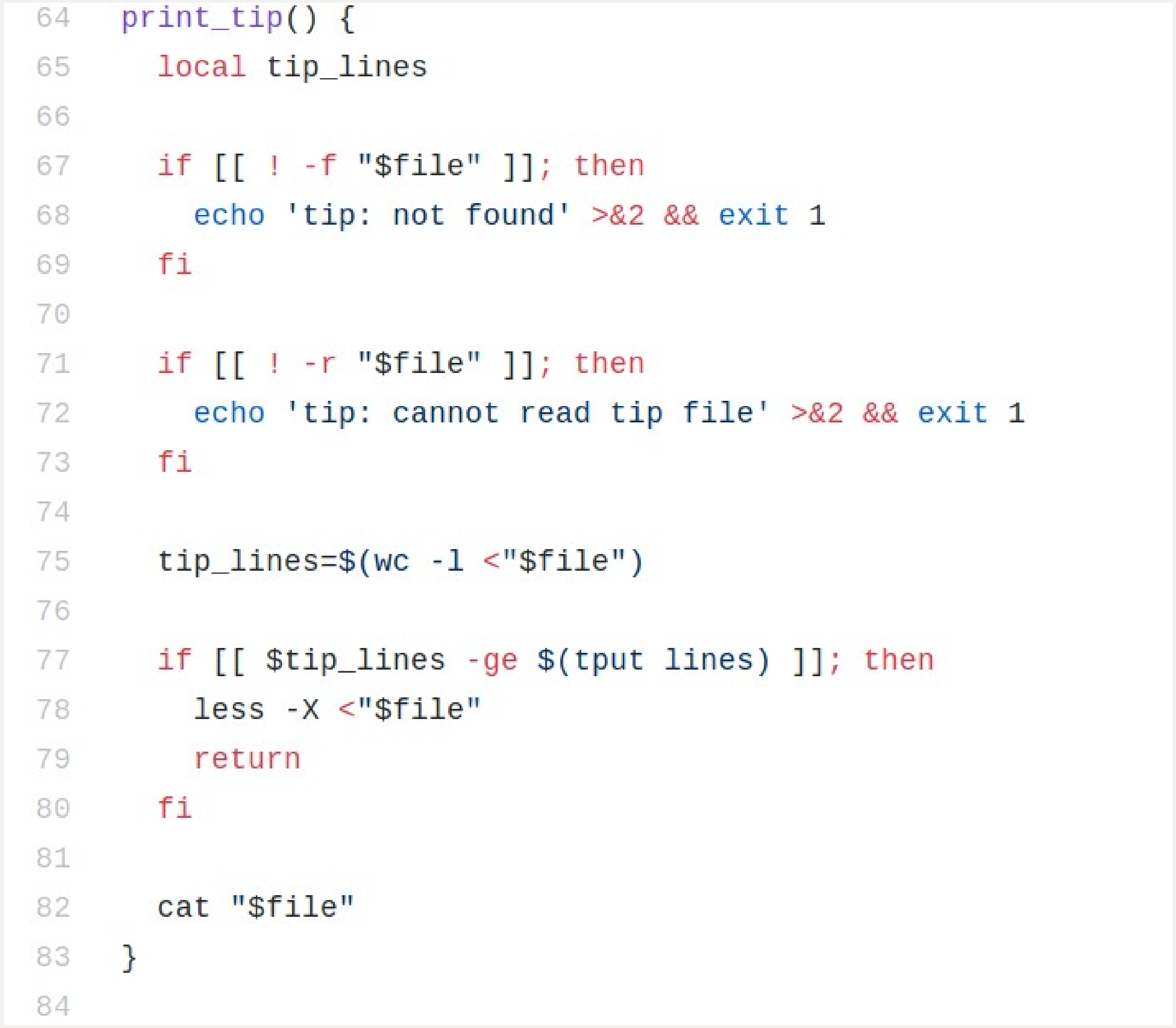
```
case $# in
  0)
    echo 'tip: argument required' >&2 && exit 2
    ;;
  1)
    file="$TIP_DIR/$1"
    ;;
  *)
    echo 'tip: too many arguments' >&2 && exit 2
    ;;
esac

```

Edit tip


```
edit_tip() {

  if [[ -f "$file" ]]; then

    if [[ ! -r "$file" || ! -w "$file" ]]; then
      echo 'tip: cannot edit tip file' >&2 && exit 1
    fi
  fi

  ${VISUAL:-vi} "$file"
}

```

Slides at https://bit.ly/31pTFmB

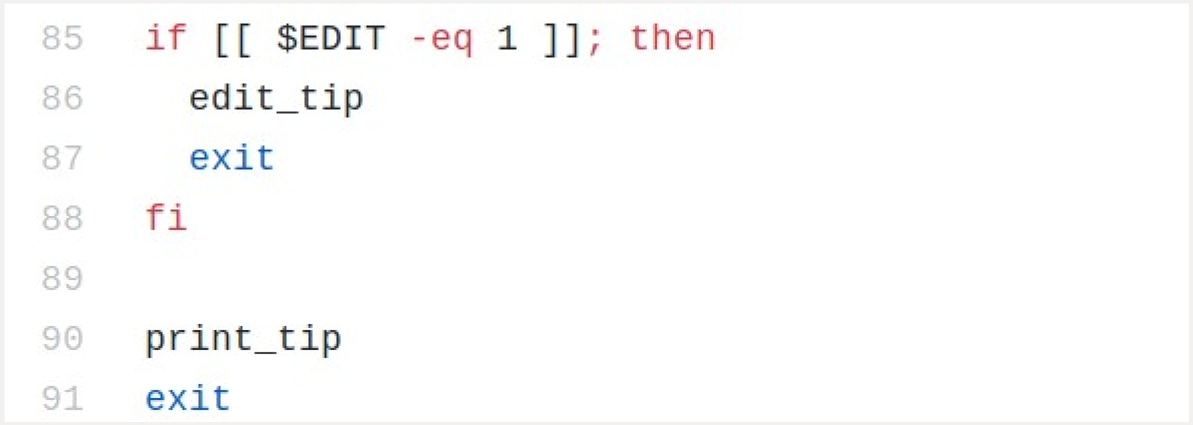
Print tip



```
print_tip() {
  local tip_lines

  if [[ ! -f "$file" ]]; then
    echo 'tip: not found' >&2 && exit 1
  fi

  if [[ ! -r "$file" ]]; then
    echo 'tip: cannot read tip file' >&2 && exit 1
  fi

  tip_lines=$(wc -l <"$file")

  if [[ $tip_lines -ge $(tput lines) ]]; then
    less -X <"$file"
    return
  fi

  cat "$file"
}

```

Slides at https://bit.ly/31pTFmB

Take action!



```
if [[ $EDIT -eq 1 ]]; then
  edit_tip
  exit
fi

print_tip
exit
```

Slides at https://bit.ly/31pTFmB

Part IV: Learning Materials

Learning Materials List

- The Unix Programming Environment (ch. 1-5)
  - https://en.wikipedia.org/wiki/The_Unix_Programming_Environment
- BashFAQ
  - https://mywiki.wooledge.org/BashFAQ
- “man bash” and “help”
  - Seriously.

Part V: Closing Comments

Please…

… leave this talk/conference with the resolve to ***do something*** for free software.

Inspiration.